
C|EH
Certified Ethical Hacker
Module 12
Evading IDS, Firewalls, and Honeypots

Module Objectives

Module Objectives

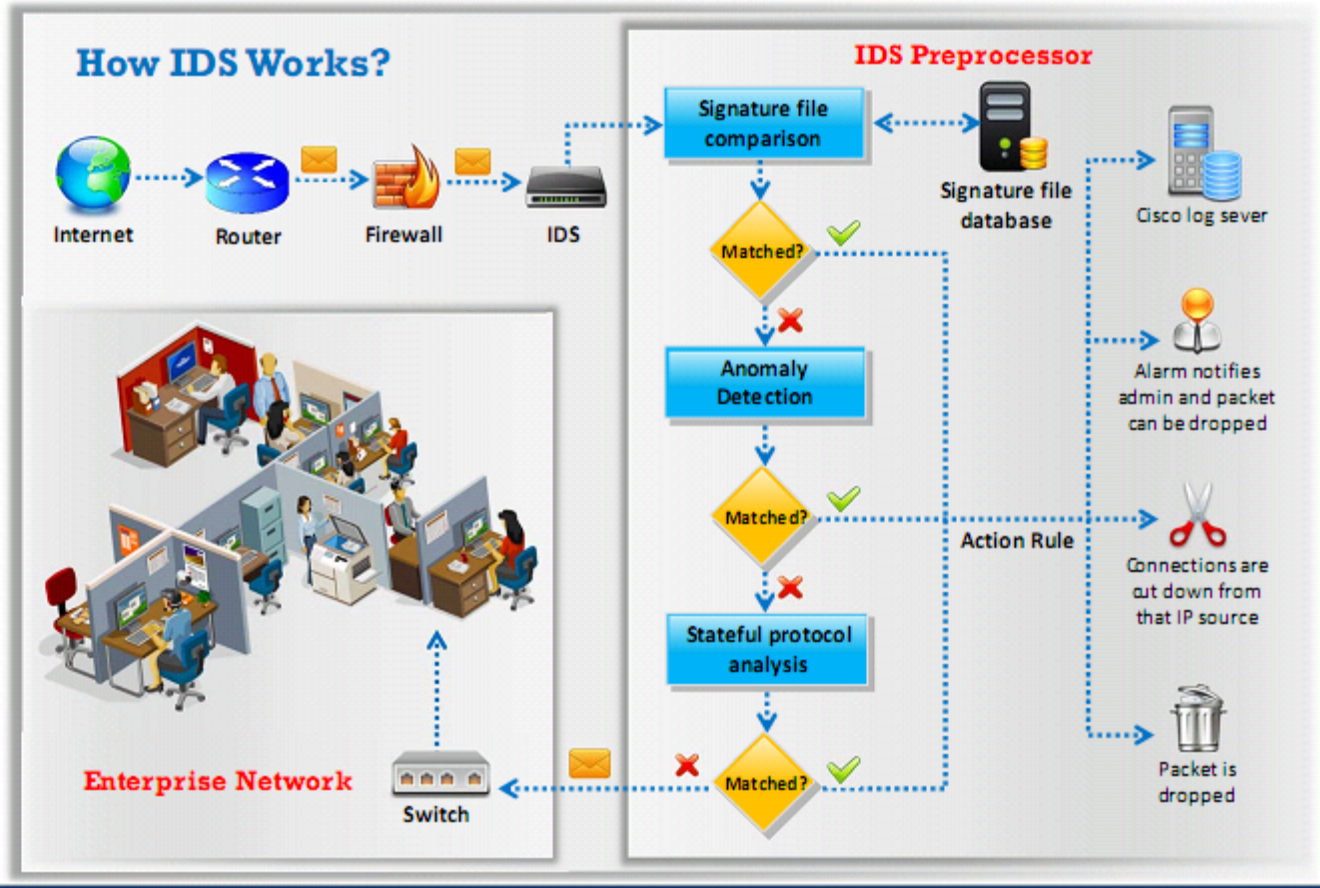
- Understanding IDS, Firewall, and Honeypot Concepts
- IDS, Firewall and Honeypot Solutions
- Understanding different Techniques to Bypass IDS
- Understanding different Techniques to Bypass Firewalls
- IDS/Firewall Evading Tools
- Understanding different Techniques to Detect Honeypots
- IDS/Firewall Evasion Countermeasures
- Overview of IDS and Firewall Penetration Testing

Module Flow

1. IDS, Firewall and Honeypot Concepts
2. IDS, Firewall and Honeypot Solutions
3. Evading IDS
4. Evading Firewalls
5. IDS/Firewall Evading Tools
6. Detecting Honeypots
7. IDS/Firewall Evasion Countermeasures
8. Penetration Testing

Intrusion Detection System (IDS)

- An intrusion detection system (IDS) is a security software or hardware device which **inspects all inbound and outbound network traffic** for suspicious patterns that may indicate a network or system security breach
- The IDS **checks traffic** for signatures that match known intrusion patterns, and **signals an alarm** when a match is found
- Depending on the traffic to be monitored, the IDS is placed **outside/inside the firewall** to monitor suspicious traffic originating from outside/inside the network



How IDS Detects an Intrusion

Signature Recognition	Signature recognition, also known as misuse detection, tries to **identify events** that indicate an abuse of a system or network resource
Anomaly Detection	It detects the **intrusion based** on the fixed behavioral characteristics of the users and components in a computer system
Protocol Anomaly Detection	In this type of detection, models are built to explore **anomalies** in the way vendors deploy the **TCP/IP specification**

General Indications of Intrusions

File System Intrusions

- The presence of new, **unfamiliar files**, or programs
- Changes in **file permissions**
- Unexplained changes in a file's **size**
- **Rogue files** on the system that do not correspond to your master list of signed files
- Missing files

Network Intrusions

- **Repeated probes** of the available services on your machines
- Connections from **unusual locations**
- Repeated login attempts from **remote hosts**
- Sudden **influx of log data**

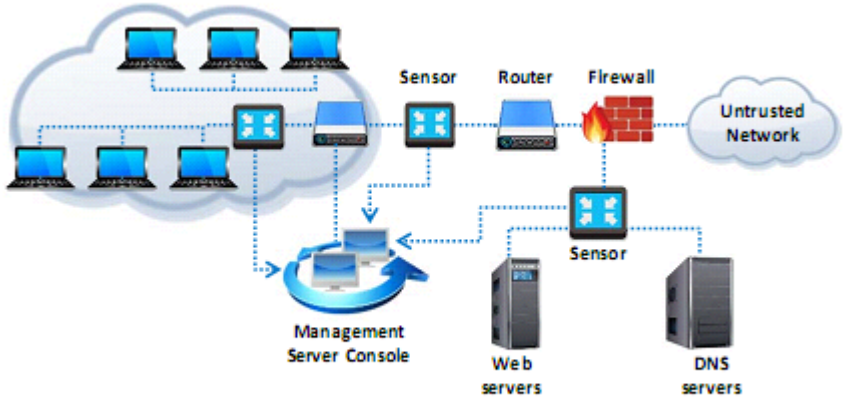
System Intrusions

- **Short** or incomplete logs
- Unusually **slow** system performance
- **Missing** logs or logs with incorrect permissions or ownership
- **Modifications** to system software and configuration files
- Unusual **graphic displays** or text messages
- **Gaps** in system accounting
- System crashes or **reboots**
- **Unfamiliar** processes

Types of Intrusion Detection Systems

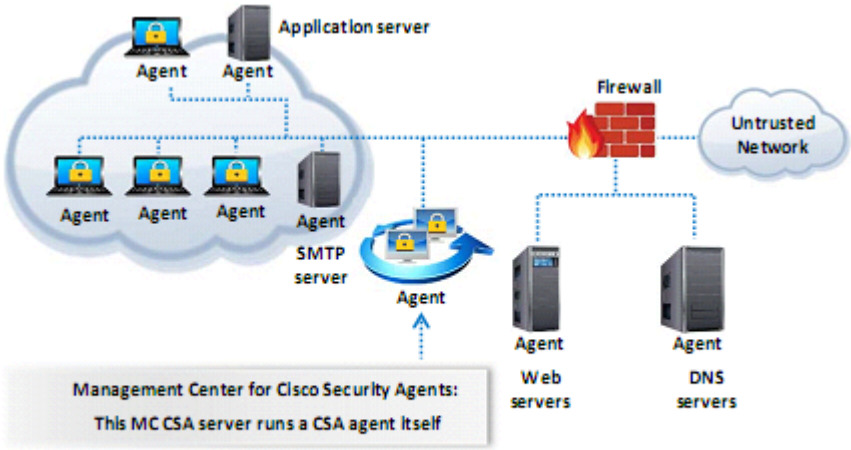
Network-Based Intrusion Detection Systems

- These mechanisms typically consist of a **black box** that is placed on the network in a promiscuous mode, listening for patterns indicative of an intrusion
- It detects malicious activity such as **Denial-of-Service attacks**, port scans, or even attempts to crack into computers by monitoring network traffic



Host-Based Intrusion Detection Systems

- These mechanisms usually include auditing for events that occur on a **specific host**
- These are not as common, due to the overhead they incur by having to **monitor each system event**

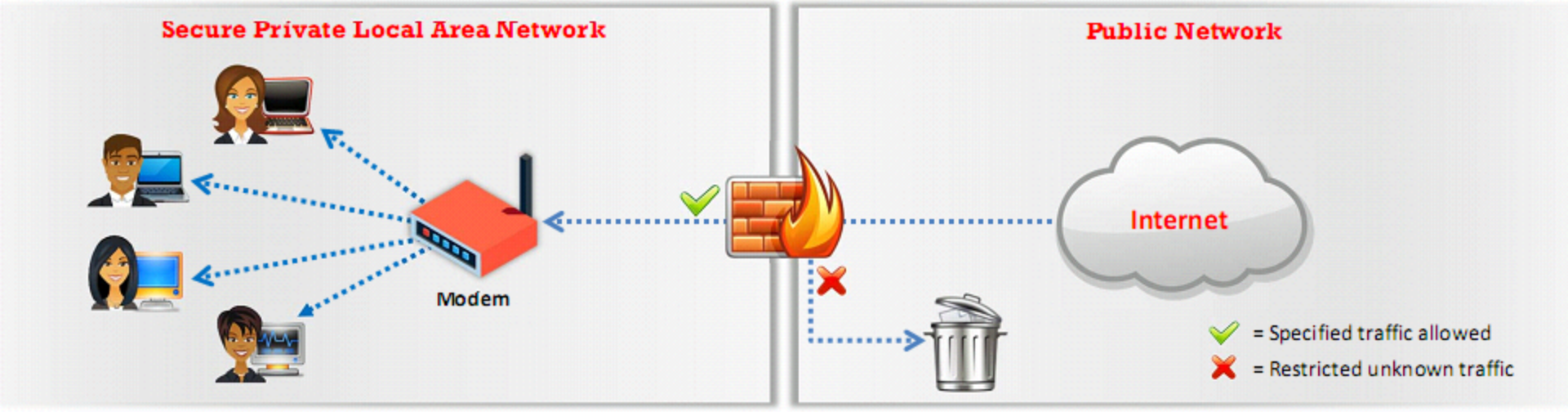


Types of IDS Alerts

True Positive (Attack - Alert)	An IDS raises an alarm when a **legitimate attack** occurs	✓ ✓
False Positive (No Attack - Alert)	An IDS raises an alarm when **no attack** has taken place	✗ ✓
False Negative (Attack - No Alert)	An IDS does not raise an alarm when a **legitimate attack** has taken place	✗ ✗
True Negative (No Attack - No Alert)	An IDS does not raise an alarm when an **attack** has not taken place	✓ ✗

Firewall

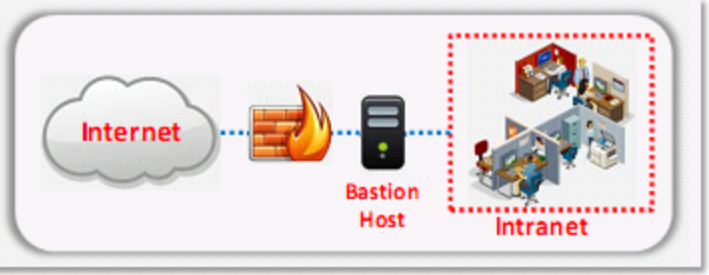
- Firewalls are hardware and/or software designed to prevent **unauthorized access** to or from a private network
- They are placed at the junction or **gateway** between the two networks, which is usually a private network and a public network such as the Internet
- Firewalls **examine all messages entering or leaving the Intranet** and block those that do not meet the specified security criteria



Firewall Architecture

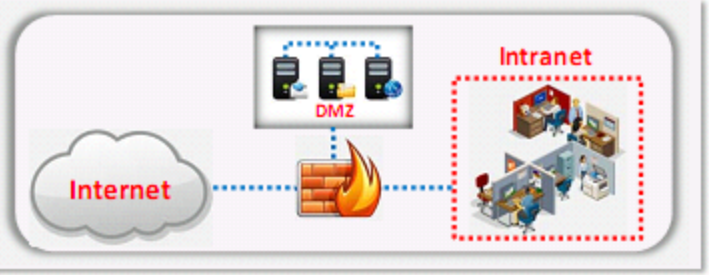
Bastion Host

- Bastion host is a computer system designed and configured to protect network resources from attack
- Traffic entering or leaving the network passes through the firewall. It has two interfaces:
  - public interface directly connected to the Internet
  - private interface connected to the Intranet



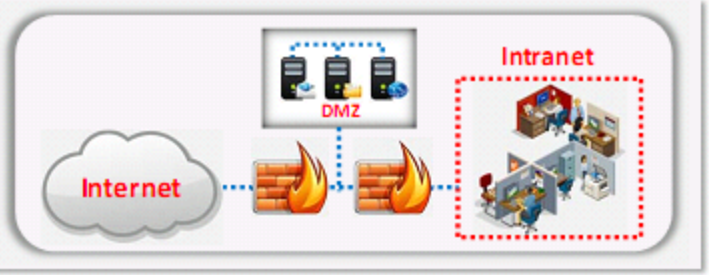
Screened Subnet

- The screened subnet or DMZ (additional zone) contains hosts that offer public services
- The DMZ zone responds to public requests, and has no hosts accessed by the private network
- Private zone can not be accessed by Internet users



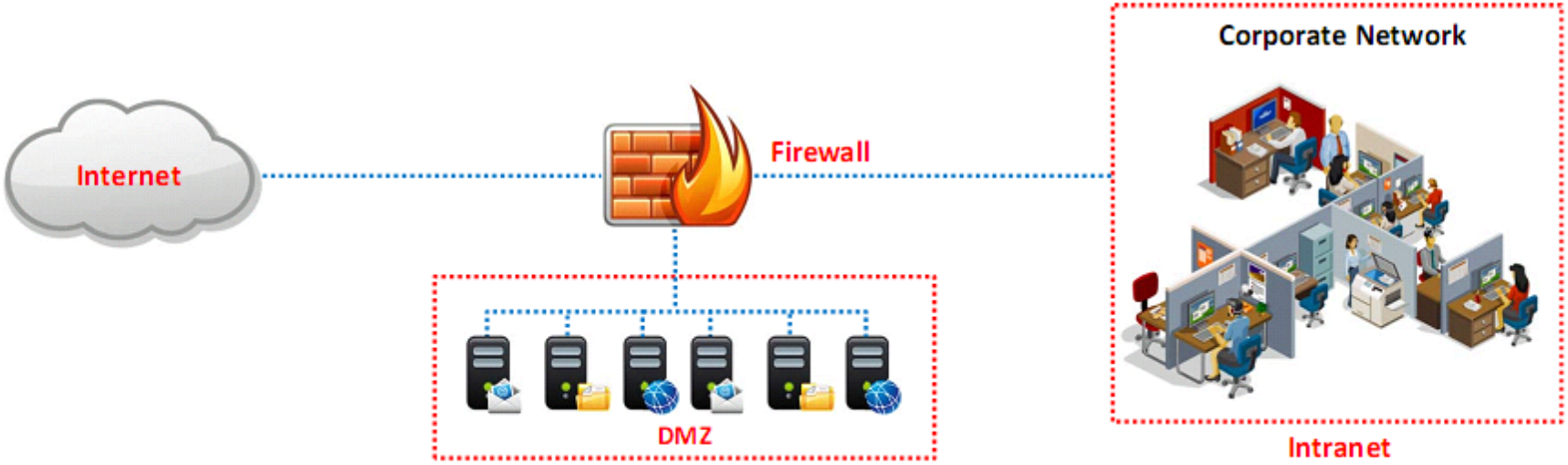
Multi-homed Firewall

- In this case, a firewall with two or more interfaces is present that allows further subdivision of the network based on the specific security objectives of the organization



DeMilitarized Zone (DMZ)

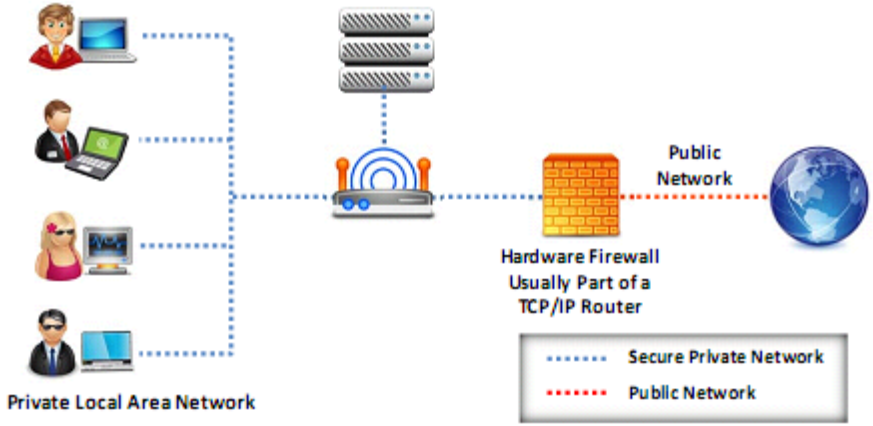
- DMZ is a network that **serves as a buffer** between the internal secure network and insecure Internet
- It can be created **using firewall with three or more network interfaces**, assigned with specific roles such as internal trusted network, DMZ network, and external un-trusted network



