Lecture 2

If we are given this particular framework, like zoom server and listening to the lecture, what can be the different attacks possible, if you are an attacker, given the current state, what would be the easiest way to get into the server?
What are the kind of applications and attacks you want to exploit the end-users?
If we are on our laptop, if we go to the address bar, there we can see a small lock, it means my connection to the laptop to the server is secure, all the information which we are communicating with the server is encrypted and no one can know what we are doing. If you are a hacker, it will be very very difficult to exploit it.
How does HTTPS work, what encryption mechanisms do we need to secure it
Foundation topics required to understand HTTPS
Types of attacks are evolving.
How do you access the services, standalone systems, browser-based services and again the standalone systems?
The platform itself was secure, standalone systems have other problems, ensuring security in this security paradigm was tough in the standalone systems.
People are shifting to apps. The research area is how to make it more and more secure.

What are the things you want to do at the server level if you want to break into the server?
DOS, DDOS, DNS attack, Logic bombs

When we talk about security, there are three actors in the security system, one is the user who is using the system, one is the administrator who is a kind of service provider like GMAIL, then there is an attacker who is trying to break into the system.

We have to design the systems keeping into mind the intention of all three users, it should be non-existing, works in the back, protecting the user all the time. If we talk of the administrator, it is worried about protecting the effectiveness of the security, kind of protecting the information, that is the administrator's goal. This needs to be a deterrent for the attacker. The majority of Windows systems have 90% vulnerabilities, but only 10% vulnerability is there for Linux as everyone can access the source code and they fix them and it is really very difficult to break into the system. Most open-source systems are more secure and efficient to work with.
Five main security properties
Any system you take needs to have a few of these if not all

1. Confidentiality- No unauthorised access of information
2. Integrity- Data has not been altered/tampered with
3. Availability- Data accessed as desired
4. Accountability- Non-repudiation, the user responsible for his action
5. Authentication- User or data origin acutely identifiable

Confidentiality is the concealment of the information. Integrity is the prevention of unauthorised changes. Authenticity is the identification and assurance of information. It is a variant of integrity. Availability is the ability to use resources/information desired.

If someone sent you an email to reset your FB account, if you give your password, accountability won't save you. In the same sense, authenticity is also violated, as you are not logging. RBI mandated multi-factor authentication. Accountability does not need strong authentication, it just means whatever actions are done is not repudiated later on. A signature is the best example. Let us say that signature is something that is unique to you, and it can is forged, if you sign a particular document, it means you are accountable.

Suppose you are authenticated but you are not maintaining records/logs of what you did, this means you didn't have any accountability. You may ask is accountability possible without authentication. Digital wallets are a perfect example, you use digital currency to do some trade, they are authenticated in a very discrete way, there your PII is not included, you may be authenticated but your identity is not revealed, all they care about is your private/public key, but no one really knows who you are. So, you would need to be authenticated for you to be accountable, there are variants of authentication, strong authentication requires your identity, on the other hand, weak authentication doesn't require your identity. Some sort of authentication is definitely required for you to be accountable for.

Anything which can link to who you are is strong authentication. We can have strong authentication and no accountability also. The authorisation is under authentication. There are different flavours of authentication, depends on what kind of authentication you need, if you are a service provider then you need weak authentication. Anonymous services also have authentication. Integrity doesn't mean confidentiality.

The whole view of the system is desired to know the weakest link.

The whole system is critical. The attacker will do all kinds of scans, vulnerability scans, active/passive scans, port scanning, these scans are done to know what the system is, what are the security settings of a particular system. The attacker does this to see what kind of security settings are there, identify the weak links. Once you know the weak link, attack it. To provide better defence, We need cryptography, implementation, people, physical security and everything in between to avoid this, we need to build a contingency plan for every day including physical security as that is also under our purview. Someone can use the vulnerability to legitimately breach your system. They will use default passwords to break into your system. Web server passwords are the default passwords. We need to have the whole system view of security, sylowed view of security won't help. Physical security is like if the server is situated in a room having access control, which cant work every time, and he is in your server room, you have to have a mechanism to protect the system. The weakest link is not necessarily the people part but most likely yes.

Lecture 3

From the user's point of view, security systems should be usable for both the provider i.e. admin and the user, it has to be usable, that they should use in the intended way it is meant to be used. Systems should be effective/desirable/efficient and this is more important to the admin and it should provide an effective deterrent for the attacker. These three properties are desired out of the security systems. Deterrence is very important.

Why any security system is not 100% secure?
No one does full-stack development i.e. if you are a developer you always rely on external libraries and you don't focus on and the second reason is the system is dynamic and it gets updated very often and the new update might bring new vulnerabilities, and you have no control over the effectiveness of a particular security system, it is always possible that the security systems have vulnerabilities that someone can exploit, so we should forget about thinking in binaries i.e either the system is secure and it is not secure it is not possible. there is always a shade of grey i.e. the security system
Provide enough deterrent to the attacker that they should not contribute to the vulnerabilities in the system. Economically it does not make any sense. Even if you develop a normal system, you don't work from scratch, you make more mistakes. You should make sure that the vulnerabilities are not exploitable. Sometimes these vulnerabilities are extrinsic to the system, we are not able to fix them. We should make sure that those vulnerabilities are not exploitable.

Confidentiality-> RSA, Integrity->Hashing, Availability->DDOS attack prevention, Authentication->Biometrics, Accountability->Signature

We can have accountability with unforgeability and it simply means with some kind of weak authentication.
The whole system is critical
Process:- What are the procedures in terms of access control, authorization and principles
Technology:- We talk about security systems, we talk about vulnerabilities, threats
People:- We talk about adversaries, competition.

What is a vulnerability: It is a weakness in a system that could be exploited to cause damage.

A vulnerability in a system doesn't mean that the system security should be breached. It is included in all kinds of systems, some of the vulnerabilities are exploited.
What is a threat: Threats are nothing but a set of actions by adversaries who try and exploit vulnerabilities to cause damage.
The threat is more active and it is more real to a system than vulnerability. It cannot pose any risk. Spoofing identity is a threat, DOS is a threat, Escalation of priveledges, tampering data is a threat as each of these exploits weakness in a system.
There are two kinds of vulnerabilities like known and unknown vulnerabilities.
What is authentication?
Binding of identity/entity to the subject. The subject can be a profile. The entity is some information about you which you can show to someone. You can consider Identity as Personally Identifiable Information (PII) and entity as a private key.
Authentication is essential for authorisation and access control. There are certain files that do not require authentication like amazon web apps. Within authentication, there are multiple factors within authentication. It is divided into three parts, what entity user knows, has and is?
The fourth factor is used in kind of particular terminals i.e. where the particular entity is.
Examples: User knows the password, User can have an IP address, smartcard, OTP etc, User is exactly biometrics example fingerprint etc. Once you authenticate, then you have to authorise, what can he do, what can he not do, authorisation are a bit different. Where the particular entity means, like you are allowed to book an Ola cab at one place whilst you are at the airport, based on your location, you are allowed to do book the taxicab.