Types of Firewalls

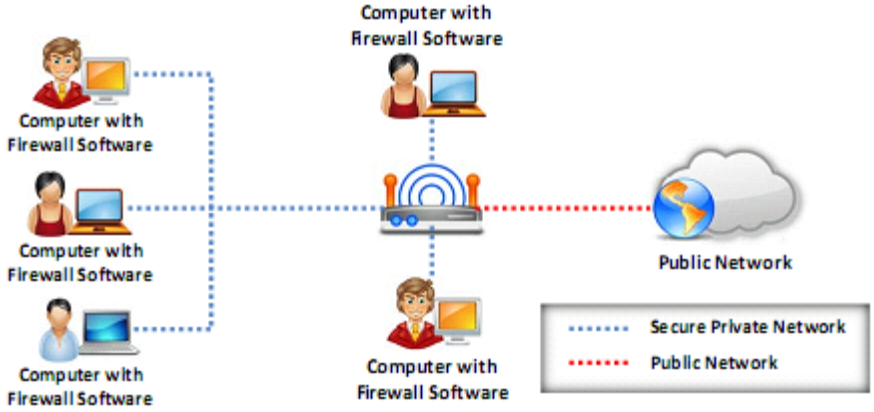
Hardware Firewall

- A hardware firewall is either a dedicated stand-alone hardware device or it comes as part of a router
- The network traffic is filtered using the packet filtering technique
- It is used to filter out the network traffic for large business networks



Software Firewall

- A software firewall is a software program installed on a computer, just like normal software
- It is generally used to filter traffic for individual home users
- It only filters traffic for the computer on which it is installed, not for the network



Note: It is recommended to configure both a software and a hardware firewall for best protection

Firewall Technologies

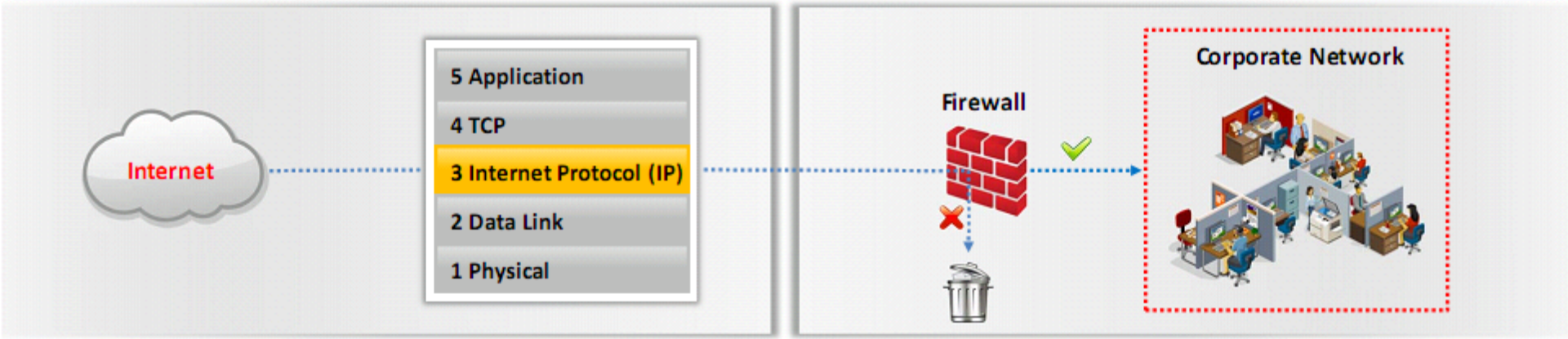
- Firewalls are designed and developed with the help of different **firewall services**
- Each firewall service provides security depending on its **efficiency** and **sophistication**

Technologies used for creating a firewall service

1. Packet Filtering
2. Circuit Level Gateways
3. Application Level Firewall
4. Stateful Multilayer Inspection
5. Application Proxies
6. Virtual Private Network
7. Network Address Translation

Packet Filtering Firewall

- Packet filtering firewalls work at the **network layer** of the OSI model (or the IP layer of TCP/IP). They are usually a part of a router
- In a packet filtering firewall, each packet is compared to a **set of criteria** before it is forwarded
- Depending on the **packet and the criteria**, the firewall can drop the packet or forward it, or send a message to the originator
- Rules can include the source and the destination **IP address**, the **source** and the **destination port number**, and the protocol used



= Traffic allowed based on source and destination IP address, packet type, and port number

= Disallowed Traffic

Circuit-Level Gateway Firewall

- Circuit-level gateways work at the **session layer** of the OSI model (or the TCP layer of TCP/IP)
- Information passed to a **remote computer** through a circuit-level gateway appears to have originated from the gateway
- They monitor **requests** to create sessions, and determine if those sessions will be allowed
- Circuit proxy firewalls **allow or prevent** data streams; they do not filter individual packets

= Traffic allowed based on specified applications (such as a browser) or a protocol, such as FTP, or combinations

= Disallowed Traffic

Application-Level Firewall

- Application-level gateways (proxies) can filter packets at the application layer of the OSI model (or the application layer of TCP/IP)
- Incoming and outgoing traffic is restricted to services supported by proxy; all other service requests are denied

- Application-level gateways configured as a web proxy prohibit FTP, gopher, telnet, or other traffic
- Application-level gateways examine traffic and filter on application-specific commands such as http:post and get

= Traffic allowed based on source and destination IP address, packet type, and port number
= Disallowed Traffic

Stateful Multilayer Inspection Firewall

- Stateful multilayer inspection firewalls **combine the aspects of the other three types** of firewalls (Packet Filtering, Circuit Level Gateways, and Application Level Firewall)
- They **filter packets** at the network layer of the OSI model (or the IP layer of TCP/IP), to determine whether session packets are legitimate, and they evaluate the contents of packets at the application layer

= Traffic is filtered at three layers based on a wide range of the specified application, session, and packet filtering rules

= Disallowed Traffic

Application Proxy

An application-level proxy works as a proxy server and **filters connections** for specific services

It filters connections based on the **services** and **protocols**, when acting as proxies

For example, A FTP proxy will only allow FTP traffic to pass through, while all other services and protocols will be blocked

Network Address Translation (NAT)

- Network address translation separates IP addresses into two sets and enables the LAN to use these addresses for **internal** and **external traffic** respectively

- It also works with a router, the same as packet filtering does. NAT will also **modify** the packets the router sends at the same time

- It has the ability to **change** the **address** of the packet and make it appear to have arrived from a valid address

- It limits the number of **public IP addresses** an organization can use

- It can act as a **firewall filtering technique** where it allows only those connections which originate on the inside network and will block the connections which originate on the outside network

Virtual Private Network

01

A VPN is a **private network** constructed using public networks, such as the Internet

It is used for the **secure transmission** of sensitive information over an untrusted network, using **encapsulation** and encryption

02

03

It establishes a virtual point-to-point connection through the use of **dedicated connections**

Only the **computing device** running the VPN software can access the VPN

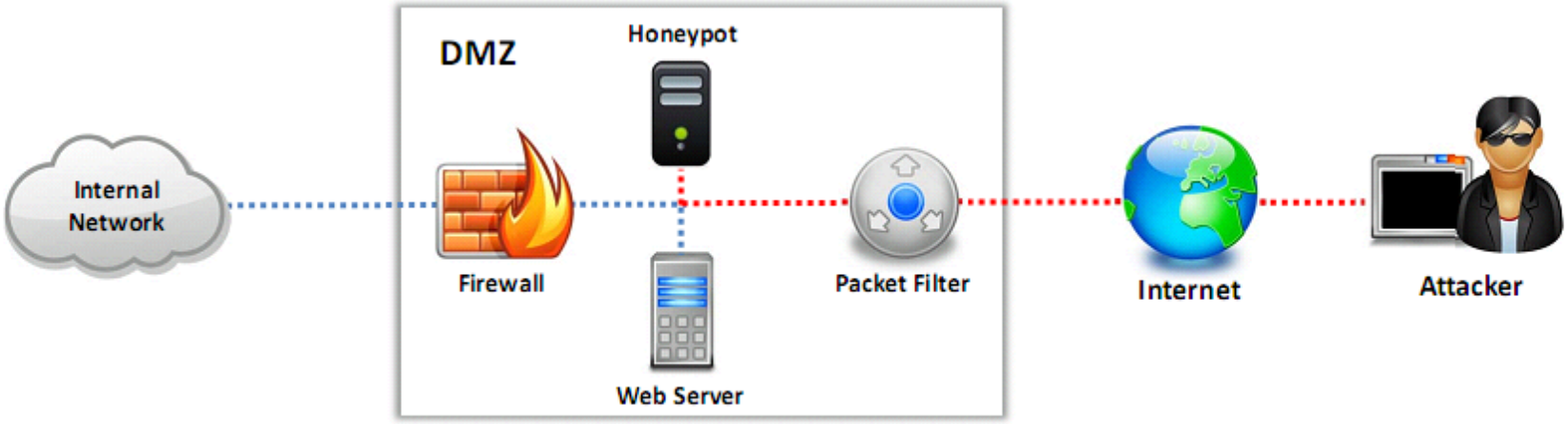
04

Firewall Limitations

- A firewall does not prevent the network from **new viruses**, **backdoor and insider attacks**
- A firewall cannot do anything if the network design and **configuration** is **faulty**
- A firewall is not an alternative to **antivirus** or **antimalware**
- A firewall cannot prevent **social engineering threats**
- A firewall does not prevent **passwords misuse**
- A firewall does not block attacks from a higher level of the **protocol stack**
- A firewall does not protect against attacks from **dial-in connections** and attacks originating from **common ports** and applications
- A firewall is unable to understand **tunneled traffic**

Honeypot

- A honeypot is an information system resource that is expressly **set up to attract and trap people** who attempt to penetrate an **organization's network**
- It has no authorized activity, does not have any **production value**, and any traffic to it is likely a **probe**, **attack**, or **compromise**
- A honeypot can **log port access** attempts, or monitor an **attacker's keystrokes**. These could be **early warnings** of a more concerted attack



Types of Honeypots

Low-interaction Honeypots	These honeypots simulate only a limited number of services and applications of a target system or network Generally, set to collect higher level information about attack vectors such as network probes and worm activities
Medium-interaction Honeypots	These honeypots simulate a real operating system, applications and its services This type of honeypots can only respond to preconfigured commands therefore the risk of intrusion increases
High-interaction Honeypots	These honeypots simulates all services and applications Capture complete information about an attack vector such as attack techniques, tools and intent of the attack
Production Honeypots	These honeypots emulate real production network of an organization Generally, set to collect internal flaws and attackers within an organization
Research Honeypots	These are high interaction honeypots primarily deployed in research institutes, government or military organizations Capture in-depth information about the way an attack is performed, vulnerabilities exploited and the attack techniques used by the attackers

Module Flow

1. IDS, Firewall and Honeypot Concepts
2. IDS, Firewall and Honeypot Solutions
3. Evading IDS
4. Evading Firewalls
5. IDS/Firewall Evading Tools
6. Detecting Honeypots
7. IDS/Firewall Evasion Countermeasures
8. Penetration Testing

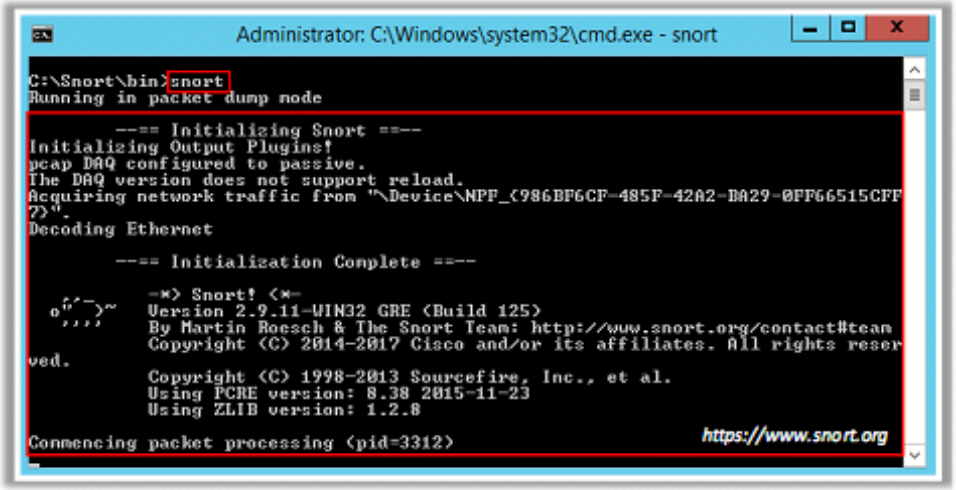
Intrusion Detection Tool: Snort

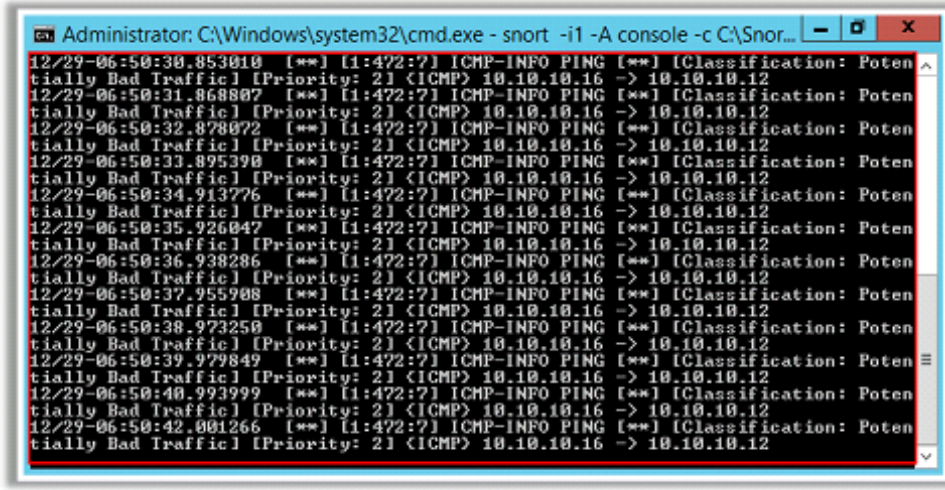
1. Snort is an open source network intrusion detection system, capable of performing real-time traffic analysis and packet logging on IP networks

2. It can perform protocol analysis and content searching/matching, and is used to detect a variety of attacks and probes, such as buffer overflows, stealth port scans, OS fingerprinting attempts, etc.

3. It uses a flexible rules language to describe traffic that it should collect or pass, as well as a detection engine that utilizes a modular plug-in architecture

4. **Uses of Snort:**
   - Straight packet sniffer like tcpdump
   - Packet logger (useful for network traffic debugging, etc.)
   - Network intrusion prevention system

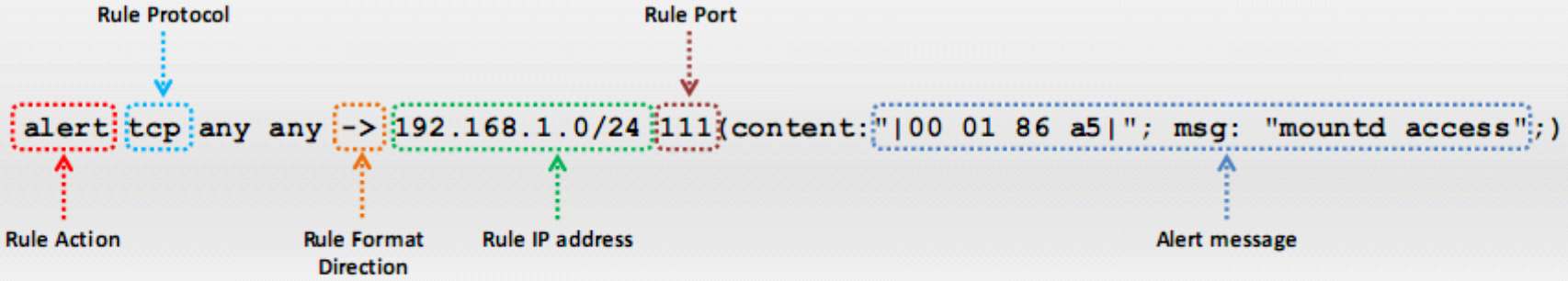




Snort Rules

- Snort's rule engine enables **custom rules** to meet the needs of the network
- Snort rules help in differentiating between **normal Internet activities** and **malicious activities**
- Snort rules must be contained on a **single line**, the Snort rule parser **does not handle rules on multiple lines**
- Snort rules come with two logical parts:
  - **Rule header**: Identifies **rule's actions** such as alerts, log, pass, activate, dynamic, etc.
  - **Rule options**: Identifies rule's **alert messages**

Example



Snort Rules: Rule Actions and IP Protocols

Rule Actions

- The rule header stores the complete **set of rules** to identify a packet, and determines the action to be performed or what rule to be applied
- The rule action **alerts Snort** when it finds a packet that matches the rule criteria
- Three available actions in Snort:
  - **Alert** - Generate an alert using the selected alert method, and then log the packet
  - **Log** - Log the packet
  - **Pass** - Drop (ignore) the packet

IP Protocols

Three available IP protocols that Snort supports for suspicious behavior:

I. TCP

II. UDP

III. ICMP

Snort Rules: The Direction Operator and IP Addresses

The Direction Operator

- This operator indicates the direction of interest for the traffic; traffic can flow in either a single direction or bi-directionally
- Example of a Snort rule using the **Bidirectional Operator**:

```
log !192.168.1.0/24 any <> 192.168.1.0/24 23
```

IP Addresses

- Identify IP address and the port that the rule applies to
- Use keyword "**any**" to define any IP address
- Use numeric IP addresses qualified with a CIDR netmask
- Example IP Address Negation Rule:

```
alert tcp !192.168.1.0/24 any -> 192.168.1.0/24 111 (content:
"|00 01 86 a5|"; msg: "external mountd access";)
```

Snort Rules: Port Numbers

- Port numbers can be listed in different ways, including "any" ports, static port definitions, port ranges, and by negation
- Port ranges are indicated with the range operator ":"
- Example of a Port Negation

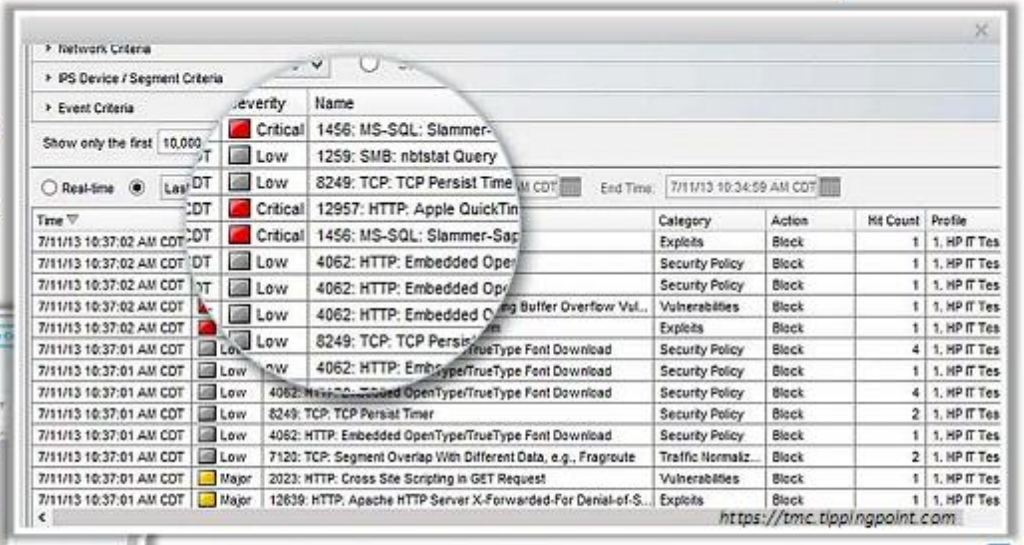
```
log tcp any any -> 192.168.1.0/24 !6000:6010
```

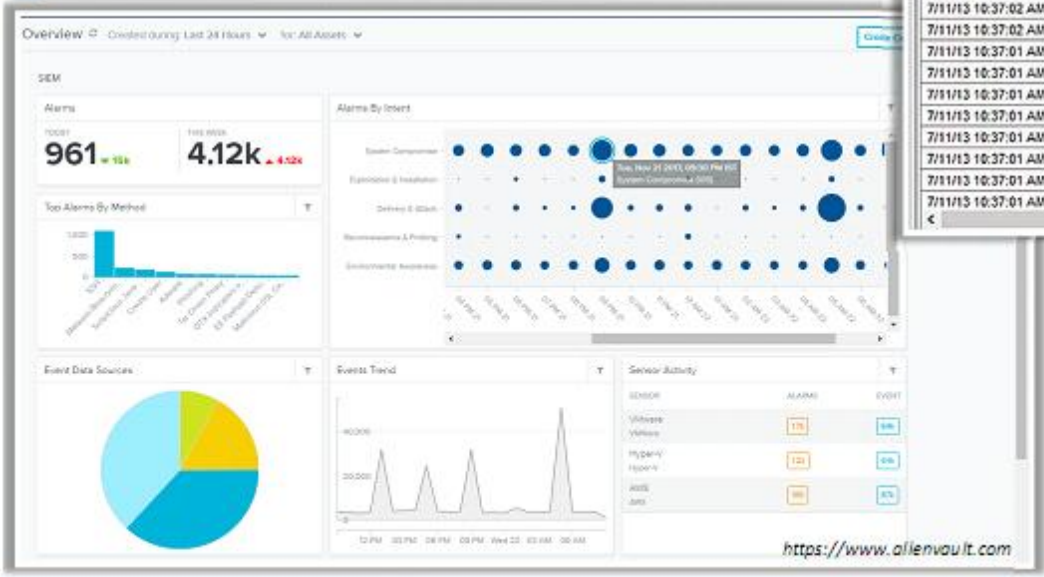
Protocols	IP address	Action
Log UDP any any ->	92.168.1.0/24 1:1024	Log UDP traffic coming from any port and destination ports ranging from 1 to 1024
Log TCP any any ->	192.168.1.0/24 :5000	Log TCP traffic from any port going to ports less than or equal to 5000
Log TCP any :1024 ->	192.168.1.0/24 400:	Log TCP traffic from the well known ports and going to ports greater than or equal to 400

Intrusion Detection Tools: TippingPoint and AlienVault® OSSIM™

TippingPoint

- TippingPoint IPS is in-line threat protection that defends critical data and applications without affecting performance and productivity
- It contains over 8,700 security filters written to address zero-day and known vulnerabilities





AlienVault® OSSIM™

- AlienVault® OSSIM™, Open Source SIEM, provides you with a feature-rich open source SIEM complete with event collection, normalization and correlation
- It provides advanced threat detection with real-time, prioritized alarms and minimal false positives

Intrusion Detection Tools

Check Point IPS Software Blade
https://www.checkpoint.com

Next-Generation Intrusion Prevention System (NGIPS)
https://www.cisco.com

OSSEC
https://ossec.github.io

IBM Security Network Intrusion Prevention System
https://www.ibm.com

FortiGate IPS
https://www.fortinet.com

Cisco Intrusion Prevention Systems
https://www.cisco.com

AlienVault Unified Security Management
https://www.alienvault.com

Next Generation Threat Prevention
https://www.checkpoint.com

AIDE (Advanced Intrusion Detection Environment)
http://aide.sourceforge.net

Cyberoam Intrusion Prevention System
https://www.cyberoam.com

Suricata
https://suricata-ids.org

Vangaurd Enforcer
https://www.go2vanguard.com

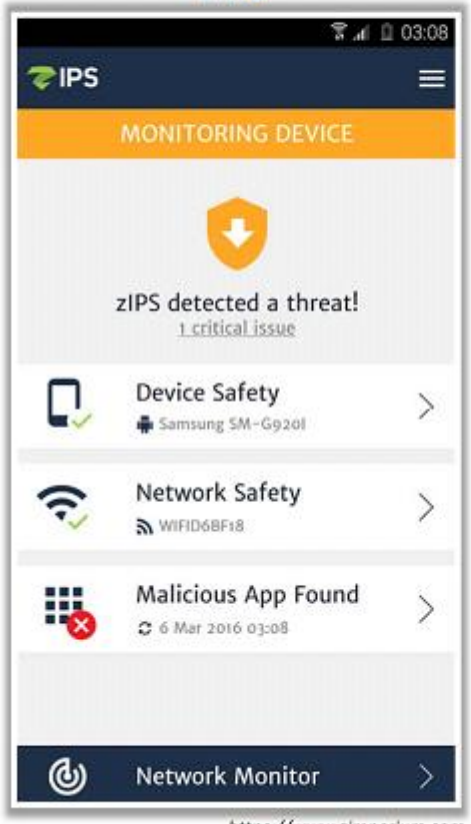
McAfee Host Intrusion Prevention for Desktops
https://www.mcafee.com

Snare
https://www.intersectalliance.com

INTOUCH INSA-Network Security Agent
http://www.ttinet.com

Intrusion Detection Tools for Mobile

zIPS



https://www.zimperium.com

Wifi Inspector

https://play.google.com

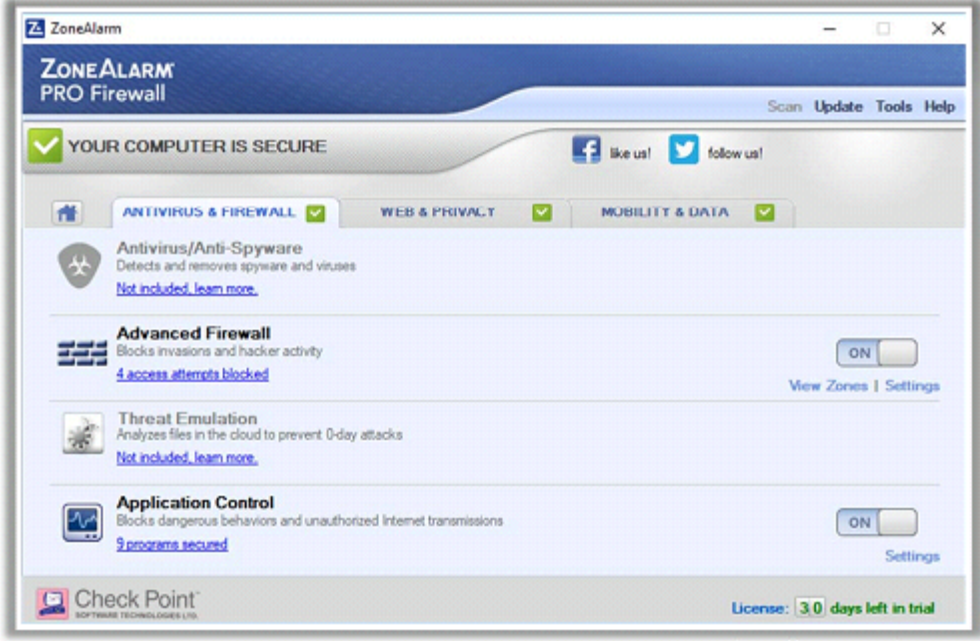
Wifi Intruder Detector pro

https://play.google.com

Firewalls: ZoneAlarm Free Firewall 2018 and Firewall Analyzer

ZoneAlarm PRO FIREWALL 2018

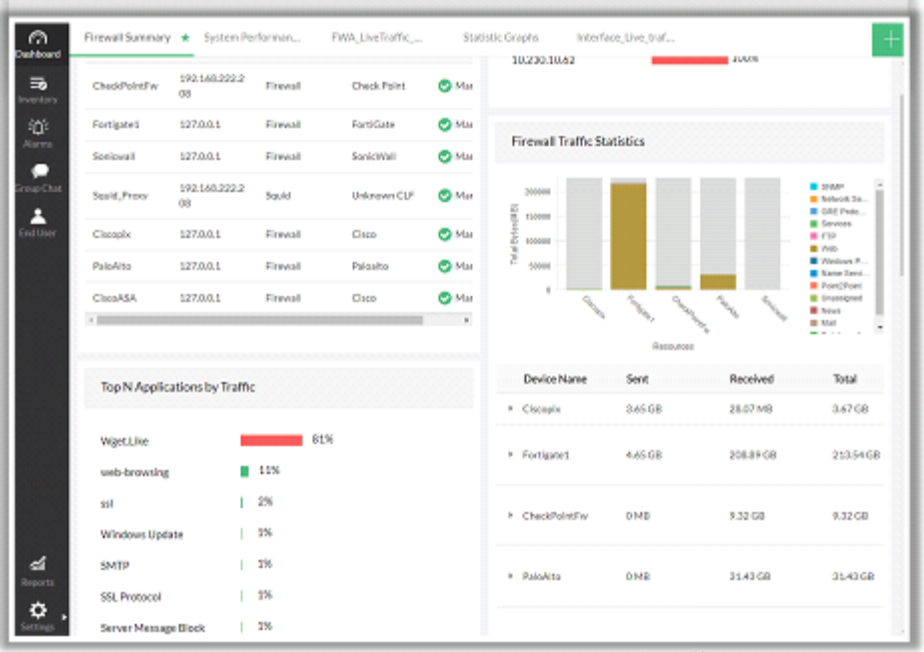
ZoneAlarm PRO FIREWALL 2018 monitors programs for suspicious behavior spotting and stopping new attacks that bypass traditional anti-virus protection



https://www.zonealarm.com

Firewall Analyzer

Firewall Analyzer offers a rich set of pre-defined reports that help in analyzing bandwidth usage and understanding network security



https://www.manageengine.com

Firewalls

Comodo Firewall
https://personalfirewall.comodo.com

Sophos XG Firewall
https://www.sophos.com

Check Point Firewall Software Blade
https://www.checkpoint.com

eScan Enterprise Edition
https://www.escanav.com

Untangle NG Firewall
https://www.untangle.com

Glasswire
https://www.glasswire.com

Zscaler Cloud Firewall
https://www.zscaler.com

TinyWall
https://tinywall.pados.hu

Cisco ASA
https://www.cisco.com

Meraki Cisco Firewall
https://meraki.cisco.com

Sonicwall NEXT GENERATION FIREWALLS
https://www.sonicwall.com

FortiGate Next-Generation Firewall
https://www.fortinet.com

Jetico Personal Firewall
http://www.jetico.com

Palo Alto Network Wildfire
https://www.paloaltonetworks.com

PeerBlock
http://forums.peerblock.com

Firewalls for Mobile

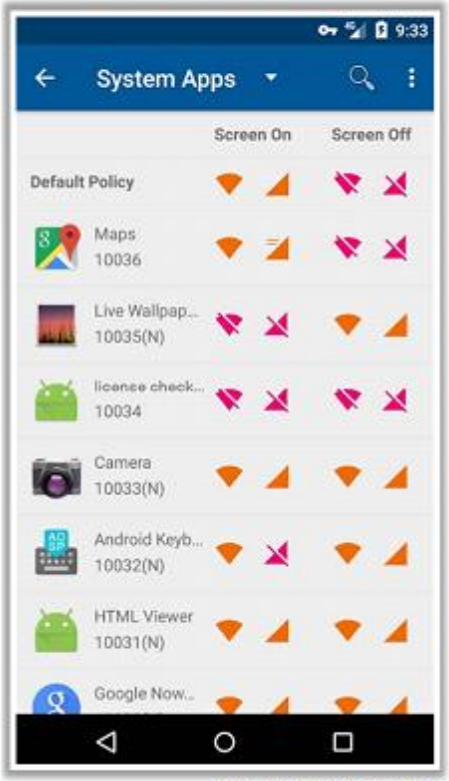
Mobiwol: NoRoot Firewall

http://www.mobiwol.com

Mobile Privacy Shield

https://shieldapps.com

NetPatch Firewall



https://firewall.netpatch.co

Firewall Gold
https://play.google.com

AFWall+
https://github.com

DroidWall - Android Firewall
https://play.google.com

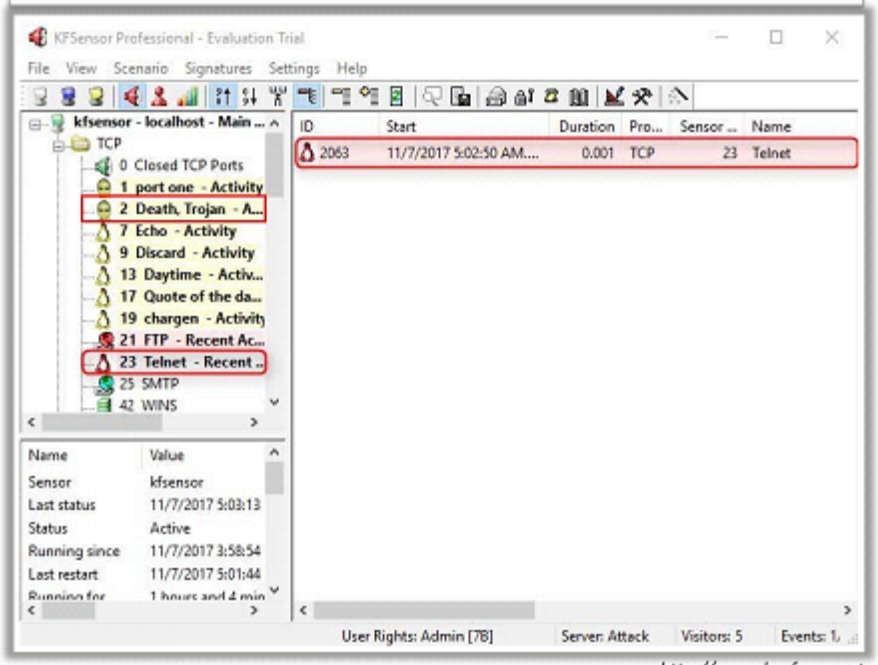
aFirewall
https://afirewall.wordpress.com

Root Firewall
http://www.rootuninstaller.com

Honeypot Tools: KFSensor and SPECTER

KFSensor

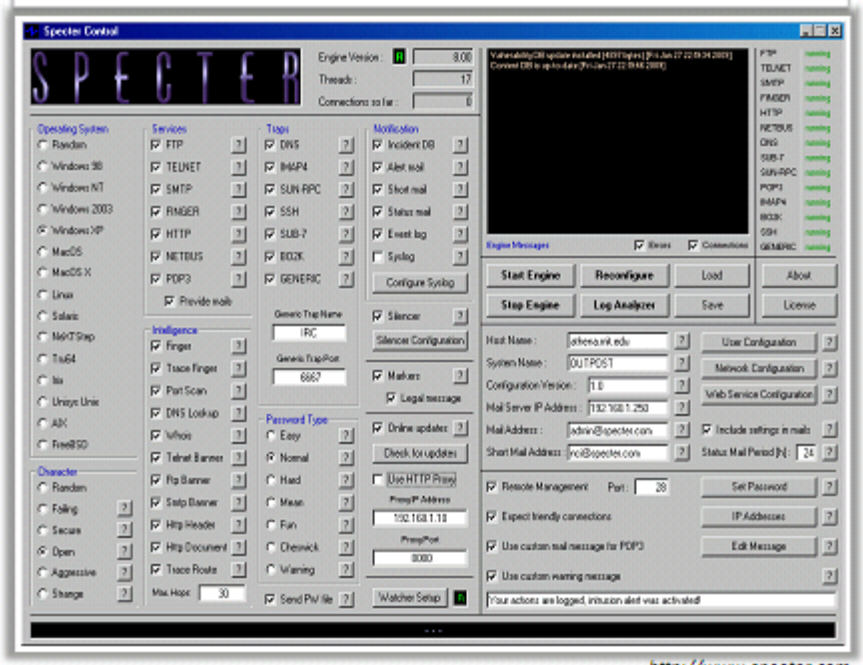
KFSensor is a host-based Intrusion Detection System (IDS) that acts as a honeypot to attract and detect hackers and worms by simulating vulnerable system services and Trojans



http://www.keyfocus.net

SPECTER

SPECTER is a smart honeypot-based intrusion detection system that offers common Internet services such as SMTP, FTP, POP3, HTTP, and TELNET which appear perfectly normal to the attackers but in fact are traps



http://www.specter.com

Honeypot Tools

HoneyBOT
https://www.atomicsoftwaresolutions.com

MongoDB-HoneyProxy
https://github.com

Honeyd
http://www.honeyd.org

Glastopf
https://github.com

Elasticsearch Honeypot
https://github.com

UML
http://user-mode-linux.sourceforge.net

Heralding
https://honeynet.org

mysql-honeypotd
https://github.com

Sebek
https://projects.honeynet.org

DCEPT
https://github.com

Super Next generation Advanced Reactive honEypot(Snare)
https://github.com

snort_inline
http://snort-inline.sourceforge.net

Modern Honey Network
https://github.com

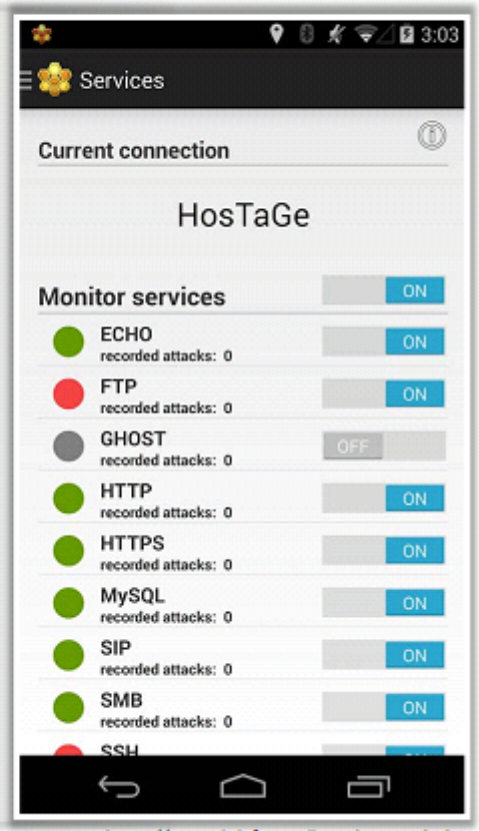
LaBrea Tarpit
http://labrea.sourceforge.net