Authentication Tuple <A,C,F,L,S>
A is information that proves identity, C is information stored in the system for validation(verified), F is a function that maps A to C and L is a function that proves identity. S is a function enabling an entity to create/alter the information in A or C. These are the five essential tuples we need to have to build an authentication system.
A is the password that is used to prove the identity of the user, C could be the hashed password. F is the hashing function. L is the comparison function, we need to compare what is given and what we stored in the database, it compares both the hashes. Many people do not understand what a hash is. It is a kind of function that uniquely maps x to y where $y = h(x)$.

Q in order to have multi-factor authentication, do we need to have a comparison function?
A.. yes

Lecture 4

Passwords

Passwords are some random sequence of characters which user chooses for authentication purposes. Sequences of words are like passphrases. Algorithms are there for One-time passwords which are generated randomly for certain levels of authentication. Entropy is when we make the password more and more complex, it is harder and harder to guess and obviously harder to remember. That is why we prefer writing them down. Storage as the clear text has a problem as if the password file is compromised, then all of the passwords will be revealed. Enciphering the file requires the need to have decipherment and encipherment keys in the memory. Again it reduces to the previous problem that if the file is compromised, then everything is revealed to the attacker.

Storing one way hash of the password means, even if the file is read, the attacker must still guess the password or invert the hash, now it depends on the hash function that we are using as if we use a very robust and uniform hash function then it is way more difficult for someone to invert the hash function to retrieve the original password.

Password Cracking

It involves social engineering attacks like malicious activities accomplished through human interaction, it is a kind of psychological manipulation to get to know some password, or by requesting money from a user and then calling him/her by saying, I have sent you the money please enter the OTP and money will be deposited in your account, which is a fraud actually. Some attackers, try to reset your password by guessing your date of birth, or your birthplace, or your pet's name or your favourite destination to visit, and then they reset the password. A surprisingly large number of password resetting is done as a way to crack the password. Some kinds of attacks are like keystroke logging, sniffing and shoulder surfing. Some ways are hash chains and rainbow tables. Rainbow table is a precomputed table for caching the output of cryptographic hash functions, usually for cracking password hashes. Rainbow tables are used to crack passwords very quickly.

OTP

They are the passwords that can be used exactly once. After use, it can be immediately invalidated, that is it cannot be used more than once. Certain OTPs also have the time in which they can be used like certain OTP's are valid only for 3 minutes, with the same restriction that they can only be used just once.

Hashing prevents unrolling of passwords even if the attacker has the password.

What the user has? OTP

Time synchronization, challenge-response and hash chains.

Challenge-response is like using a challenge from the server like for example c_n be the current challenge from the server. So f(c_n)= p_n then the passwords p in the order of use are $p_1, p_2, p_3,..., p_n$

Hash chain is like using a chain of hash functions with some seed s

h(s) =p_1

h(p_1)=p_2

h(p_2)=p_3 and so on

$h(p_{n-1}) = p_n$

The passwords in the order of use are p_n, $p_{n-1},..., p_3, p_2, p_1$

Challenge-response

The user and the system share a secret function f.

The user sends the request to the system to authenticate himself then the system sends the random message r as the challenge, now the user replies with a response f(r) back to the system. This is known as the challenge-response.

Hardware Support

Token-Based-> Used to compute the response to the challenge, it may require pin from the user. Others are like MS Authenticator which generates random tokens which then combined with the password jointly authenticates the user.

Biometrics

Automated measurement of biological, behavioural features that identify a person eg fingerprint, voice, eyes, face, keystroke dynamics like pressure and interval of stroke

The false reject rate is the rate at which authentic users are denied from accessing due to failures of the biometric device. It is often referred to as Type 1 error. The next one is the False accept rate, this is when non-legitimate users are allowed access to the systems, these are known as Type 2 errors. The crossover error rate is when the number of cross rejections are equal to the number of cross acceptance.

Lecture 5

Lecture 4 revision

We discussed passwords, how are they handle in the real world system, they are hashed and we discussed what are hashes and how passwords are hashed, we discussed three methods time synchronization, biometrics and challenge-response. Hash chaining is very simple, you have a seed and then the server and the client has access to the seed let us say they exchange it offline,

so the client and server start with the seed then they find hash on seed and then hash on hash on seed and so on and so forth. The client and server use a list of hashes in reverse order.

Biometrics

Different biometrics system which is available is facial detection, retina, fingerprint, gait, behavioural, voice which is based on what the user is, pictures.

Advantages of biometrics are:

1. It can be continuous
2. It is unobservable
3. It is convenient

Disadvantages of Biometrics are:

1. Needs additional hardware
2. False negatives are relatively higher

Authorisation

Authentication establishes credentials/identity. The next step is assigning specific access rights. Authentication does not mean unlimited access to any resource. Authorization is the process of assigning specific access rights to resources.

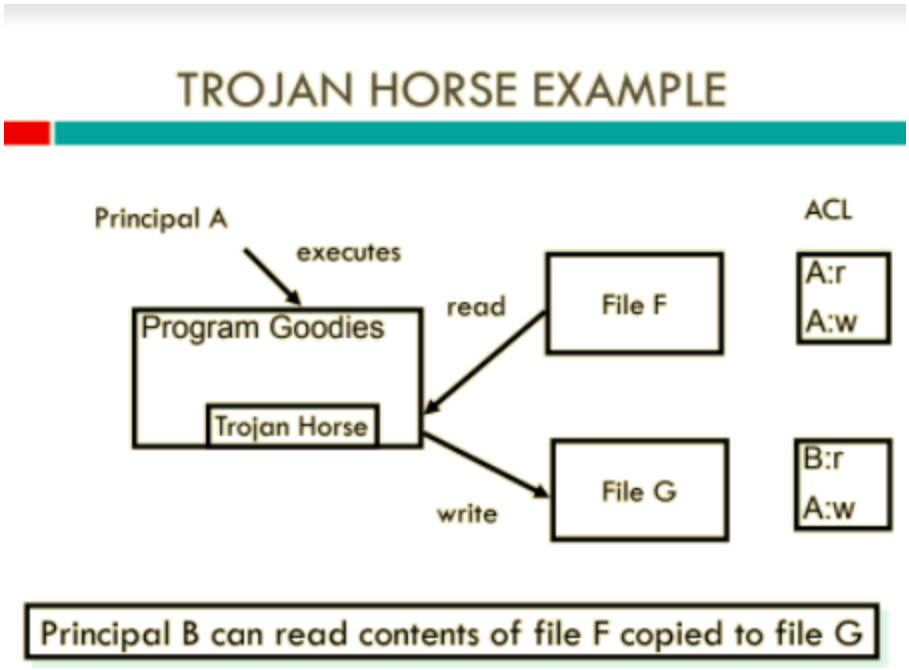
Usually, we have an access control matrix, which has a table in which we have users and files and then we have read, write and open rights to specific users.

There is something called the Access Control List (ACL), ACL is something like

ACL(file 1) - {User 1, R}, {User2, O}. It is nothing but the variant of the access control matrix. It is done after the authentication takes place.

Capabilities are like user can do what which specific file like he can read File 1 and he can only write into file 2 and so on. Capabilities require unforgeability i.e. weak authentication but ACL is file-based, it needs strong authentication to access resources.

Capabilities provide finer grain control, it is user-based.