Bait and Switch Honeypot
http://baitnswitch.sourceforge.net

Honeypot Tools for Mobile

HosTaGe

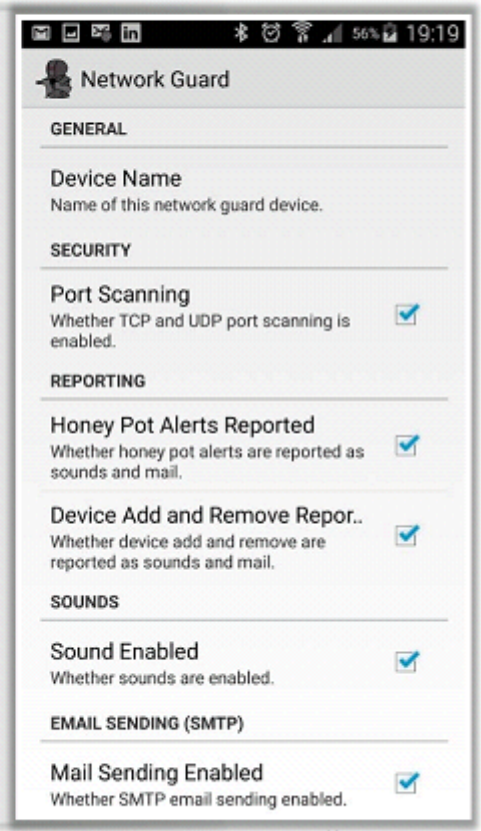
- HosTaGe is generic honeypot for mobile devices that aim on the detection of malicious, wireless network environments
- As most malware propagate over the network via specific protocols, a low-interaction honeypot located at a mobile device can check wireless networks for actively propagating malware



https://www.tk.informatik.tu-darmstadt.de

Network guard

- Network guard is a specialized App with automated network analysis and network honey pot for guarding your network
- All network assets are analyzed automatically and can be easily categorized and labelled
- It reports TCP connections to honey pot ports from remote hosts and marks these hosts automatically as suspects



https://play.google.com

Module Flow

1. IDS, Firewall and Honeypot Concepts
2. IDS, Firewall and Honeypot Solutions
3. Evading IDS
4. Evading Firewalls
5. IDS/Firewall Evading Tools
6. Detecting Honeypots
7. IDS/Firewall Evasion Countermeasures
8. Penetration Testing

IDS Evasion Techniques

1. Insertion Attack
2. Evasion
3. Denial-of-Service Attack
4. Obfuscating
5. False Positive Generation
6. Session Splicing
7. Unicode Evasion
8. Fragmentation Attack
9. Overlapping Fragments
10. Time-To-Live Attacks
11. Invalid RST Packets
12. Urgency Flag
13. Polymorphic Shellcode
14. ASCII Shellcode
15. Application-Layer Attacks
16. Desynchronization
17. Encryption
18. Flooding

Insertion Attack

1. Insertion is the process in which the **attacker confuses the IDS** by forcing it to read invalid packets
2. An IDS blindly believes and accepts a packet that an end system rejects and an attacker exploits this condition and **inserts data into the IDS**
3. This attack occurs when **NIDS is less strict** in processing packets than the internal network
4. The attacker obscures extra traffic and IDS concludes the traffic is harmless. Hence, the **IDS gets more packets** than the destination

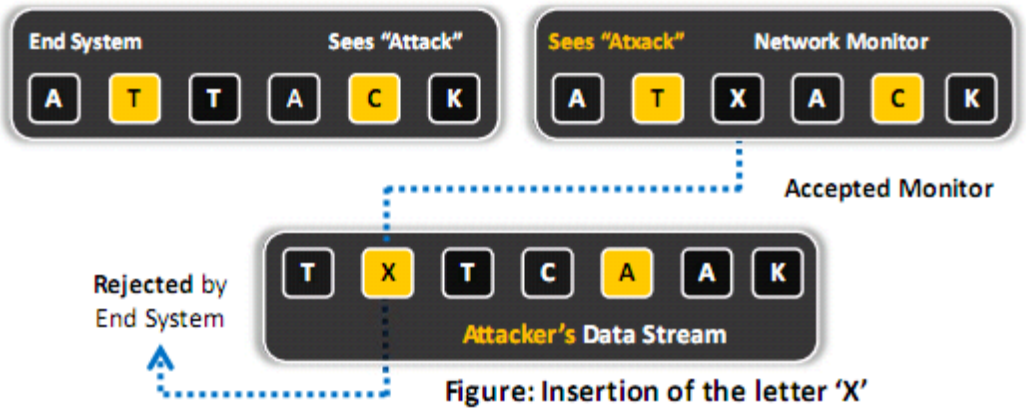


Figure: Insertion of the letter 'X'

- An attacker sends one-character packets to the target system via the IDS with **varying TTL** such that some packets reach the IDS but not the target system
- This will result in the IDS and the target system having **two different character strings**

Evasion

- In this evasion technique, an end system **accepts a packet** that an IDS rejects
- Using this technique, an attacker **exploits the host computer** without the IDS ever realizing it
- The attacker sends **portions of the request** in packets that the IDS mistakenly rejects, allowing the removal of parts of the stream from the IDS

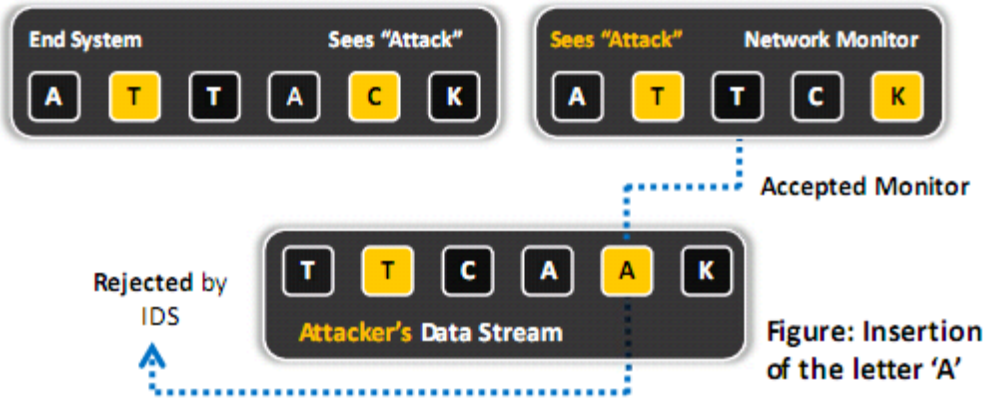


Figure: Insertion of the letter 'A'

- For example, if the malicious sequence is sent **byte-by-byte**, and one byte is rejected by the IDS, the IDS cannot detect the attack
- Here, the **IDS gets fewer packets** than the destination

Denial-of-Service Attack (DoS)

- Many IDSs use a **centralized server for logging** alerts
- If attackers know the **IP address of the centralized server** they can perform **DoS** or other hacks to slow down or crash the server
- As a result, attackers **intrusion attempts will not be logged**

Using this evasion technique, an attacker:

1. Causes the device to lock up
2. Causes personnel to be unable to investigate all the alarms
3. Causes more alarms than can be handled by management systems (such as databases, etc.)
4. Fills up disk space causing attacks to not be logged
5. Consumes the device's processing power and allows attacks to sneak by

Obfuscating

1. Obfuscating is an IDS evasion technique used by **attackers to encode the attack packet payload** in such a way that the destination host can only decode the packet but not the IDS
2. Attackers manipulate the **path referenced in the signature** to fool the HIDS
3. Attackers can **encode attack patterns in unicode** to bypass IDS filters, but be understood by an IIS web server
4. **Polymorphic code** is another means to circumvent **signature-based IDSs** by creating unique attack patterns, so that the attack does not have a single detectable signature
5. Attacks on **encrypted protocols** such as HTTPS are obfuscated if the attack is encrypted

False Positive Generation

1. Attackers with the knowledge of the target IDS, craft malicious packets just to generate alerts
2. These packets are sent to the IDS to generate a large number of false positive alerts
3. Attackers then use these false positive alerts to hide the real attack traffic
4. Attackers can bypass IDS unnoticed as it is difficult to differentiate the attack traffic from the large volume of false positives

Session Splicing

01 A technique used to bypass IDS where an attacker **splits the attack traffic** in to many packets such that no single packet triggers the IDS

02 It is effective against IDSs **that do not reconstruct** packets before checking them against intrusion signatures

03 If attackers are aware of **delay in packet reassembly** at the IDS, they can add delays between packet transmissions to bypass the reassembly

04 Many IDSs **stops reassembly** if they do not receive packets within a certain time

05 IDS will stop working if the target host keeps session active for a time longer than the **IDS reassembly time**

06 Any attack attempt after a successful splicing attack will **not be logged** by the IDS

Unicode Evasion

01 Unicode is a **character coding system** to support the worldwide interchange, processing, and display of the written texts

02 In the Unicode code space, all the code points are treated differently but it is possible that there could be **multiple representations of a single character**

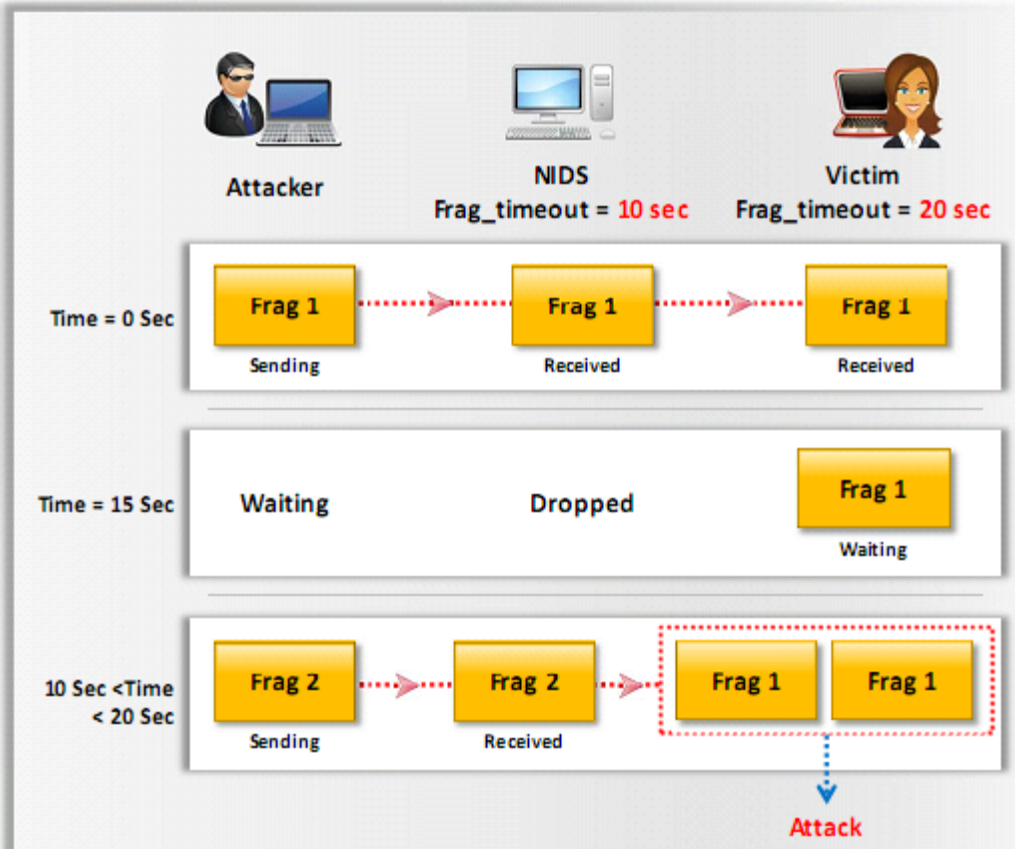
For example, / → %u2215, e → %u00e9 (UTF-16) and © → %c2%a9, ≠ → %e2%89%a0 (UTF-8)

03 Because of this complexity, some **IDS systems handle Unicode improperly** as Unicode allows multiple interpretations of the same characters

04 Taking this as an advantage, attackers can **convert attack strings to Unicode characters** to avoid pattern and signature matching at the IDS

Fragmentation Attack

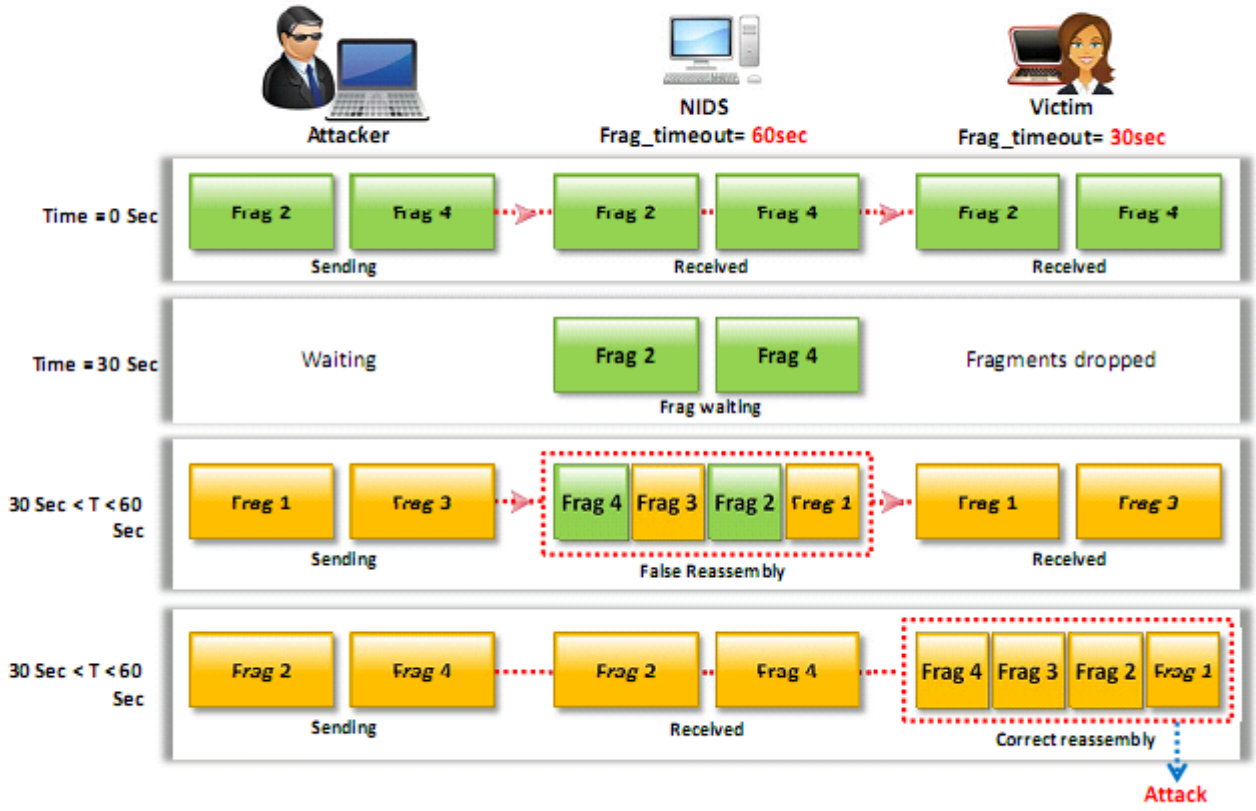
- Fragmentation can be used as an attack vector when **fragmentation timeouts** vary between IDS and host
- If fragment reassembly timeout is **10 seconds** at the IDS and **20 seconds** at the target system, attackers will send the second fragment after **15 seconds** of sending the first fragment
- In this scenario, the IDS will **drop the fragment** as the second fragment is received after its reassembly time but the target system will reassemble the fragments
- Attackers will keep sending the fragments with **15 second delays** until all the attack payload is reassembled at the target system



Fragmentation Attack (Cont'd)

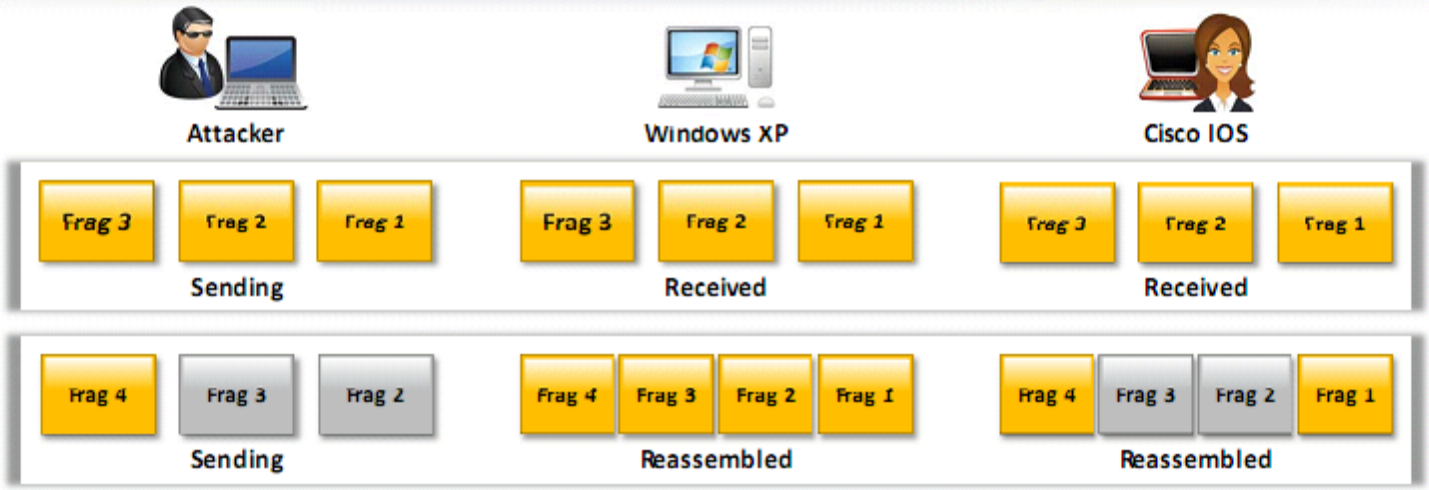
A similar fragmentation attack works when the IDS timeout exceeds the victim's

- Victim and IDS receive frag 2 and 4 out of 4 fragments, both carrying a false payload
- Victim drops these two fragments after 30 sec, and does not send ICMP since frag 1 never received
- Victim and IDS receive frag 1 and 3 out of 4 fragments
- IDS reassembles 4 received fragments, but computed net checksum is invalid, so packet is dropped
- Victim and IDS receive real frag 2 and 4 out of 4 fragments
- Victim reassembles 4 received fragments and is attacked; IDS times out frag 2 and 4 and drops



Overlapping Fragments

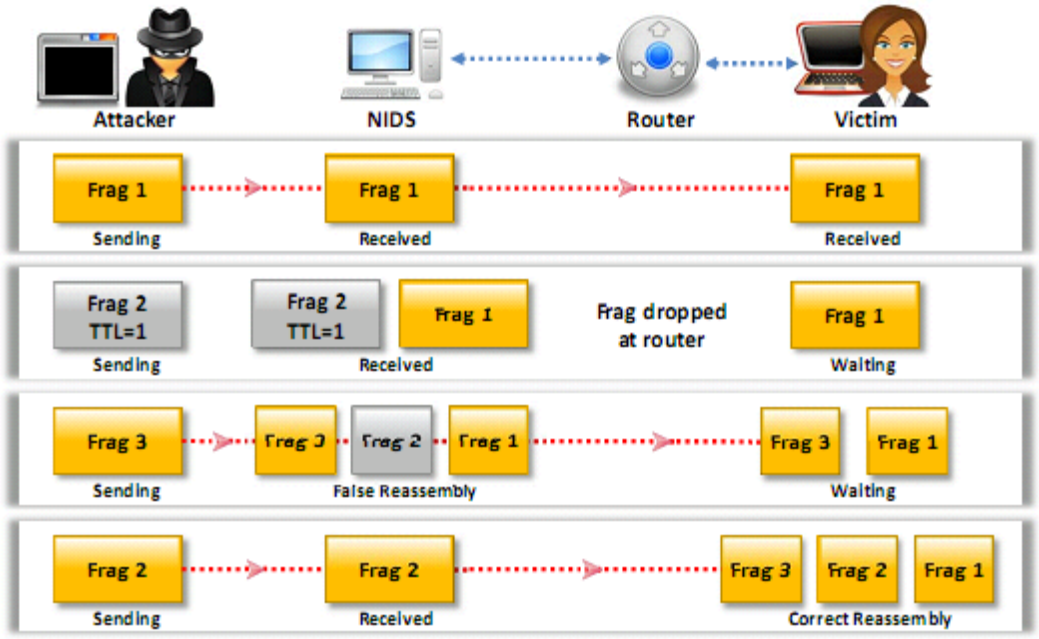
- An IDS evasion technique in which attackers **generate a series of tiny fragments** with overlapping TCP sequence numbers
- For example, the initial fragment consists of **100 bytes** of payload with a sequence number 1, the second fragment includes an overlapping sequence number 96 bytes, and so on
- At the time of **reassembling the packet** the destination host must know how to assemble the overlapping TCP fragments
- Some OSs will take the **original fragments with a given offset** (e.g., Windows W2K/XP/2003) and some operating systems will take the subsequent fragments with a given offset (e.g., Cisco IOS)



Time-To-Live Attacks

- These attacks require the attacker to have a **prior knowledge of the topology** of the victim's network
- This information can be obtained using tools such as **traceroute** which gives information on the **number of routers between the attacker and the victim**

1. Attacker breaks malicious traffic into **3 fragments**
2. Attacker sends **frag 1 with high TTL**, false frag 2 with low TTL
3. IDS receives both fragments, victim receives **first fragment only**
4. Attacker sends **frag 3 with high TTL**
5. IDS reassembles 3 fragments into meaningless packet and **drops**
6. Victim receives real frag 2, and **suffers attack**, while no log entry created



Invalid RST Packets

1. TCP uses 16-bit checksum field for **error-checking** of the header and data
2. **Reset (RST) flag** in a TCP header is used to close a TCP connection
3. In an invalid reset attack, attackers **send RST packet** to the IDS with an invalid checksum
4. IDS stops processing the packet thinking that the **TCP communication session** has ended but the target system will receive the packet
5. The target system **checks the RST packet's checksum** and drops it
6. The attack enables **attackers to communicate** with the target system while the IDS thinks that the communication has ended

Urgency Flag

1. Urgent (URG) flag in the TCP header is used to mark the data that require **urgent processing** at the receiving end
2. If the URG flag is set, the TCP protocol sets the Urgent Pointer field to a **16-bit offset value** that points to the last byte of urgent data in the segment
3. Many **IDSs do not consider the urgent pointer** and process all the packets in the traffic whereas the target system process only the urgent data
4. This results in the IDS and the target systems having **different sets of packets**, which can be exploited by attackers to pass the attack traffic

Urgency Flag Attack Example

```
"When a TCP packet contains both Urgent data and normal
data then 1-byte data after the urgent data is lost"
Packet 1: XYZ
Packet 2: LMN Urgency Pointer: 3
Packet 3: PQR
End result: XYZLMNQR
```

- The above example demonstrates the working of an urgency flag in a TCP packet
- According to the RFC 1122, when a TCP segment consists of an urgency pointer then one byte of data after the urgent data will be lost.

Polymorphic Shellcode

1. A signature-based network intrusion detection system (NIDS) identifies an attack by **matching attack signatures** with incoming and outgoing data packets
2. Many IDSs identify signatures for the **commonly used strings** embedded in the shellcode
3. Polymorphic shellcode attacks include **multiple signatures** making it difficult to detect the signature
4. Attackers **encode the payload** using some technique and then place a decoder before the payload
5. As a result of this the **shellcode is completely rewritten** each time it is sent evading detection
6. This technique also **evades the commonly used shellcode strings**, thus making shellcode signatures unusable

ASCII Shellcode

ASCII shellcode includes characters which are present only in **ASCII standard**

Attackers can use ASCII shellcode to bypass the IDS signature as the **pattern matching** does not work effectively with the ASCII values

Scope of ASCII shellcode is **limited** as all assembly instructions cannot be converted to ASCII values directly

This limitation can be overcome by using other **sets of instructions** for converting to ASCII values properly

The following is an ASCII shellcode example:

```
char shellcode[] =
"LLLLYhb0pLX5b0pLHSSPPWQPPaPWSUTBRDJfh5t
DS"
"RajYX0Dka0TkafhN9fYf1Lkb0TkdjfY0Lkf0Tkg
fh"
"6rfYf1Lki0tkkh95h8Y1LkmjpY0Lkq0tkrh2wnu
X1"
"Dks0tkwjfX0Dkx0tkx0tkyCjnY0LkzC0TkzCCjt
X0"
"DkzC0tkzCj3X0Dkz0TkzC0tkzChjG3IY1LkzCCC
C0"
"tkzChpfcMX1DkzCCCC0tkzCh4pCnY1Lkz1TkzCC
CC"
"fhJGfXf1Dkzf1tkzCCjHX0DkzCCCCjvY0LkzCCC
jd"
"X0DkzC0TkzCjWX0Dkz0TkzCjdX0DkzCjXY0Lkz0
tk"
"zMdgvvn9F1r8F55h8pG9wnuvjrNfrVx2LGkG3ID
pf"
"cM2KgmnJGgbinYshdvD9d";
```

When executed, the shellcode above executes a `"/bin/sh"` shell. `'bin'` and `'sh'` are contained in the last few bytes of the shellcode

Application-Layer Attacks

- Applications accessing media files (audio, video and images) **compress** them to smaller size for maximizing data transfer rate

- IDS cannot verify the **signature of compressed file** format

- This enables an attacker to **exploit the vulnerabilities** in compressed data

- IDS can recognize particular conditions favorable for attack but other alternative forms of attack are also possible, for example, various integer values can be used to **exploit integer overflow vulnerabilities**

- This makes the detection of attack traffic **extremely difficult** at the IDS

Desynchronization

Pre-Connection SYN

- This attack is performed by sending an **initial SYN before the real connection** is established, but with an invalid TCP checksum
- If a SYN packet is received **after the TCP control block is opened**, the IDS resets the appropriate sequence number to match that of the newly received SYN packet
- Attackers send **fake SYN packets** with a completely invalid sequence number to desynchronize the IDS
- This **stops IDS** from monitoring all, legitimate and attack, traffic

Post-Connection SYN

- For this technique, attempt to **desynchronize the IDS** from the actual sequence numbers that the kernel is honoring
- Send a **post connection SYN packet** in the data stream, which will have **divergent sequence** numbers
- However, the target host will ignore this **SYN packet**, as it references an already established connection
- The intent of this attack is to get the IDS to **resynchronize** its notion of the sequence numbers to the new SYN packet
- It will then ignore any data that is a **legitimate part of the original stream**, because it will be awaiting a different sequence number
- Once successful in resynchronizing the IDS with a SYN packet, send an **RST packet with the new sequence number** and close down its notion of the connection

Other Types of Evasion

Encryption

When the attacker has already established an **encrypted session with the victim**, it results in the most effective evasion attack

Flooding

The attacker sends loads of **unnecessary traffic to produce noise**, and if IDS does not analyze the noise traffic well, then the true attack traffic may go undetected

Module Flow

Firewall Evasion Techniques

1. Firewalking
2. Banner Grabbing
3. IP Address Spoofing
4. Source Routing
5. Tiny Fragments
6. Using IP Address in Place of URL
7. Using Proxy Server
8. ICMP Tunneling
9. ACK Tunneling
10. HTTP Tunneling
11. SSH Tunneling
12. Through External Systems
13. Through MITM Attack
14. Through Content
15. Through XSS Attack

Firewall Identification

Port Scanning

- Port scanning is used to **identify open ports** and services running on these ports
- Open ports can be further probed to identify the **version of services**, which helps in finding vulnerabilities in these services
- Some firewalls **will uniquely identify themselves** in response to simple port scans
- For example: **Check Point's FireWall-1** listens on TCP ports 256, 257, 258, and 259. Microsoft's Proxy Server listens on TCP ports 1080 and 1745

Firewalking

- A technique that uses TTL values to determine gateway **ACL filters** and map networks by analyzing IP packet responses
- Attackers send a TCP or UDP packet to the targeted firewall with a **TTL set to one hop greater** than that of the firewall
- If the packet makes it through the gateway, it is forwarded to the next hop where the TTL equals one and elicits an ICMP "**TTL exceeded in transit**" to be returned, as the original packet is discarded
- This method helps locate a firewall. Additional probing permits **fingerprinting** and **identification of vulnerabilities**

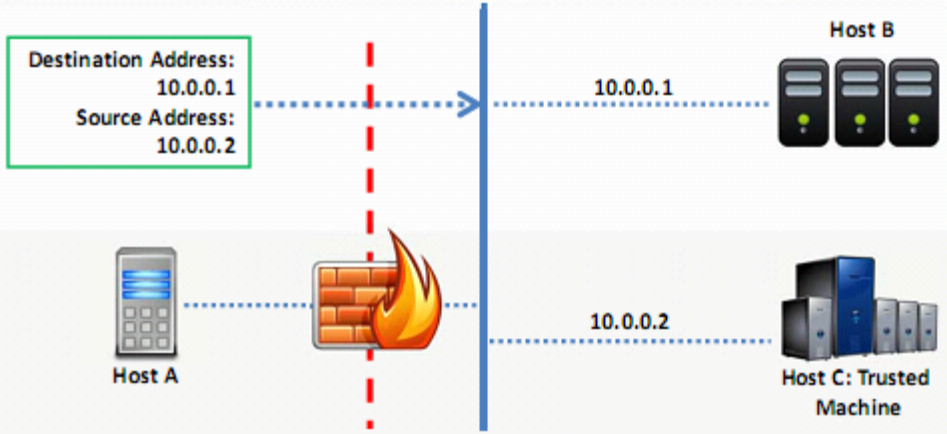
Banner Grabbing

- Banners are **service announcements** provided by services in response to connection requests, and often carry vendor version information
- Banner grabbing is a simple method of **fingerprinting** that helps in detecting the vendor of a firewall, and the firmware's version
- The three main services which send out banners are **FTP**, **telnet**, and **web servers**
- An example of SMTP banner grabbing is: telnet mail. targetcompany.org 25

IP Address Spoofing

- IP address spoofing is a hijacking technique in which an attacker **masquerades as a trusted host** to conceal his identity, spoof a Web site, hijack browsers, or gain unauthorized access to a network
- Attackers modify the **addressing information** in the IP packet header and the source address bits field in order to bypass the firewall

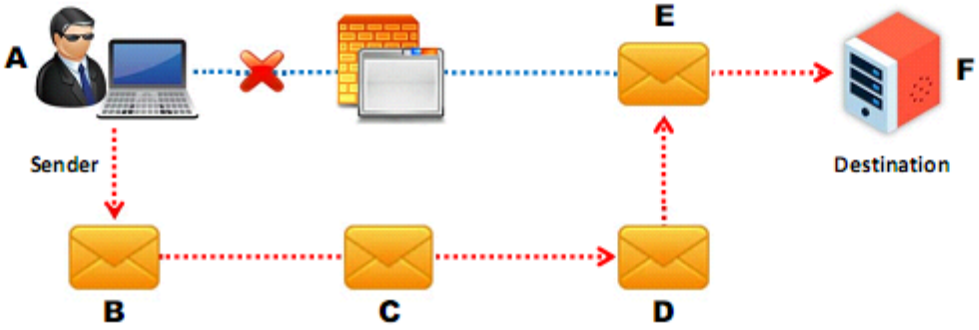
- For example, let's consider **three hosts**: A, B and C
- Host **C is a trusted machine** of host B
- Host A masquerades as host C by **modifying the IP address** of the malicious packets that he intends to send to the host B
- When the **packets are received**, host B thinks that they are from host C, but are actually from host A



Source Routing

- Source routing allows the sender of a packet to partially or completely **specify the route**, the packet takes through the network
- As the packet travels through the nodes in the network, each **router examines** the destination IP address and **chooses the next hop** to direct the packet to the destination
- In source routing, the **sender** makes some or all of these decisions on the router

The figure shows source routing, where the originator dictates eventual route of traffic



Tiny Fragments

- Attackers create **tiny fragments** of outgoing packets forcing some of the TCP packet's header information into the next fragment
- The IDS filter rules that specify **patterns will not match** with the fragmented packets due to broken header information
- The attack will succeed if the **filtering router examines only the first fragment** and allows all the other fragments to pass through
- This attack is used to **avoid user defined filtering rules** and works when the **firewall checks only for the TCP header information**

IP-3ar0JI0B0K			MK=1, Fragment Offset=0					
Source Port			Destination Port					
Sequence Number								
Acknowledgement Sequence Number								
Data Offset	Reserved	-	ACK	-	-	-	-	Window
Checksum								Urgent Pointer=0
0								

Bypass Blocked Sites Using IP Address in Place of URL

1. This method involves typing the **IP address** directly in browser's address bar in place of typing the **blocked website's domain name**
2. For example, to access Facebook, type its **IP address** instead of typing its domain name
3. Use services such as **Host2ip** to find the IP address of the blocked website
4. This method fails if the blocking software **tracks the IP address** sent to the web server

Bypass Blocked Sites Using Anonymous Website Surfing Sites

- There are many online anonymizer services that enable anonymous surfing on the Internet
- Some websites provide options to encrypt the URL's of the websites
- These services hide the actual IP address of the surfer and enable bypassing the IP-based firewall filter rules

Anonymizers

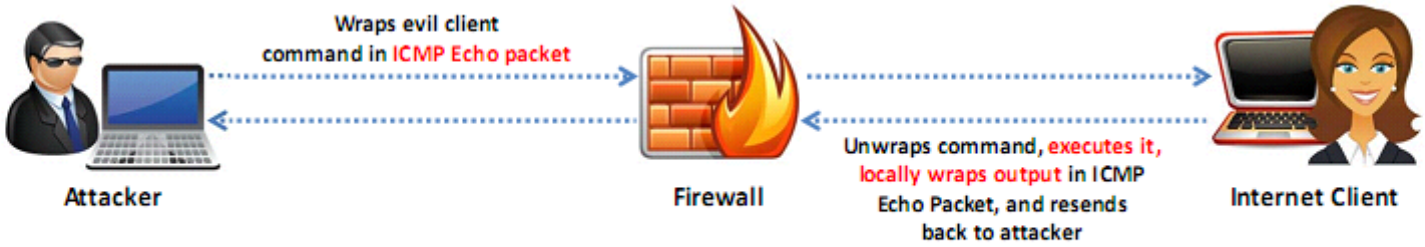
1	https://www.anonymizer.com	6	http://www.guardster.com
2	http://www.webproxyserver.net	7	http://anonymouse.org
3	https://anonymous-proxy-servers.net	8	http://www.boomproxy.com
4	https://zendproxy.com	9	http://anype.com
5	https://proxify.com	10	http://www.spysurfing.com