How to prevent this from happening. We have the Bell-LaPadula security model, it is a very old and simple model. It classifies files in four categories like unclassified, confidential, secret and top secret. Something which can cost 1 human life is marked as secret, something which can cost many lives, it is marked as top secret. Basically no read up, No write down.

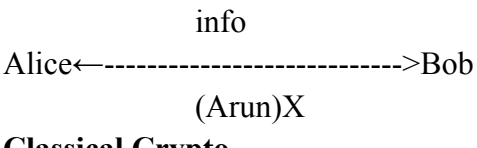
Discretionary Access Control: Restricting access to objects based on identity.
Mandatory Access Control: Restrict access to objects based on the sensitivity of the information.
Role-Based Access Control: We have users, roles and permissions, Here permissions are not directly assigned to the users. The permissions are directly assigned to roles. Then based on certain criteria, roles are assigned to the users. Each role has specific functionality.
RBAC0 is the base model, RBAC1 has a role hierarchy, it has a concept of inheritance, here roles are hierarchical, top-level roles inherence all responsibilities of bottom level roles. RBAC2 basically constraints level policies can be incorporated, it is more situational, based on a certain situation, it can decide, it is more heuristic. RBAC3 is a combination of RBAC1 and RBAC2.
Situational Aware RBAC: Here RBAC is a heuristic, it is not deterministic, it is semi-deterministic.

Lecture 6

Cryptography

It is used to enforce confidentiality and integrity(two properties of security of models).

Problem how do Alice and Bob exchange information without Arun (hacker) knowing about info.



Classical Crypto

1] substitution - substitute one character with another .example, caesar cypher
Key:2
Ciphertext: Fgxcujk (shifting each char by key-value)
Ciphertext: code which we get after converting plain text by using the key.
Problem: hacker can find the key by comparing difference .finding replacement of commonly used char like e--->__ , a--->__. Easy to crack key, and we can substitute for getting original info.
This happens because there is 1 to 1 mapping. Prone to attack more.

2] Transposition: instead of substitution, we will change their location.
Text: Meet me

```
M  e  m
  e  t  e
   |||
```

Cypher text: Memete

Hackers can also decipher this.
Another type:
Vigenere Cipher: built in 1863.
Plain Text: Iattack
Key: 234
Cipher Text:kdxvdym
How are we doing: first char shifted by 2, second char moved by 3, third by four and repeat.
I.e. variable substitution. No 1 to 1 character mapping.
Dependant on the key, still someone broke into it, so still not much secure.

Crypto System

Three basic structures:
Symmetric cryptosystem:
M=D(K,E(K,M))

M= message
K=key
E(K, M): gives Cipher Text
D(K,E(K,M)):gives decrypted message of ciphertext
The same key is used to encrypt and decrypt the message.

Asymmetric cryptosystem

$M=D(K_d,E(K_e,M))$:

$K_d != K_e$

Keys are generated by using key generating algo. These are not random keys. They are key-pair but are different keys.

Hashing: No keys are there. Used in password storing, digital signature
C=>H(M)
From H(M), we cannot get M . It is a lossy conversion. One way translation.

Implementation Point of view for **ASymmetric cryptosystem &Symmetric cryptosystem:**

Both enforce confidentiality and are safe.

Differences exist in key management. Includes key generation, key exchange, key replacement.

Asymmetric crypto is difficult for key management and overhead more and provides the same level of security.

Problem how do Alice and Bob exchange information without Arun (hacker) knowing about the info
Suppose symmetric key crypto is implemented.

info
ALice←----------->Bob
(K) Arun (K)
(attacker)

Step1] Ensure Alice and Bob must have the same key K without Arun knowing.
Step2] we can do a physical exchange of keys.(not efficient)
Solution: Diffe hellman Key exchange

What everyone knows: mod key P (prime number), base g

Alice has its secret key ‘a’: $A = g^a \bmod P$

Bob has its secret key ‘b’ : $B = g^b \bmod P$

Now we exchange A, B

Now Arun knows: A,B,P,g

Alice : $B^a \bmod P = (g^a \bmod P)^b \bmod p$

Bob: $A^b \bmod P = (g^b \bmod P)^a \bmod p$

Hence both arrive at the same key. This is Modulo exponential algo, Discrete exponentiation Problem.

Again Problem :

Arun impersonates Bob to Alice and performs Diffie Hellman key exchange and got to know the key.

Again arun does with Bob by impersonating Alice. So A, B are exchanged via Arun.

This attack is Man in the middle attack (MITM).

Limitation of Diffie Hellman: enable secure key exchange between two parties without authentication.

Prone to MITM.

Lecture 7

Assume key management is handled :

1] Either key is distributed offline

Or 2] shared key is assumed

Or 3] use other mechanisms for setting up shared key

Symmetric key crypto is secured based on the above assumptions.
Symmetric key crypto:

1] Block Cipher:

One complete block of size 668 bits

|||

Divided into multiple block.

128 bit	**128bit**	**128bit**	**128bit**	**100bit+padding**

Plain Text= "Roberto"(7bytes)
If it is 128 bits (16 bytes)
We can pad extra bits 9 padding character to 'Roberto'
Plain Text :'RobertoXXXXXXXXX' ,where X is padding character.

|||

Block Cipher

|||

Cypher text

Block cipher modes:

1] Electronic code book(simplest mode):

For every, i belongs 1 to n (parallel execution) each cipher is independent :

P(i) is each block.

P(i)=> $C[i]=E_k(P[i])$

C(i)=> $P[i]=D_k(C[i])$

Strengths:

A] very simple

B]Allows parallel encryption - fast

C]Tolerate damage to blocks, resend and do again not much overhead

Weakness:

A] extremely vulnerable to pattern matching algorithm => not compatible for images & doc

2] Cipher Block Chain(CBC)

$C[i]=E_k(C[i-1] \text{ Xor } P[i])$

$P[0]$ with key (K) give $C[0]$

$C[1]=E_k(C[0] \text{ Xor } P[1])$

$C[2]=E_k(C[1] \text{ Xor } P[2])$

.

.

.

This will make pattern recognition difficult.

Idea: Convert one cypher block into chain blocks.

Strengths:

1]Do not show patterns

2]Most commonly used

Weakness:

1] requiresreliable transmission

2] Not suitable for application that allows losses.util complete transmission not completed , decryption not possible.

3] only sequential block transfer . dependency is present.

Block Ciphers in practice:

Data encryption standard(DES)

->1977 build

->56 bit key,64 bit is block size

->secured util 90's

Triple DES

-> nested application of DES K_a,K_b,K_c

->Total /effective key standard=165 bit

->$C=E_{Ka}(D_{Kb}(E_{Kc}(P)))$ [Encryption]

$P= D_{Ka}(E_{Kb}(D_{Kc}(C)))$ [Decryption]

AES(Advanced Encryption Standard)

->started use in 2001

->128 bit,198 bit,256 bit

Current standard AES-256bit

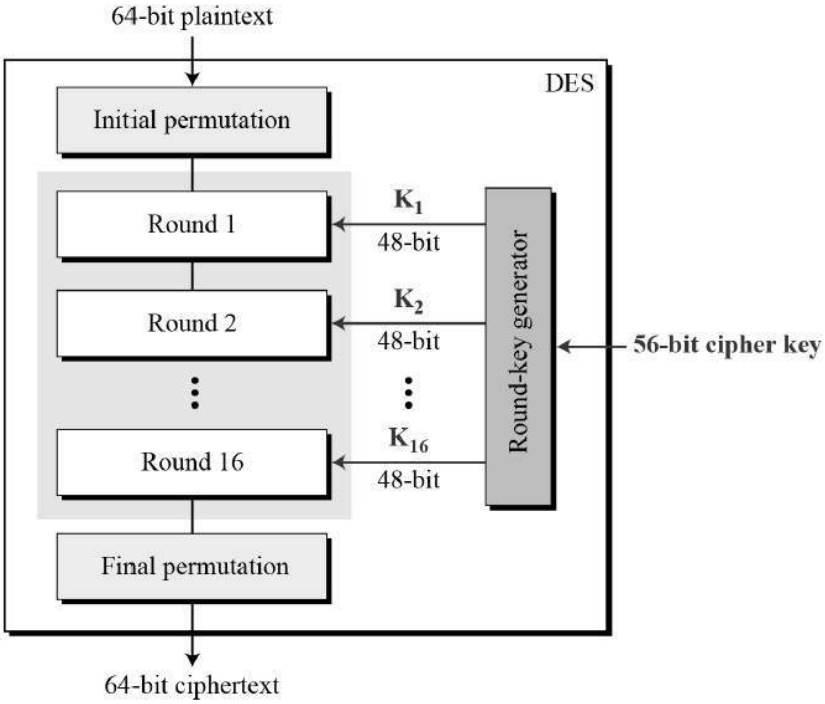
Overhead is directly proportional to **the length of the key**

DES

64-bit input

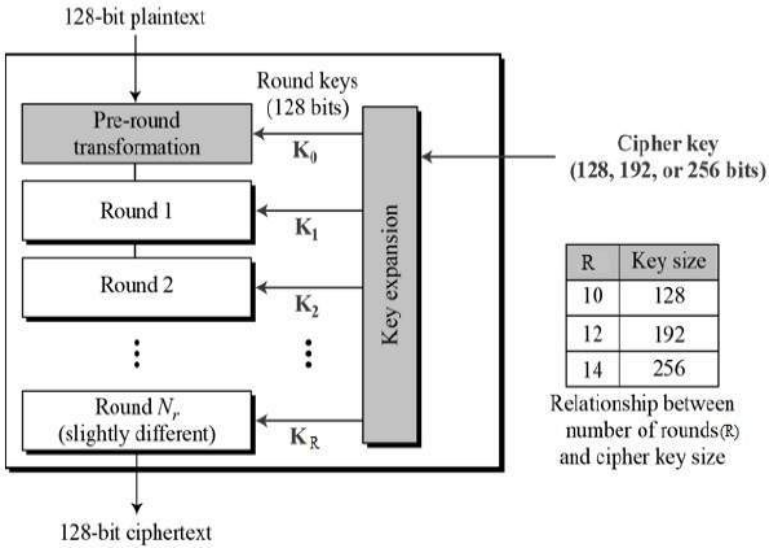
16 identical rounds with the application of a 48-bit key

Shift L1, R1 apply 8 bits of (L,R, K) we get L2, R2 and repeated 16 times



Limitation of DES : the low key size of 56 bit

AES



R	Key size
10	128
12	192
14	256

Relationship between number of rounds(R) and cipher key size

Steps performed in each round:

1] S-box: substitution Step
2] P- box:Permutaiton step
3] Matrix Multiplication
4] Xor Step
Key size can be :128,192,256,512 bits
Faster compared to an asymmetric key

2]stream Cipher

_____The pseudo-Random sequence of bits continuously

LECTURE 8

Hashing: Given a plain text p, if we have hashing function h, we do h(p) we get a hash value, and this is a one-way process, we cannot take h and get back p, it is irreversible. No matter what the size of p is, we will get a fixed hash value. It depends on an algorithm we use, like 128 bits, 256 bits etc. No matter how many times we do it, h(p) will be one to one.

Properties: Easy to compute, strong one-way function, resistance to hash collisions. It means it is hard to find p and q, such that h(p)=h(q). So we don't want this to be a case, we want difficulty in order to find p,q such that h(p)=h(q). MD5 is the most common hash algo, 128-bit digest, SHA1 give 160-bit digest, SHA-2 gives 256-bit digest, it is standard, for extra security, we have SHA-3, we can also use that for our course project. 256-bit digest, collision possibility is rare, like we need 1000 years to know what is p,q such that h(p)=h(q). SHA-3 is a 512-bit message digest.

Q. What could be a simple hash function?

Ans. Modulo can be used as a good hash function. h(x) = xmod10, but poor resistance to collision, but it is one way and easy to compute.

Rolling Hash function: https://en.wikipedia.org/wiki/Rolling_hash
Iterated Hash function: SHA, MD5

Asymmetric Key Cryptography (AKC)
In Symmetric Key Crypto, key management was done. If we lose the assumption of key management, can symmetric key Cryptography work? No!!

In Asymmetric Key Cryptography, we have 2 keys one key for encryption and one for decryption. In almost 99% of cases, we have the same encryption and decryption algorithms are same. Just that, we use different keys for encryption and decryption.
P------E-------->C
C--------D------->P
E=D

Alice has plain text, she encrypts the plain text into Cipher Text and sends it to Bob.
$E_{K_B}(P)$ and sends to Bob.

K_B^+: $Bob's\ public\ key\ and\ K_B^-$ is Bob's private key

$D_{K_B}^-(C) \rightarrow P$: $Bob\ decrypts\ the\ cipher\ text\ using\ its\ private\ key.$

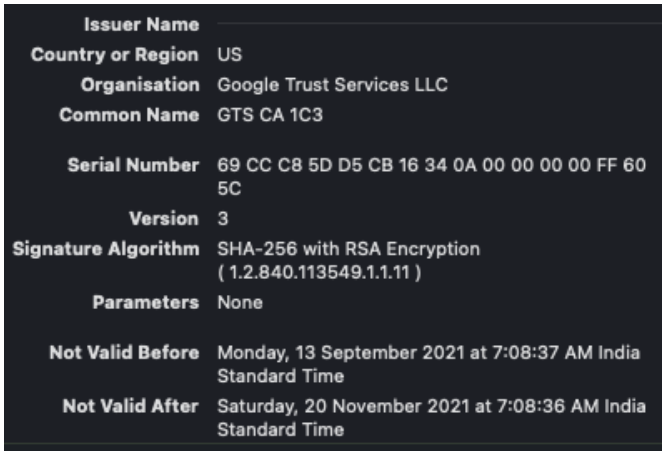
The assumption for AKC :
Bob's Public key is known to Alice
Asymmetric key cryptography is not safe, the same problem is there with Symmetric-key Cryptography, that is the man in the middle attack. Someone in the middle can spoof. How will this particular framework solve it? If they can talk directly to each other then what is the purpose of AKC?
We need other structures to solve this problem. No mechanism to verify the public key, whether it is the public key of a legitimate user or attacker.
In order to understand AKC, we should know about GCD. Two integers a,b are coprime if gcd(a,b)=1
We can use Euclid GCD to effectively calculate the GCD and then Extended Euclid Algorithm in order to get a,b given ax+by = gcd(x,y)