Bypass a Firewall Using Proxy Server

1. Find an appropriate **proxy server**
2. On the Tools menu of any Internet browser, go to "**Proxy Settings**" and in the **Internet Properties** dialog box under **Connections** tab, click "**LAN settings**"
3. Under LAN Settings, click on a "**Use a proxy server for your LAN**" check box
4. In the **Address** box, type the **IP address** of the proxy server
5. In the **Port** box, type the **port number** that is used by the proxy server for client connections (by default, 8080)
6. Click to select "**Bypass proxy server for local addresses**" check box if you do not want the proxy server computer to be used when connected to a computer on the local network
7. Click **OK** to close the **LAN Settings** dialog box
8. Click **OK** again to close the **Internet Properties** dialog box

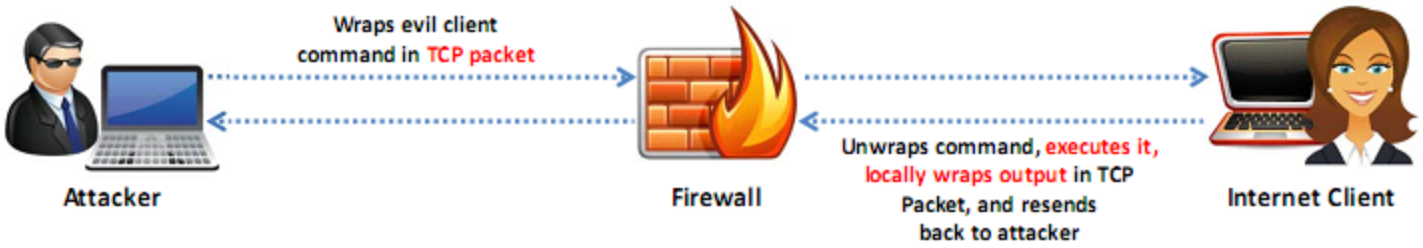
Bypassing Firewall through ICMP Tunneling Method

- It allows tunneling a **backdoor shell** in the data portion of ICMP Echo packets
- RFC 792, which delineates **ICMP operation**, does not define what should go in the data portion
- The **payload portion** is arbitrary and is not examined by most of the firewalls, thus any data can be inserted in the payload portion of the ICMP packet, including a **backdoor application**
- Some administrators keep **ICMP open** on their firewall because it is useful for tools like **ping** and **traceroute**
- Assuming that ICMP is allowed through a firewall, use **Loki ICMP tunneling** (*https://tools.cisco.com*) to execute commands of choice by tunneling them inside the payload of **ICMP echo packets**



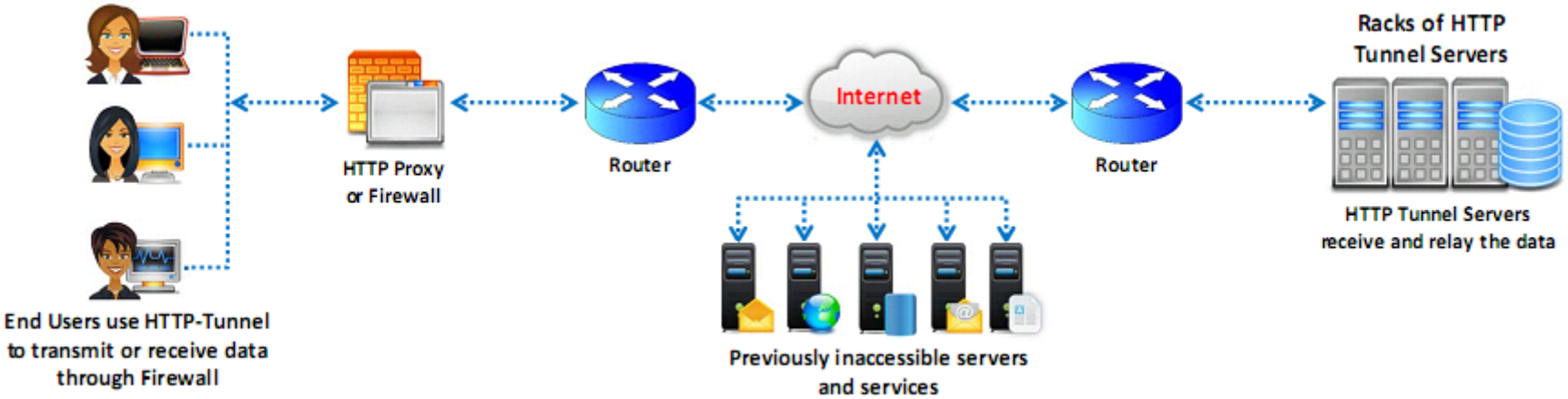
Bypassing Firewall through ACK Tunneling Method

- It allows tunneling a backdoor application with **TCP packets with the ACK bit set**
- ACK bit is used to **acknowledge receipt of a packet**
- Some firewalls **do not check packets with ACK bit set** because ACK bits are supposed to be used in response to legitimate traffic
- Tools such as **AckCmd** (*http://ntsecurity.nu*) can be used to implement ACK tunneling



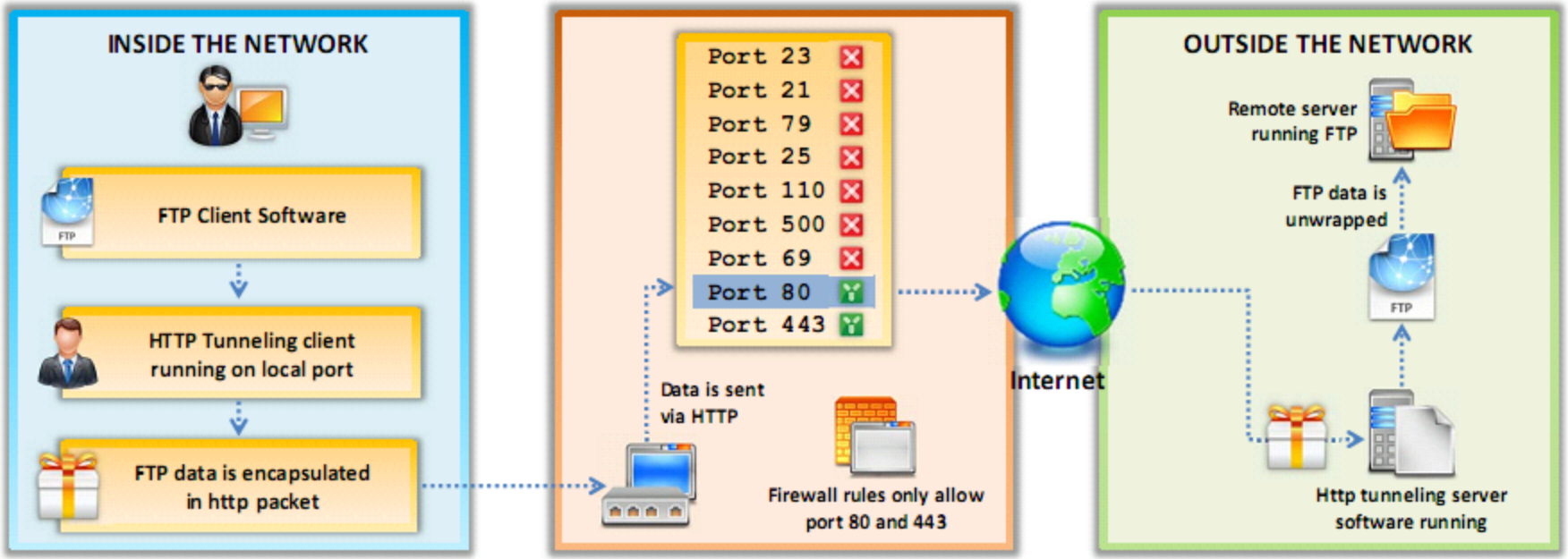
Bypassing Firewall through HTTP Tunneling Method

1. HTTP Tunneling technology allows attackers to **perform various Internet tasks** despite the restrictions imposed by firewalls
2. This method can be implemented if the target company has a **public web server with port 80** used for HTTP traffic, that is unfiltered on its firewall
3. Encapsulates data inside **HTTP traffic** (port 80)



Why do I Need HTTP Tunneling

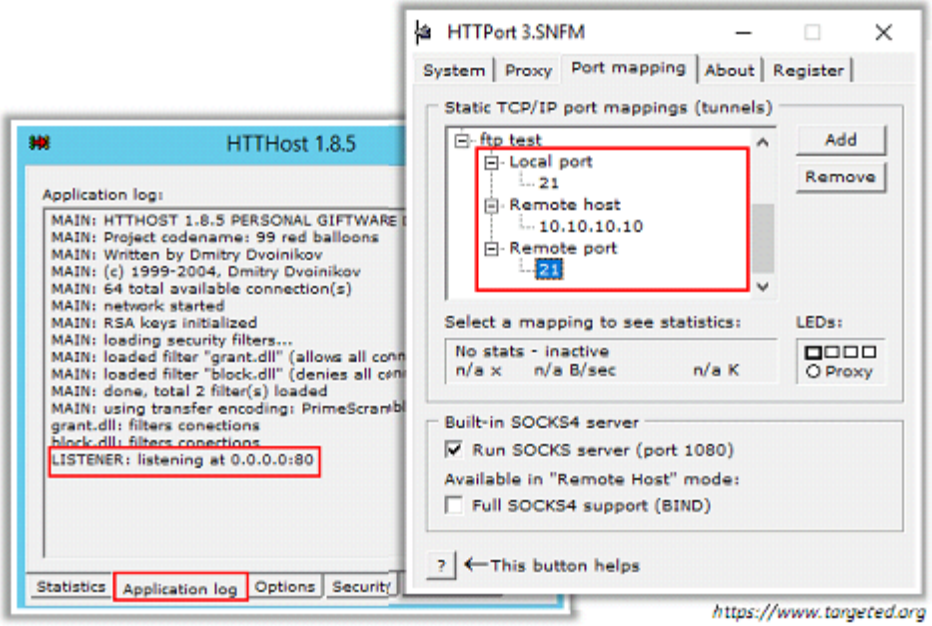
- For instance, consider that an organization's firewalls restrict users to access all ports except **80** and **443**, and a user may want to use FTP
- HTTP tunneling will enable use of **FTP via HTTP protocol**



HTTP Tunneling Tools

HTTPort and HTTHost

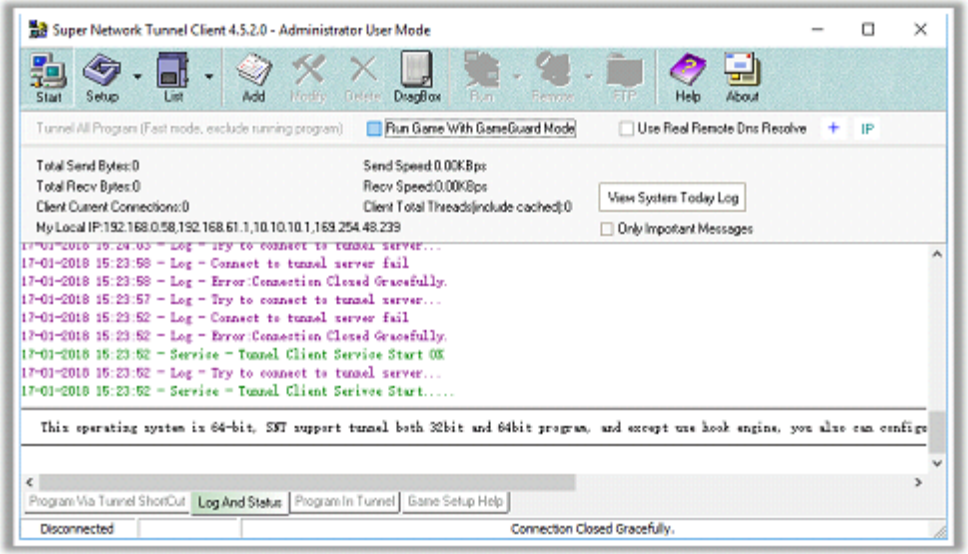
- HTTPort allows you to bypass your HTTP proxy, which is blocking you from the Internet
- It allows you to use various Internet software from behind the proxy, ex. e-mail, instant messengers, P2P file sharing, ICQ, News, FTP, IRC, etc.



https://www.targeted.org

Super Network Tunnel

- A two-way http tunnel software connecting two computers
- Works like VPN tunneling but uses HTTP protocol to establish a connection



http://www.networktunnel.net

Bypassing Firewall through SSH Tunneling Method

OpenSSH

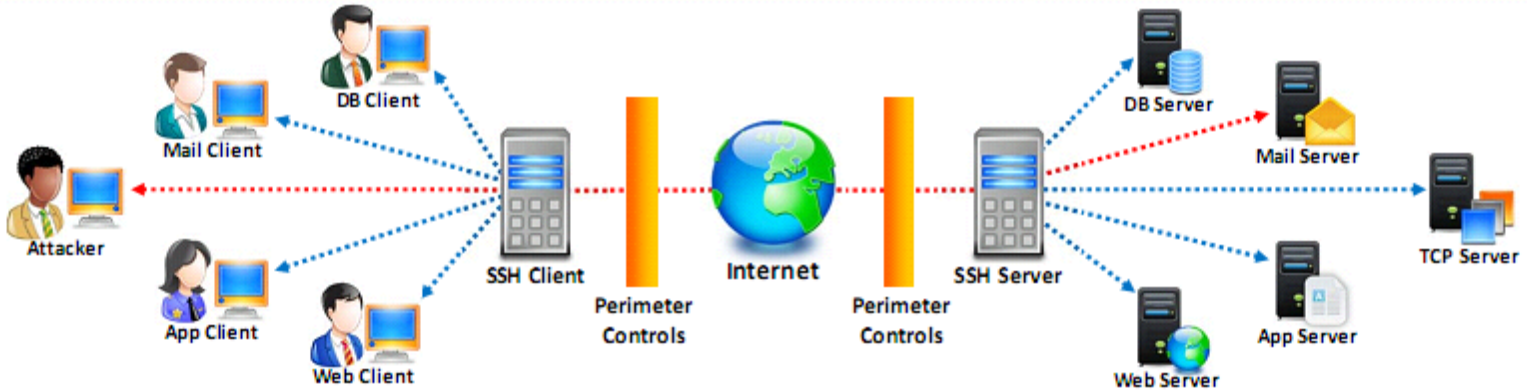
- Attackers use OpenSSH to **encrypt and tunnel all the traffic** from a local machine to a remote machine to avoid detection by perimeter security controls

Example

```
ssh -f user@certifiedhacker.com -L 5000:certifiedhacker.com:25 -N
```

`-f` => background mode, `user@certifiedhacker.com` => user name and server you are logging into, `-L 5000:certifiedhacker.com:25` => local-port:host:remote-port, and `-N` => Do not execute the command on the remote system

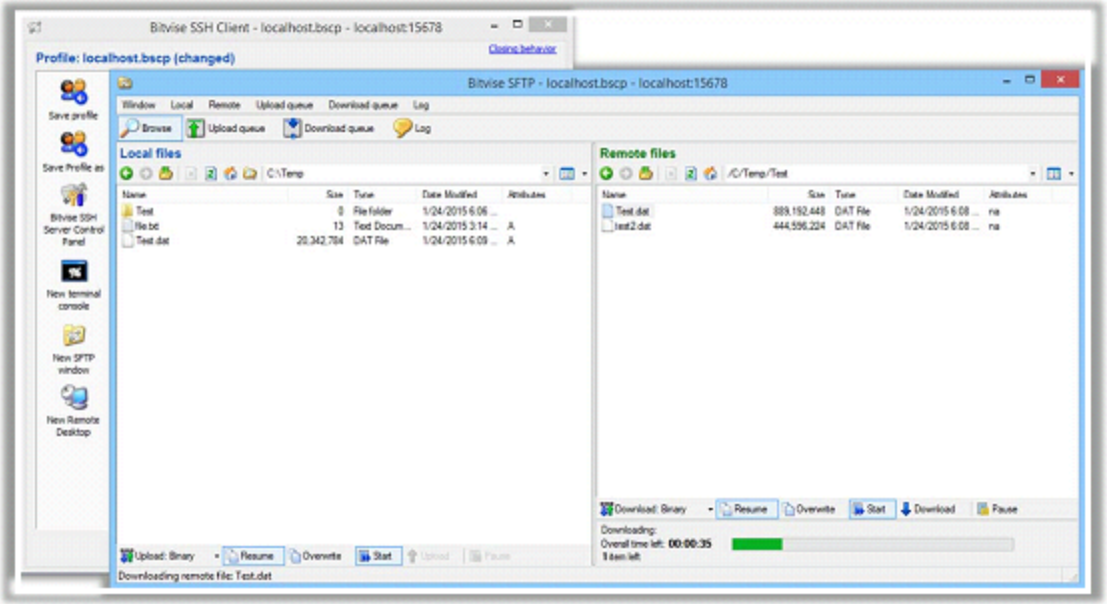
- This forwards the local port 5000 to port 25 on certifiedhacker.com encrypted
- Simply point your email client to use localhost:5000 as the SMTP server