RSA:
Chose any two large prime numbers, p and q (1024 bits) are private.
Compute n=pq and z=(p-1)(q-1) and n is public

Chose e<n such that gcd(e,z)=1
Chose d such that e*d modz=1
The public key becomes (n,e) and the private key is (n,d)

Encrypt: $c = m^e modn$
Decrypt: $m = c^d modn = (m^e modn)^d modn$

Lecture 10



Certificates

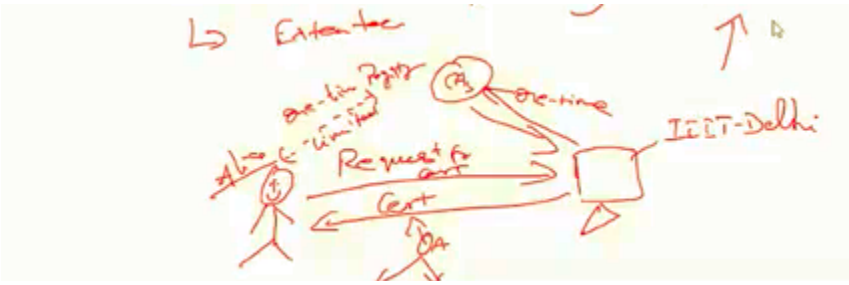
Root certificate Authorities are entities that the browsers trust. If a certificate is verified, then who is verifying it? It is the browser. The browser maintains a list of certificate authorities, it has a public key of all root certificates authorities eg GoDaddy, Digicert etc. We have to apply for a certificate through the website of one of these companies, we will get a certificate based on the application. We have many kinds of certificates. There are many levels of certificates. One is **domain validation, organisation validation** and **extended validation** certificate. DV certificate is least costly, it is binding of a public key with a domain name, OV needs background checks etc also, EV certificate requires government level permissions etc. Based on this we get a single, wildcard or multi-domain certificate, Single means we get the certificate for your domain and that is it. The wildcard means we get a Single domain certificate and also that it enables you to provide additional certificates for your sub-domains. Multi-domain means you can use the authorities of the CAs to actually provide certificates for multiple domains. IIIT D has a website, there are many subdomains within it. IIIT D should have a wildcard certificate for it to certify other sub-domains within it. So, these are the classes of certificates and based on the level of certificates you got, they will kind of enable you to either certificate provider or single user certificate.

Why it is not easy to go back to the Certificate provider when you need the certificate every time. Every time Alice wants to contact IIITD, it will request a certificate, and IIIT D will send the certificate, and Alice Browser verifies IIITD's certificate is valid or not. Alice does not need to communicate with the CA. This communication is one time. Alice needs to get the CA

information, for which it needs to communicate with CA. There will be a registry that Alice downloads that contain a list of CA's. This is the one-time communication that is done by Alice.

Q. Once a website has a wildcard certificate, it can host a subdomain? Yes.
Q. Whenever we host the IIITD website, we do need the certificate for every subdomain also.? Yes
Someone has to issue the certificate for every Subdomain. Once IIITD as an organisation got the certificate, then IIITD itself can issue the certificate to its subdomains. We have to check up the chain for verification that goes in the back.
Every time we need to go to CA, then it can become the bottleneck, and the internet does not work like this, therefore, once we contact a service provider, then we contact them one time, CA give certificates to IIITD when IIITD communicates with the CA.

Q. One attacker visit IIITD, it has a certificate that was issued by some CA, he does a man in the middle attack with this CA.? How to deal with this?
A. Attacker does not have a private key of IIITD, then it doesn't hamper anything.



CA communicates with a central registry, duplicate certificates cant be issued, CA will verify the domain and the public key is valid or not.

Network Basics/Security Concerns

Jo ki humein aata hai.
Slides kafi hain iske lie.

Lecture 11: 21 September 2021

Syllabus: Until Friday 24/9/21

Revision→ Boring

DDOS: DOS is not coming from one particular place, it is distributed across many geo locations so that we are not able to ascertain what patterns there are to prevent the attacks. Of Course, there are many prevention mechanisms like DNS sec and IP sec. They place additional requirements towards many of these systems.

Sometimes it is possible to do, sometimes it is not

Source Spoofing
Replay attacks
Confidentiality Integrity based attacks→ happen because there is no confidentiality and integrity in IP.
DOS/DDOS/ replay attacks/ spying attacks and many more
These are central to the protocol

IPV6 attacks
If we are really able to make something end to end then we are actually able to do away with spoofing attacks.
IPV6 mandated IPSec because we are able to do end to end
IPSec is itself very secure, it is IP level security

It kind of protects integrity as well as confidentiality. Everything is encrypted and secured.
It required IPV6, IPSec is still possible with IPV4, but compatibility issues, MAC level compatibility is required. MAC is not compatible with IPV6.
It is not possible as of now.

Gateway, we have internet, we have A, B, we have communication links. End to End means A to B using TCP. Currently, the addressing is not End to end.
We kind of have a gateway here, and If A wants to communicate with B, then it puts its Destination Address here. IPV4 is not meant for end to end. IPV6 can be end to end, we

can implement IPSec using IPV6, it uses IKE, internet key exchange, it is end to end encrypted. It is a mechanism through which we have different layers like PL, LL, NL, TL, SL, PL, AL. If we use IPSec, everything bove is encrypted. So that no one can change the IP Address if it is encapsulated. Encryption and integrity checks are at the application layer. IPSec put everything above NL, encrypts it and authenticate it.
TLS, public and asymmetric key cryptography to establish keys, to do network-layer encryption.
Two modes of operation
Transport mode: Between A to A
Tunnel Mode: A to B for VPN's

It is TSL equivalent to NL, it is not application level. No device in between cannot change your IP Address. MAC is required to change IP, if we change, we can spoof. But we can't tamper with these things.

End to End encryption: only two parties can communicate with each other securely using NL and the above layer. IKE has two phases, phase 2, application-level or session-level, custom secure channels. . for each service different session.

1. End to end secure channel (use DH algo) (device to device)
2. Negotiate and establish custom security channels. (use DH algo)

This is a broader level device to device or end to end security channel. Once we establish the sec channel, we can use it to set custom channels.
This we can think of like a session key channel.
IKE requires AKC(Asymmetric Key Cryptography)
Application to Network

The IP address is authenticated first.
IPV6 is used and implemented in many ways, for organisational level exchange. Like google servers in two-three places, they use IPV6 and IPSec and make it more secure. We still rely on Application-level security.

IPV6 is different, Application security is different, and IPSec is different.
IKE is a service, it is a third party internet key exchange, we can set this exchange service at the organisational level also.

RFC 2409→ refer for comments for IPSec (Jo hume nhi padhna h)

IPSec protocol says we cannot change IP addresses.
We have enough addressing space for all devices, we can stop the devices if we want, to stop them from transitioning.

Firewalls
Packet Filtering Firewalls
Proxy Server Firewalls

Two categories of Firewalls.
Why do we need a firewall?
A firewall is one of the most important network functions. The complexity comes into place when we don't configure it properly. Many people use tools to configure firewalls, we should not use tools, we should configure them using ourselves.
IP Tables/ NF tables are integrated into the kernel, execute certain commands to implement firewalls into our system.
There are three chains

1. Input chain
2. Forward chain
3. Output chain

Each IP table supports four tables

1. Filter
2. Mangle
3. NAT
4. Raw

In most cases, we will be using Filter Table. Otherwise, mangle. Each table has chains.
Input chain is what can come into the system. If we don't want something to come into the system, we can drop it. I want to communicate with 192.168.0.1 only, I don't want anyone to ping my system, we can enable this in the firewall, through the input chain, and once we do that, no other system can discover us.

Forward Chains- You get this packet, then you transfer this to this particular IP Address, used for proxy and other servers
Output: Whom can you communicate with, where can you send this, enforce policy with respect to that, enforce that particular rule. Output is something that is coming from our system.

Ip Table rules are not that difficult, extremely simple.
If we want to kind of use to, we have to be extremely careful.

Firewall: Packet Filter Firewall and Proxy server Firewall

Packet Filter Firewall

1. **Router Packet Filtering:** Single packet attacks are captured, it is stateless

The packet header is only one, only single packet attacks can be captured. We cant link the first and the second packets, only the packet header is inspected. Low overhead.

2. **Stateful Inspection**: More fields in the packet is inspected. Here the state is retained. Since the state is retained, multi-packet attacks can be captured, which has more overhead. Assign more resources. Much slower.

Proxy Server

1. **Circuit level Firewall:** A and B and we have a firewall, the firewall takes the packet from A, it kinds of terminates the connection, create a connection to B and then send it to B. Firewall does not allow us to directly communicate A with the host B. packet sessions are terminated and are recreated through proxy. Multi-packets attacks are captured. Very high overhead. This is the difference between packet filtering and proxy level firewall.
2. **Application-level Firewall**: Same except that we have more inspection.

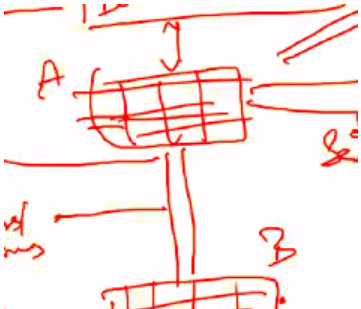
Design a firewall for IIIT Delhi.

Lecture 12

Recap: Firewalls, types of tables, input, output and forward chains etc. A firewall is very simple, and how it is built into the kernels, and we have to directly work with NFT tables to kind of enforce the firewalls, we have applications that also do this, so the complexity in firewalls is not in firewalls, it is in the configuration of the firewall there is n number of ways to configure the firewalls. It is checking what we want to do to filter what the traffic you are seeing, what you want to check and at what intent you want to check, checks are at different levels, do it on application and network level, generally, they are network-level firewalls.

Designing a firewall at IIIT D

Wifi, Labs, Faculty terminals, student registration and health services, we have servers, we have a bunch of classes of system, so how do we restrict traffic depending on the necessities, each of the systems has different requirements. We have a border router. Firewall can be infrared facing or outward facing firewall, it depends on how we want to enforce these rules. So generally when you have a bunch of systems having diverse requirements and not all groups of resources need to be very restrictive, then we don't have one firewall, because if we have one firewall then we cannot frame rules that effectively for one firewall. We have an organisation having different groups of resources, then we have multiple firewalls, we have two firewalls, to actually manage two separate requirements, and if we have one firewall and make it very restrictive then it becomes so much restrictive, so we use two firewalls to balance the restrictions and utilities, we have this type of firewall which is called as dual inline firewall, and we can have a firewall in parallel, in series and whichever configuration we want to have.



Firewall A is a little bit less restrictive it need not be that secure, firewall B is very much secure not everyone can access it,

unfortunately, this firewall cannot be implemented at the first level, as it becomes too much restrictive, there are BlackBox and Whitebox kind of firewall, in firewall B we need whitelisting, we need to mention what kind of systems can access these resources, in the black listing, we can mention what kind of resources cannot use. Try to make the rules very simple, someone has to make sense of the rules, otherwise, it would be quite complex, the reason we use two firewalls is because of the defence and depth principle in security, this is one of the cardinal principles of security, even if one part of the system fails, it does not let the whole system become vulnerable. That is why when we see most of the cases in our university, we will see dual inline firewalls, and we can also put in A and B in parallel, what are the rules, how to design the rules, outward-facing inward-facing, types of firewalls, packet filtering, proxy server firewall. A is outward-facing it pays more attention to what is coming inside the system. B is an inward-facing firewall it pays more attention to what is going out of the firewall. These are network devices. We are not inspecting application-level data, rather a decision is made on whether I can trust this data or cannot trust this data. It depends on where are you making connections to, you are not paying attention to how you are restricting the network once you come into the network. Firewall B is concerned with what system I am connecting to. It is very relevant when we are designing the rule of the firewalls, the nature of the design depends on the user which is sitting inside the firewall. Terminals have fairly stable IP connections, they are not dynamic. Internal to the network is a locally assigned IP, it is not public-facing IP. The dynamic part is controlled by the internal network, so it knows.
How we access ERP using FortiClient. When we talk about VBC, it is related to VPN or not?
For connecting VPN, we need credentials.

Privacy
It is an ability of an individual or a group to prevent their identity or personal information from being known to people other than legitimate users.
Why What and How
How do we enforce privacy from a technical perspective?
It is very very important to understand what is privacy and why it is required to understand what to do and what not to do.
Anonymisation is related to privacy.
K-Anonymity is used to enforce privacy.

Q. Why is privacy important?

1. User demand and that is why it is very important.
2. Safety, security, protection
3. Regulations/Laws/Government Demands Article 16, IPC, International Laws also

Once we develop a system/piece of code, we need to be aware of how can we end up in trouble by keeping into consideration all the laws etc.

Threats for Privacy

1. Application-level: Large transfer of data, cross-correlate.
2. Communication level: Anonymity of sender/receiver
3. System-level: How to gain system-level details, unauth access to the system
4. Audit level: Logs to compromise privacy./ audit files to compromise privacy.

Actors

1. Government threats
2. Terrorism related concerns
3. Aggregation threats and data mining

Regulators: Law enforcement agencies LEA's, there are MLA/MP called as legislatures, and there are a bunch of other executives, but LEA are the one's who might have access to personal information under few circumstances. They can access data. The warrant has to go through the judiciary/Magistrate, based on which they can access your data, Voluntary access to the right of privacy to evade.