SSH Tunneling Tool: Bitvise and Secure Pipes

Bitvise

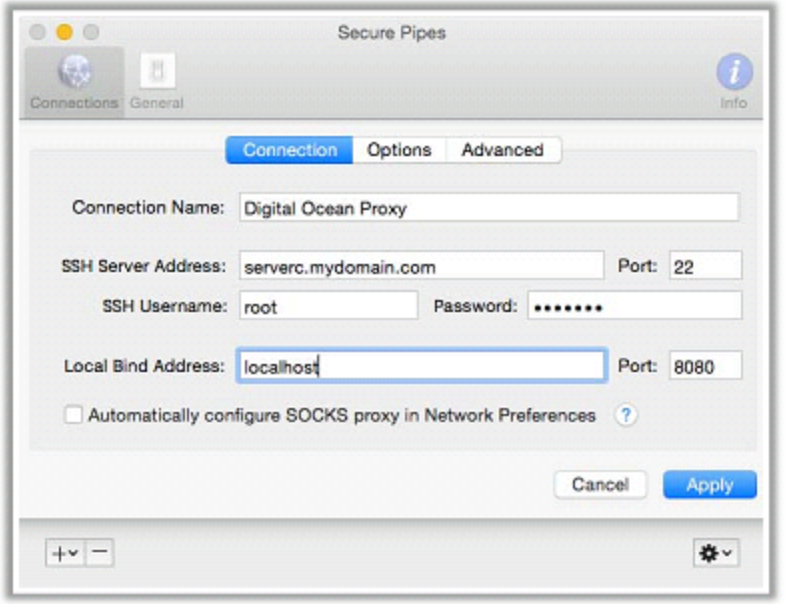
- Bitvise SSH Server provides secure remote login capabilities to Windows workstations and servers
- SSH Client includes powerful tunnelling features including dynamic port forwarding through an integrated proxy, and also remote administration for the SSH Server



https://www.bitvise.com

Secure Pipes

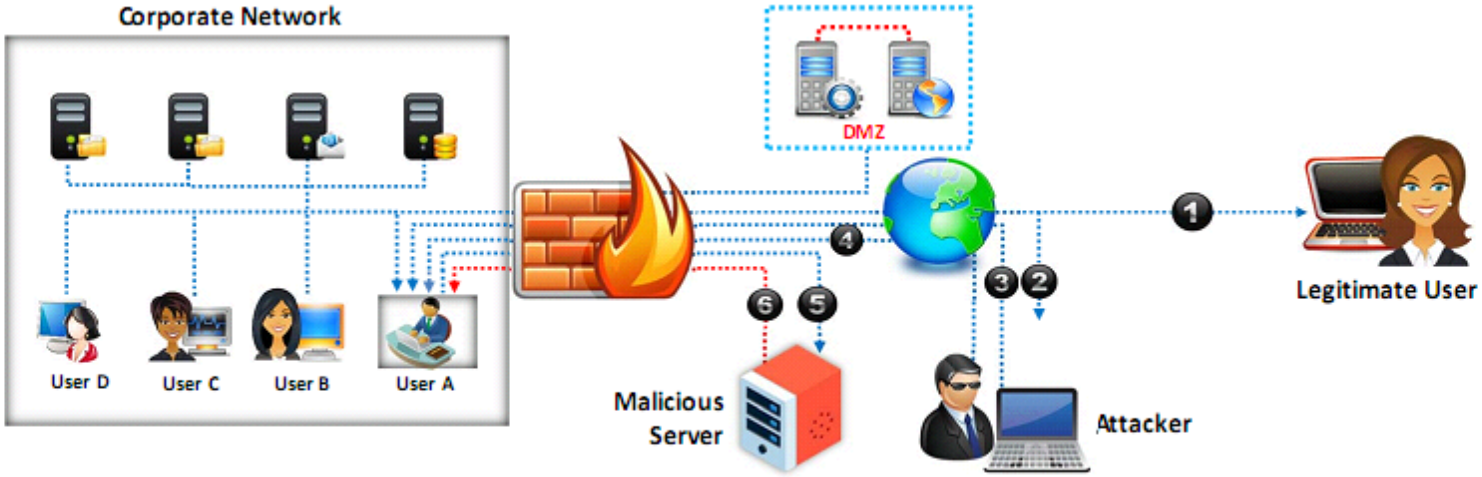
- Secure Pipes makes managing SSH tunnels simple
- It selectively opens up access to application ports normally not easily accessible due to network or service provider configuration restrictions



https://www.opoet.com

Bypassing Firewall through External Systems

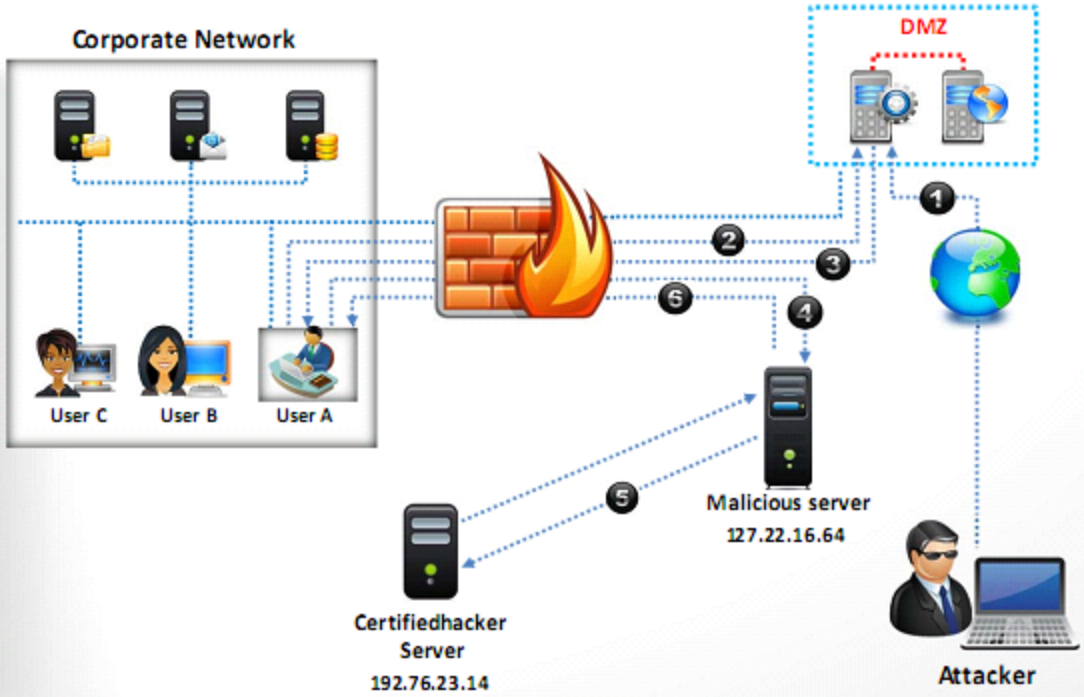
1. Legitimate user works with some **external system** to access the corporate network
2. Attacker sniffs the **user traffic**, steals the **session ID** and **cookies**
3. Attacker **accesses the corporate network** bypassing the firewall and gets **Windows ID** of the running Mozilla process on the user's system
4. Attacker then issues an **openURL() command** to the found window
5. User's web browser is redirected to the **attacker's Web server**
6. The malicious codes embedded in the attacker's web page are **downloaded and executed** on the user's machine



Bypassing Firewall through MITM Attack

In MITM attacks, attackers **make use of DNS servers and routing techniques** to bypass firewall restrictions

1. Attacker performs **DNS server poisoning**
2. User A requests for www.certifiedhacker.com to the **corporate DNS server**
3. Corporate DNS server sends the **IP address** (127.22.16.64) **of the attacker**
4. User A accesses the **attacker's malicious server**
5. Attacker connects with the **real host and tunnels the user's HHTP traffic**
6. The malicious codes embedded in the attacker's web page are **downloaded and executed** on the user's machine



Bypassing Firewall through Content

In this method, the attacker **sends the content containing malicious code** to the user and tricks him/her to open it so that the malicious code can be executed

Examples:

Sending an email containing a malicious executable file or Microsoft office document capable of exploiting **macro bypass exploit**

There are many file formats that can be used as **malicious content carrier**

Bypassing WAF using XSS Attack

- XSS attack exploits vulnerabilities that occur while processing **input parameters** of the end users and the **server responses** in a web application
- Attackers inject **malicious HTML code** in the victim website to **bypass the WAF**
- Consider the following XSS payload

```
<scirpt>alert("XSS")</script>
```

Using ASCII values to bypass WAF

- After replacing XSS payload with its equivalent ASCII values

```
<scirpt>String.fromCharCode(97, 108, 101, 114, 116, 40, 34, 88, 83, 83, 34, 41)</script>
```

Using Hex Encoding to bypass WAF

- After encoding the XSS payload,

```
%3C%73%63%69%72%70%74%3E%61%6C%65%72%74%28%22%58%53%53%22%29%3C%2F%73%63%72%69%70%74%3E
```

Using Obfuscation to bypass WAF

- After encoding the XSS payload,

```
<sCiRPt>aLeRT("XSS")</sCriPT>
```

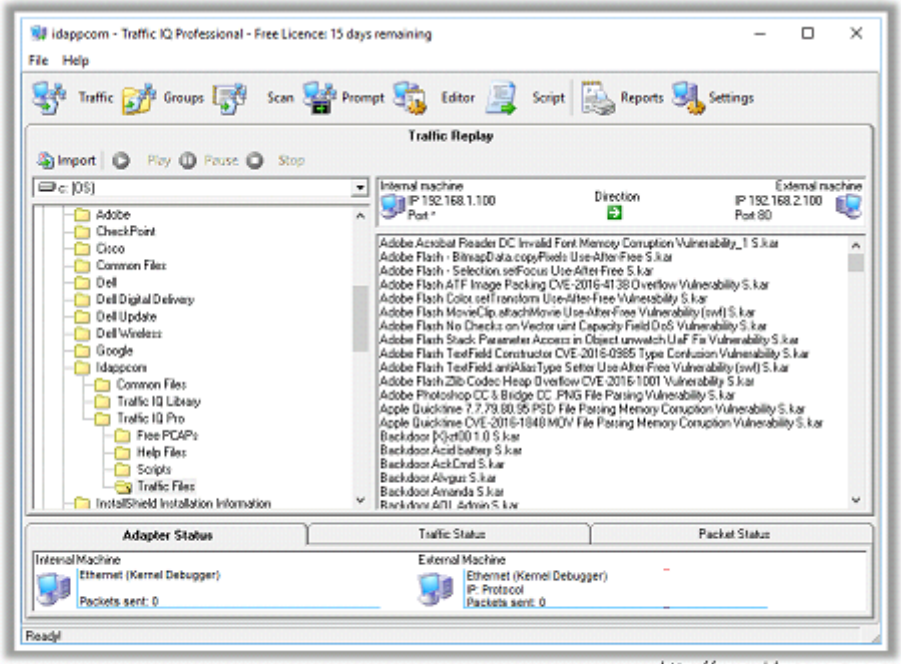
Module Flow

1. IDS, Firewall and Honeypot Concepts
2. IDS, Firewall and Honeypot Solutions
3. Evading IDS
4. Evading Firewalls
5. IDS/Firewall Evading Tools
6. Detecting Honeypots
7. IDS/Firewall Evasion Countermeasures
8. Penetration Testing

IDS/Firewall Evasion Tools

Traffic IQ Professional

Traffic IQ Professional enables security professionals to audit and validate the behavior of security devices by generating the standard application traffic or attack traffic between two virtual machines



http://www.idappcom.com

Hotspot Shield
https://www.hotspotshield.com

FTester
https://inversepath.com

Snare Agent for Windows
https://www.intersectalliance.com

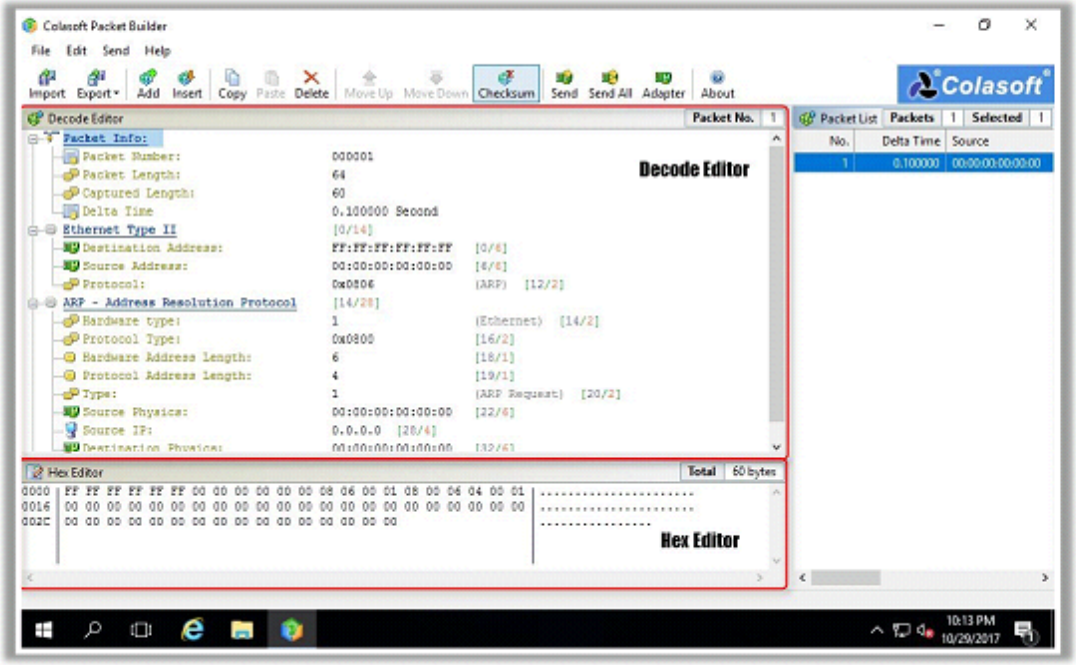
Tomahawk
http://tomahawk.sourceforge.net

Atelier Web Firewall Tester
http://www.atelierweb.com

Packet Fragment Generator Tools

Colasoft Packet Builder

Colasoft packet builder is a network packet crafter, packet generator or packet editor that network professionals use to **build (or craft) all types of custom networks**



http://www.colasoft.com

- **CommView**
  https://www.tamos.com
- **NetScanTools Pro**
  https://www.netscantools.com
- **Ostinato**
  http://ostinato.org
- **WAN Killer**
  https://www.solarwinds.com
- **WireEdit**
  https://wireedit.com

Module Flow

1. IDS, Firewall and Honeypot Concepts
2. IDS, Firewall and Honeypot Solutions
3. Evading IDS
4. Evading Firewalls
5. IDS/Firewall Evading Tools
6. Detecting Honeypots
7. IDS/Firewall Evasion Countermeasures
8. Penetration Testing

Detecting Honeypots

- Attackers can determine the **presence of honeypots** by probing the services running on the system
- Attackers craft **malicious probe packets** to scan for services such as HTTP over SSL (HTTPS), SMTP over SSL (SMPTS), and IMAP over SSL (IMAPS)
- Ports that show a particular service running but deny a **three-way handshake connection** indicate the presence of a honeypot

Tools to probe honeypots:

- **Send-safe Honeypot Hunter**
- **Nessus**
- **Hping**

Note: Attackers can also defeat the purpose of honeypots by using multi-proxies (TORs) and hiding their conversation using encryption and steganography techniques

Detecting and Defeating Honeypots

Detecting presence of Layer 7 Tar Pits	Look at the **latency of the response** from the service
Detecting presence of Layer 4 Tar Pits	Analyze the **TCP window size**, where tar pits continuously acknowledge incoming packets even though the TCP window size is reduced to zero
Detecting presence of Layer 2 Tar Pits	If an attacker is present on the same network as the Layer 2 tar pits, then the attacker can detect the presence of this daemon by looking at the **responses with unique MAC address** 0:0:f:ff:ff:ff which act as a kind of black hole
Detecting HoneyPots running on VMware	Look at the **IEEE standards for the current range of MAC addresses** assigned to VMWare Inc.
Detecting presence of Honeyd Honeypot	Perform time based **TCP Finger printing** methods (SYN Proxy behavior)

Detecting and Defeating Honeypots (Cont'd)

Detecting presence of User-Mode Linux (UML) Honeypot	Analyze the files such as **/proc/mounts**, **/proc/interruppts**, and **/proc/cmdline**, etc. which contain UML-specific information
Detecting presence of Sebek-based Honeypots	Sebek logs everything that is accessed via read() before transferring to the network causing the congestion effect. Analyze the **congestion in the network layer**
Detecting presence of Snort_inline Honeypot	Analyze the **outgoing packets** by capturing the Snort_inline modified packet through another host system and identifying the packet modification
Detecting presence of Fake AP	Fake access points only send beacon frames but do not generate any fake traffic on the access points and an attacker can **monitor the network traffic** and easily notice the presence of Fake AP
Detecting presence of Bait and Switch Honeypots	Look at specific **TCP/IP parameters** like the Round-Trip Time (RTT), the Time To Live (TTL), the TCP timestamp, etc.