Q. Organisations need to collect data for personal benefits.
A.

K Anonymity
Suppression and Generalisation are two principles that they use

Disease	Gender	Age	ZIP
Heart	M	30	123456
Heart	M	33	123456
Diab	M	45	345677
HepA	F	42	345683

C1 is sensitive/confidential, C2,C3,C4 are not sensitive.
k=2, two anonymity, what we basically do is we build the table

Disease	Gender	Age	Zip
H	M	3*	123456
H	M	3*	123456
D	*	4*	3456*
H	*	4*	3456*

Builder clusters of rows with a minimum of size 2, at least 2 i.e. at least k. In those clusters, all the data has to be identical. That is the rule of anonymity. It has to have a min of two rows
We generalised 30,33 into one, we made it 3*

* is suppression
3* is a generalisation

What can happen if we increase the number of k when we increase the anonymity.
2 anonymity vs 4 anonymity?
Privacy increases when k increases, but utility decreases. We might have to do more suppression and generalisation, that is the tradeoff that the 3rd party user has to the contempt with, k-anonymity in its core is very simple. We can use it in million different ways. Another parameter is in the way in which we group things. We can do this grouping in a bunch of different ways, we can use clustering algorithms to see what rows are similar and if we do that what happens. It is used by almost all major companies which are used today, Differential privacy is another class of privacy it is the improvement of k-anonymity, it offers better privacy and utility, offers better in terms of videos and audios. Facebook uses k-anonymity.

Lecture 13

What is privacy, why is it important and certain techniques to enforce privacy like k-anonymity?
Clustering brings rows together, to make each of the clusters the same would be reduced.
Transactions done within the country will remain inside the country and no one can do anything in that regard. Whatever data is being generated within the country, it will only be consumed within the country and this is data localisation and have a copy for jurisdictional purposes. Data localisation is asking the service provider that as law enforcement agencies has jurisdiction above this data so it should be stayed in the country itself and should not be moved. The country has to have laws and legal framework. Privacy is a big area. Anonymity is the way to ensure privacy. Anonymity contradicts accountability but we need accountability on certain aspects of the things.
K-anonymity has a lot of limitations, ends up removing lots of data, if we are conservative and liberal, we work on one data set and not many datasets, in most cases, if we anonymise the dataset, in future, we release a new data and cross-correlate and de-anonymise then we can attack, there are no ways as we don't have any knowledge about these datasets, across multiple organisations. There can be cross-correlation attacks on voter id and Facebook posts, K-anonymity won't be able to address it.
GDPR has made privacy user-centric. Users can sue us if their anonymity is lost. We cannot sell it like today it is anonymous. Tomorrow it can be lost. Some of the service providers are studying that take raw data and send it through some technique consuming data, understanding the distribution, generate new data with similar distribution, the generated data is different from raw data, generic data is synthetic, it is not related to the raw dataset. This space of anonymisation, it is known as differential privacy, here we train machine learning models, which consumes a lot of this datasets and generate new data set. In this way, we can say that we have given to synthetic dataset. It resembles a lot to the raw data, the service

providers are safe in that we are not sharing actual data, in differential privacy, we are sharing synthetic records but not the real data.

Using TOR, still, does not guarantee anonymity, but we can achieve anonymity to a larger extent. Service providers, technologies of anonymisation and payment gateway are the three pillars that can compromise anonymity. We need anonymity and there are many cases when we don't need anonymity and there are some places where we do want anonymity. The adversarial model is complicated, the recipient may want to anonymise you and the sender may want to de anonymise. There are many people who really want to know your anonymity. There are many anonymous tools on the internet, we will use TOR systems. Tor is a service, it can be installed at many places. It maintains anonymity within the network. We should have three nodes, entry, middle and exit nodes. OR is the onion routers, there is something called a directory and directory has a list of all the onion routers, tor client builds the circuit of these OR's and this selection is random and there can be at least 1 middle OR and each OR maintain a TLS connection with other routers.
Anyone can become an entry and exit node, we can become OR, we just need good bandwidth to become the OR. the TOR network is vastly decentralised and democratised. To become an entry and exit node we need to satisfy certain criteria.
Everyone identity must be protected. Service provider anonymity must be protected from users also. The server should also be anonymized. Let us suppose I am hosting a website on the IIIT Delhi server and I must maintain my anonymity. There are ways and methods over which we can provide a service without having any static ip address, we have something called onion URLs, there are randomly generated and we can list these randomly generated URLs so that someone can access these services without the URLs, the service provider can retain the anonymity if he/she wants.
We can do spoofing and fishing attack on the routers, we also have to understand that the law enforcement agencies also use TOR, as it is unregulated therefore all kinds of things can happen. If we have legal and legitimate servers, use those, it is really very difficult to maintain anonymity and to guarantee services, there are other things, which stops us from operating, unfortunately, there is something known as the dark web as there are no regulations. There are certain service providers. We use digital currency to retain anonymity. The service provider access bitcoins. No one can track it. If you transact in bitcoins, no one can know that it is you. TOR nodes and onion routers maintain secure communication with each other. This is how message encapsulation works. Using TOR client we encrypt the message with three layers of encryption, first layer using asymmetric key cryptography is encryption of public key of exit node, then the public key of middle node and public key of the entry node. And then the whole message is sent to the entry node. Entry node now has it has its private key and decrypt the message using that and send it to the middle node, and it decrypts using its private key and then it sends this to the exit node, and then the exit node decrypts using its private key, the entry node knows user and middle, no node in tor has no more than one hop information.
Directory service store public key and entry node, exit node etc. All these three nodes are like courier service, they cannot know as it is encrypted, when the package reaches the exit node, the exit node communicates with the service, the user maintains anonymity from the service provider.

Each Onion router maintains a TLS connection with all the other OR's. OP stands for Onion Proxy. The high-level idea behind TOR is to transmit user data through a bunch of nodes in between such that each node has no information about where the packet is coming from and where it is going. To achieve this

functionality, TOR uses onion routing. In onion routing, messages are enclosed in layers of encryption called Onions due to their layered architecture. Each node in this network is called an onion router and each OR is responsible to peel away a single layer(decryption) uncovering the data for the next node in the network.
Before communicating with the server, a circuit needs to be established between the host and the client. It is established via Onion Proxy (OP), it is a local software and it runs on the client's machine which uses SOCKS. TOR has a directory of approved OR's along with their public keys. The OP picks the OR's randomly in a sequence by downloading the list of available OR's from the directory and using their public keys to form the circuit. The first OR in the circuit is called the entry guard and the last OR which decrypts the final layer is called the exit node. The current design of TOR uses 3 OR's and onion routers communicate with each other and OP via TLS connections with temporary keys.
To create a circuit, OP picks secret key x1 and then it sends a create operation to the entry guard, OR1 and they do a DH key exchange as follows

- OP sends "c_1, $Enc_{pk_1}(g^{x_1})$"
- OR_1 chooses random y_1 and replies with "created c_1, g^{y_1}"
- OP computes $k_1 = g^{x_1 y_1}$

Here amazon.com does not know where is the origin connection from, it does not know anything, for amazon, it looks like the connection is coming from the Netherlands. The entry node is the guard node, middle node and exit node, these are the websites that are publically accessible. If you know the URL, you can access it, 95% of websites that are not publicly available, are known as the dark net. What happens if amazon.com wants to be anonymous. Using TOR, the buyer, seller or service provider can remain anonymous from each other.
If we want to remain hidden from everyone you can do so, how will you enforce it? That is an interesting question, the user needs to know the address where you can reach them. You want other's to know but you don't want the government to know, how will you ensure it? We can host the server on the TOR network, then normal search engines won't be able to index it, and why do we need to host it on the TOR network, if we host it on the normal network, it will reveal the identity of the server? This is wrong

Tor gives you the platform, it does not give you the ability to do something. There is a difference between getting access through only the TOR network and hosting on the TOR network.
The website is hidden behind three proxy servers. They will make a list of exit node IP addresses and they will whitelist it, you can access the IP addresses from one of the exit nodes because exit nodes are available online, we can download it, and we won't be able to access these websites from any random browser, it is extremely hard to track these guys down.
These URLs keep on changing, so there is no reason to track them. Let us say the government blocks Facebook from India, and how will you access Facebook, so Facebook has an onion site, then we will be able to access the website from TOR. Facebook has a stable IP address for us to reach them, the only way to ban them is to ban/shut down the entire internet in the country or ban the IP addresses on the TOR directory. That is what CHINA does. We won't be able to establish any communication.
We basically monitor all the communication that is coming in or going out of the country, this is the notion of the huge firewall. All of the ISP's are mandated by the government. They are the ones who basically have to backbone. All the network-related devices are in their control. They have ISP .

The government is not able to shut those down because these are dynamic proxies that they are using.

Lecture 14

Tor client ⟷ Entry node ⟷ Middle node⟷ Exit node⟷ Server

We have the directory services. TOR client gets IP addresses and associated public keys as well,
$E_{entry}(E_{middle}(E_{exit}(m)))$ *this message usually happens with HTTPS*
Entry node decrypts and it can only see $E_{middle}(E_{exit}(m))$ and middle node see the message and decrypts it and it sees $E_{exit}(m)$and then it sends it to the exit node and then the exit node can see the message and decrypt it.
ISP only knows only one hop distance information.
The main strength of TOR is also its weakness, it is very open, anyone can maintain it, if we have enough resources and bandwidth, we can become entry nodes and exit nodes also. Malicious actors can also become entry nodes and exit nodes.

Threats to TOR

1. **Traffic analysis**: If malicious actors control entry and exit nodes, then it is possible to do a timing or correlation attack. It is basically monitoring the traffic and looking at the size of the packets and correlating from the entry node and the exit node what and how close or different a particular traffic pattern is. Or we can maliciously inject traffic to the particular packet in the entry node and exit node and see how the traffic pattern is propagating within the network and through that, we can deanonymise the particular communication to a certain extent. This requires access to entry and exit nodes.
2. **DNS Leak:** When we are using TOR, we have to be extremely careful, if we do not use DNS services provided by TOR then ISP will know which servers that we are reaching out because the DNS resolver will give this info, so there are certain proxies which are within the TOR and they provide us DNS within the TOR like Privoxy and Foxy Proxy so that no one is able to track what DNS addresses are being resolved.
3. **Rouge Exit Nodes:** If the communication over TOR is unencrypted then we can use rouge exit nodes to basically get the entire information out, that is why it is really important to use HTTPS. Because we never know which exit node we get is random, we can get malicious exit node also na, that is why HTTPS is required.

Web Basics and Security Concerns

Why web browsers, web application, HTTPS, TLS protocol, how JS is used to build/launch attacks, what is HTTP req, res, how content is built using HTTP, how to exploit/launch an attack on DOM model, we will look at Cross-Site Scripting CSS and cross-site request forgery or CSRF and then code injection.

Q. If we have a VPN which is owned by a private company, then how is it different from TOR?
The difference between VPN and TOR is we are not anonymous to Third-party (Service Provider). No log policy of VPN is a marketing gimmick. The mandate for VPN is not for anonymity. VPN is not

primarily built for anonymity purposes, it is very brittle and it is not as good as TOR is. VPN is just used to prevent the collection of data/ or maintaining privacy, and VPN is managed by someone, a decentralised VPN is good, but VPN is not decentralized nowadays, there must be a centralised authority that is logging your data. VPN is much faster, provide you with security but it is not decentralised.

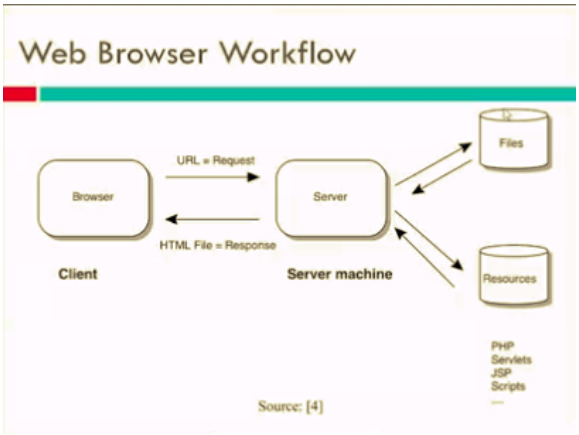
Q. How can I host an onion site using TOR?
We will have to host the server, and it is behind multiple proxies and we can buy them using TOR network, and we can basically ensure all the connections are through TOR i.e. whitelisting all the IP addresses.

Q Do I need to send certificates using TOR?
There is a way, as we have to make inions compatible with how it is hosted, there are no certificates that are needed from CA's, self-censored certificates would suffice.

Q. How are Onion sites get DNS resolved?
TOR has their own services for onion sites, when we are looking at these listing services then we are doing initial resolution if they have this info with the listing services, if the info is not available, then they would use a separate service hosted within the TOR network to resolve those addresses. And remember these URLS are very dynamic in nature, they may not be in life for a short period, and keep on changing.

Lecture 15

We will look at web basics from a security point of view. When we say web, what comes to mind?
In the 2000s we tried to make browsers more secure, the whole paradigm shifted to apps, it was more complicated in dealing with the security implications of apps themselves. Browers vs alls is a whole different ballgame.
Browsers, we have few vendors like MS, Google, Mozilla, Safari, it is very easy to regulate these browsers, apps come to many different vendors, that is why it is very difficult for people to ensure to standardise app development models even though recently android app store and the apple app store actually have enforced security and regulations and static and dynamic checks, but it is not that safe still. Now and then, we hear certain apps getting pulled down from app stores even though we do some checks. Several threats and attacks are like cross-site scripting, cross-site request forgery and code injection. Average users spend 70 hours in a month online, Users spend a significant amount of time, lack of security has a major impact on these users this is not just stealing of data, it also involves observing and modelling and profiling user behaviour.



We have 7 layers, physical layer, data link layer, network layer, transport layer, presentation layer, session layer and applicatio layer. At the PL, we have Etherloop, DSL(Digital Subscriber Line), 802.11 WIFI, Token Ring etc . At data link layer, we have ARP, RARP, BOOTP,PPP, etc. At Network layer we have IP, ICMP, IGMP, NAT, IPSec etc. At Transport layer, we have TCP, UDP, TSL, DCCP, AH(Authentication header). At layer 7, we have FTP, DNS, SMTP, HTTP, HTTPS,NFS, etc.

We have to look at the protocol and vulnerability within the protocol to know what is wrong with the system.

Document Object model is really very important as we can manipulate the DOM object if we have the script with thm, we acan create, append, edit, change, remove the objects on top of HTML, which we may/ may not apparently see. JS over HTML is important wrt security paradigm.