Honeypot Detection Tool: Send-Safe Honeypot Hunter

Send-Safe Honeypot Hunter is a tool designed for checking **lists of HTTPS and SOCKS proxies** for "honey pots"

Features:

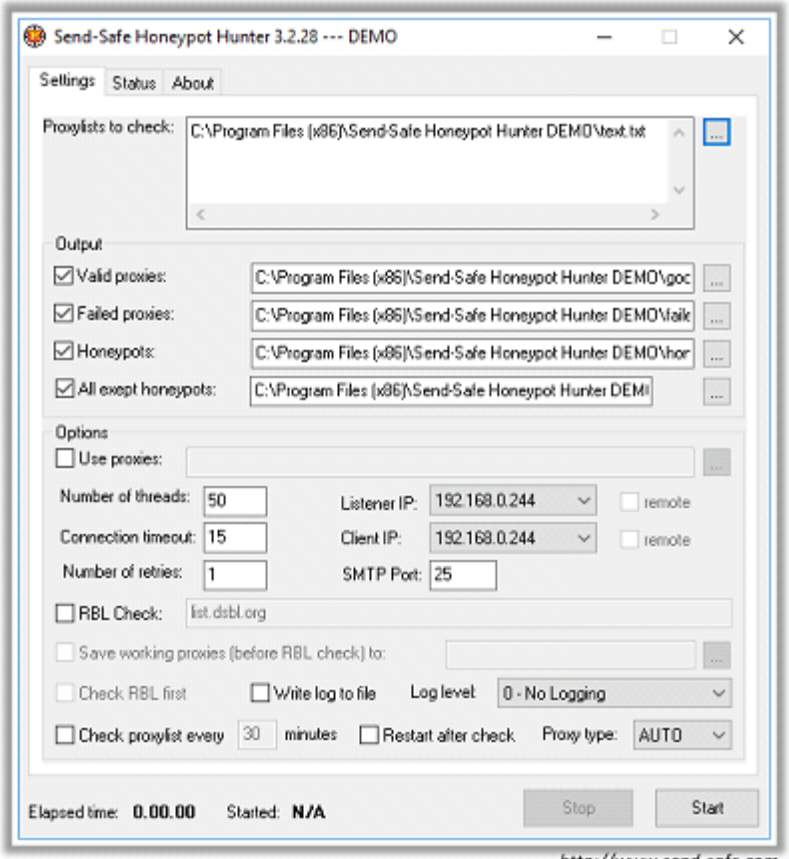
01 Checks lists of **HTTPS**, **SOCKS4**, and **SOCKS5 proxies** with any ports

02 Checks **several remote or local proxylists** at once

03 Can upload **"Valid proxies"** and **"All except honeypots"** files to FTP

04 Can process **proxylists** automatically every specified period of time

05 May be used for **usual proxylist** validating as well



http://www.send-safe.com

Module Flow

1. IDS, Firewall and Honeypot Concepts
2. IDS, Firewall and Honeypot Solutions
3. Evading IDS
4. Evading Firewalls
5. IDS/Firewall Evading Tools
6. Detecting Honeypots
7. IDS/Firewall Evasion Countermeasures
8. Penetration Testing

How to Defend Against IDS Evasion

1. Shut down switch ports associated with known attack hosts
2. Perform an in-depth analysis of ambiguous network traffic for all possible threats
3. Use TCP FIN or Reset (RST) packet to terminate malicious TCP sessions
4. Look for the nop opcode other than 0x90 to defend against the polymorphic shellcode problem
5. Train users to identify attack patterns and regularly update/patch all the systems and network devices
6. Deploy IDS after a thorough analysis of network topology, nature of network traffic, and the number of host to monitor
7. Use a traffic normalizer to remove potential ambiguity from the packet stream before it reaches the IDS
8. Ensure that IDSs normalize fragmented packets and allow those packets to be reassembled in the proper order
9. Define DNS server for client resolver in routers or similar network devices
10. Harden the security of all communication devices such as modems, routers, switches, etc.
11. If possible, block ICMP TTL expired packets at the external interface level and change the TTL field to a large value
12. Regular update of antivirus signature database
13. Use a traffic normalization solution at the IDS to prevent the system against evasions
14. Store the attack information (attacker IP, victim IP, timestamp) for future analysis

How to Defend Against Firewall Evasion

1. Configuration of the firewall should be done in such a way that the IP address of an intruder should be filtered out
2. Set the firewall ruleset to deny all traffic and enable only the services required
3. If possible, create a unique user ID to run the firewall services. Rather than running the services using the administrator or root IDs
4. Configure a remote syslog server and apply strict measures to protect it from malicious users
5. Monitor firewall logs at regular intervals and investigate all suspicious log entries found
6. By default, disable all FTP connections to or from the network
7. Catalog and review all inbound and outbound traffic allowed through the firewall
8. Run regular risk queries to identify vulnerable firewall rules
9. Monitor user access to firewalls and control who can modify the firewall configuration
10. Specify the source and destination IP addresses as well as the ports
11. Notify the security policy administrator on firewall changes and document them
12. Control physical access to the firewall
13. Take regular backups of the firewall ruleset and configuration files
14. Schedule regular firewall security audits

Module Flow

1. IDS, Firewall and Honeypot Concepts
2. IDS, Firewall and Honeypot Solutions
3. Evading IDS
4. Evading Firewalls
5. IDS/Firewall Evading Tools
6. Detecting Honeypots
7. IDS/Firewall Evasion Countermeasures
8. Penetration Testing

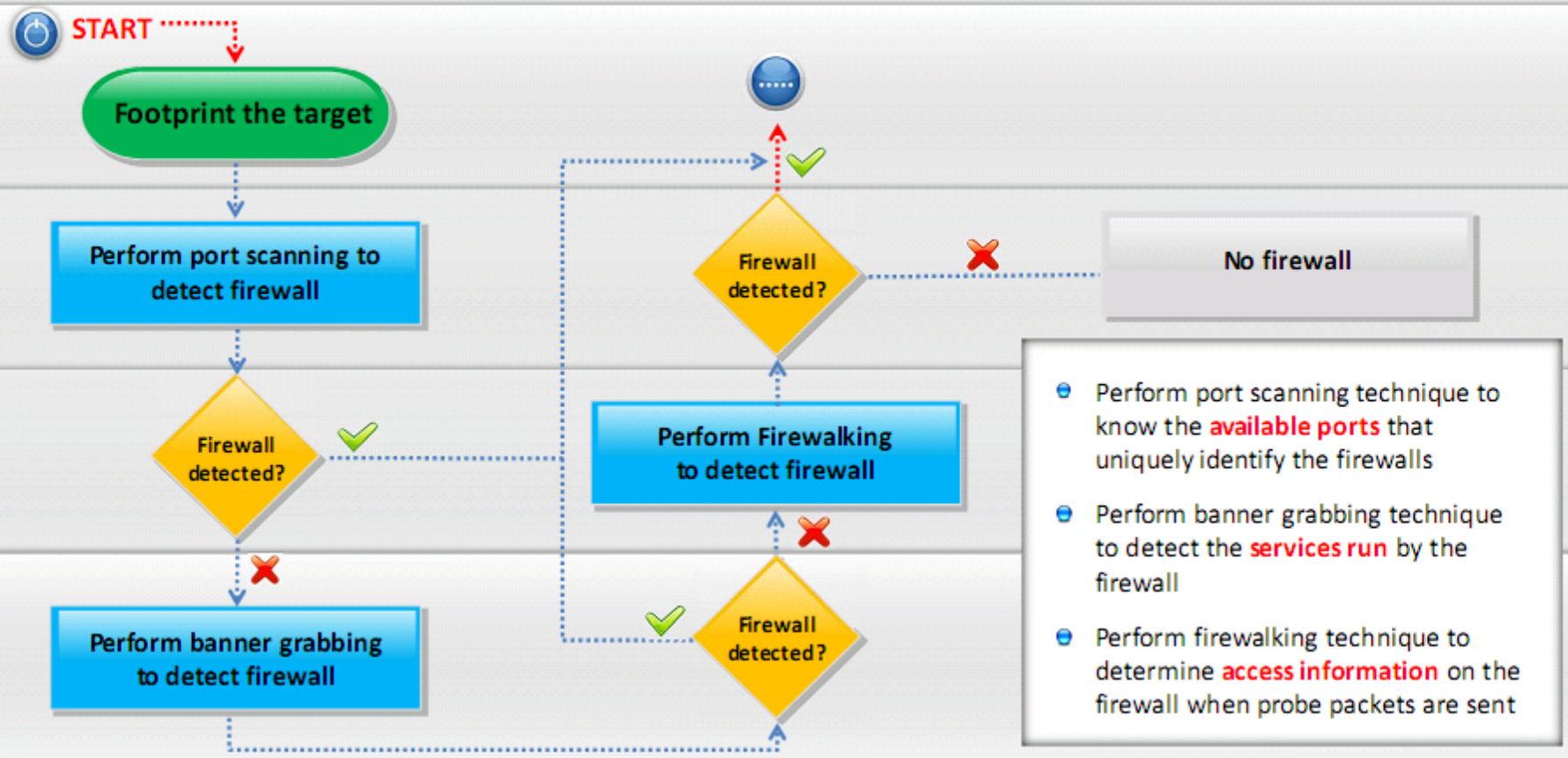
Firewall/IDS Penetration Testing

Firewall/IDS penetration testing helps in evaluating the Firewall and IDS for **ingress and egress traffic filtering capabilities**

Why Firewall/IDS Pen Testing?

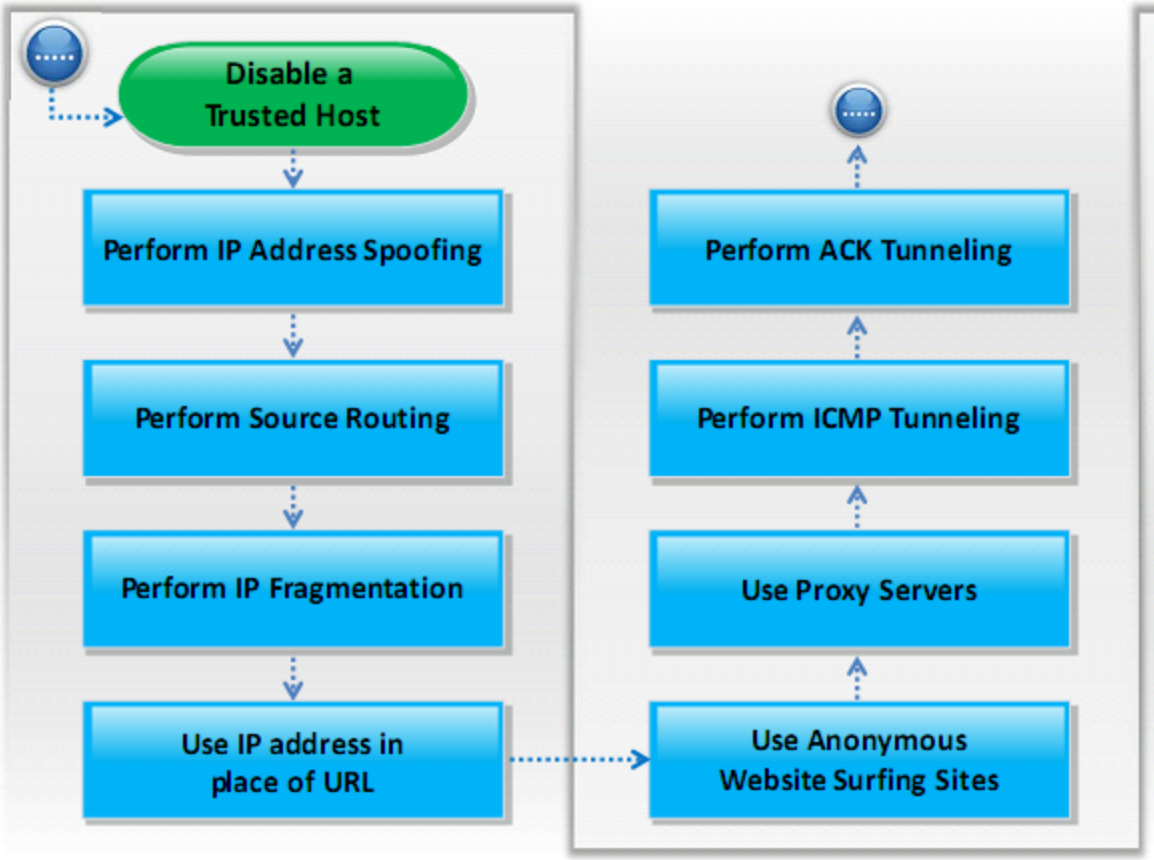
1. To check if firewall/IDS properly enforces an **organization's firewall/ IDS policy**
2. To check if the **IDS and firewalls** enforces organization's network security policies
3. To check if the firewall/IDS is good enough to **prevent the external attacks**
4. To check the effectiveness of the **network's security perimeter**
5. To check the **amount of network information accessible** to an intruder
6. To check the firewall/IDS for **potential breaches of security** that can be exploited
7. To evaluate the **correspondence of firewall/IDS rules** with respect to the actions performed by them
8. To verify whether the **security policy is correctly enforced** by a sequence of firewall/IDS rules or not

Firewall Penetration Testing



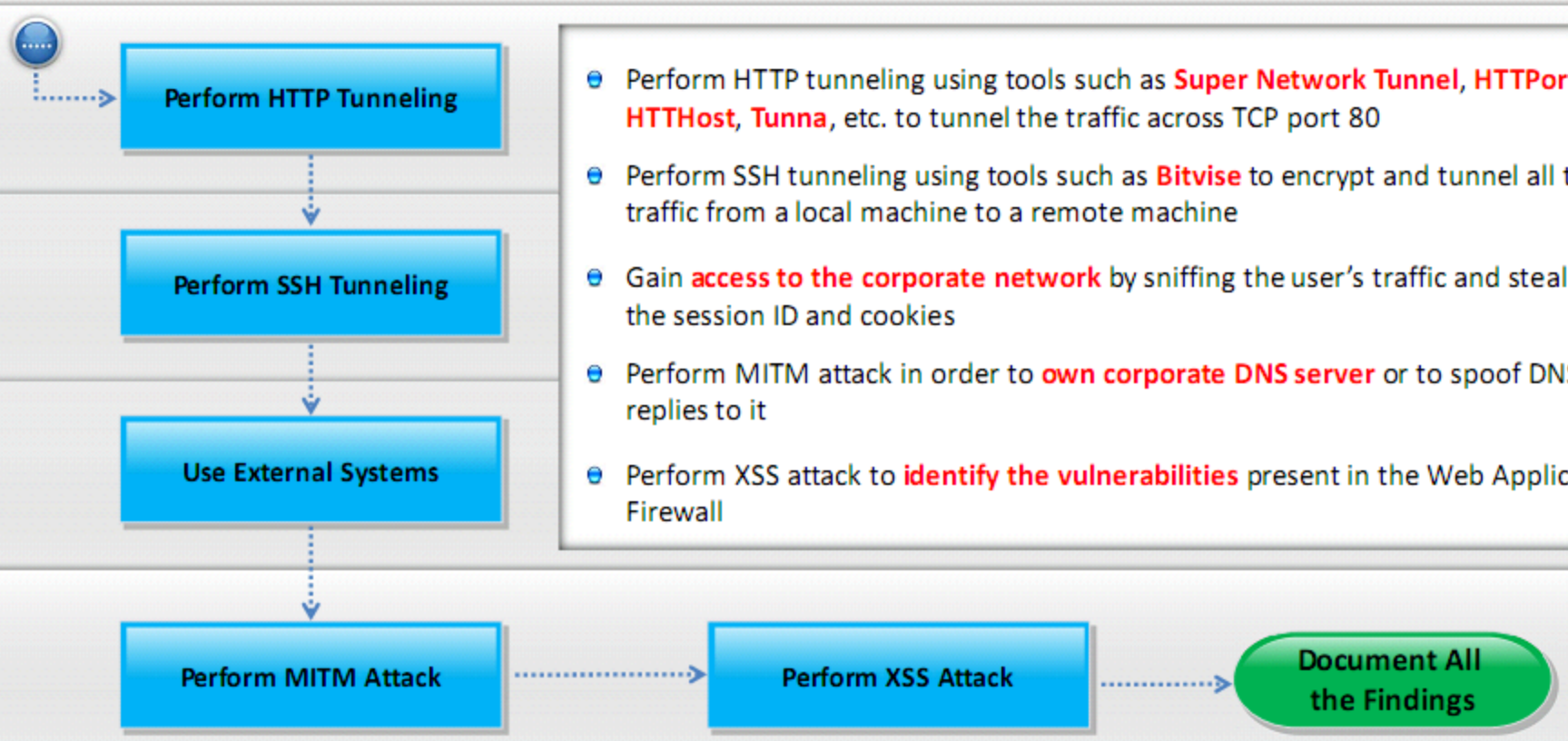
- Perform port scanning technique to know the **available ports** that uniquely identify the firewalls
- Perform banner grabbing technique to detect the **services run** by the firewall
- Perform firewalking technique to determine **access information** on the firewall when probe packets are sent

Firewall Penetration Testing (Cont'd)



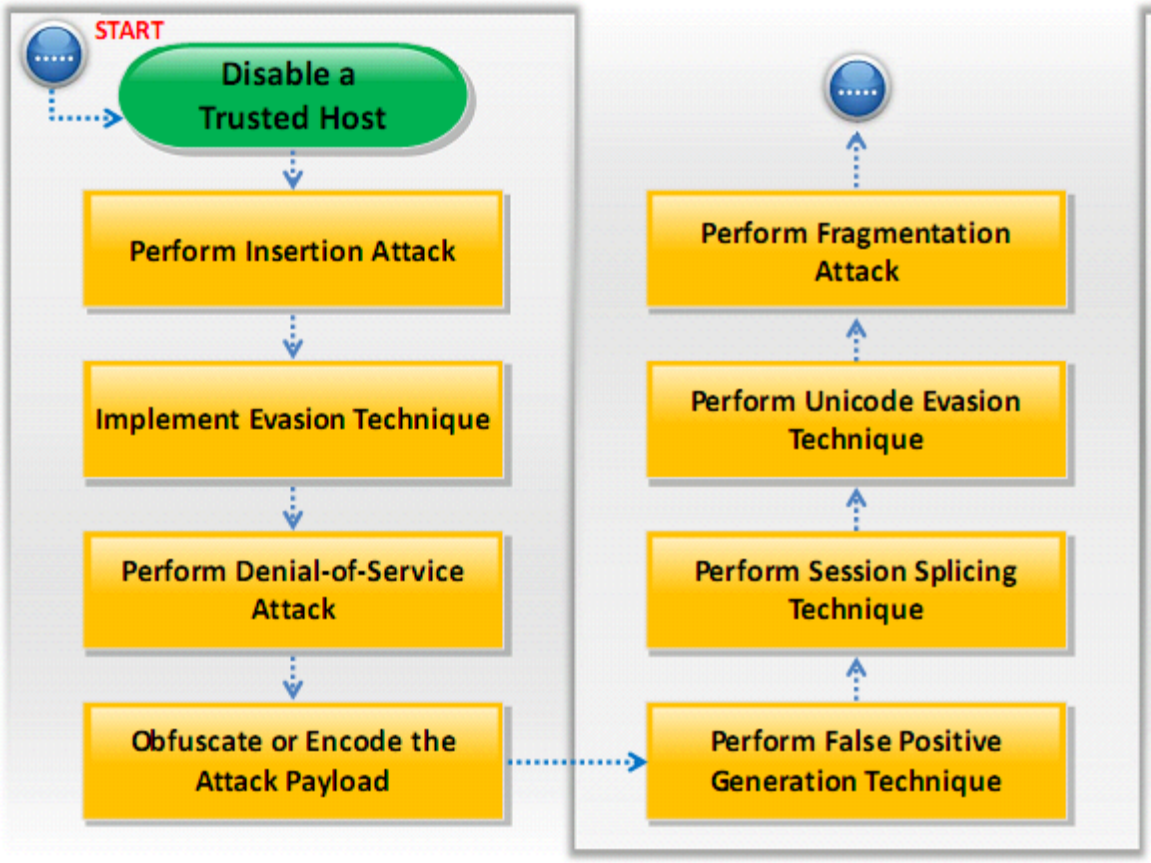
- Perform **IP address spoofing** to gain unauthorized access to a computer or a network
- Perform source routing to **designate the packet route** in order to bypass the firewall
- Perform **fragmentation attack** to force the TCP header information into the next fragment in order to bypass the firewall
- Type the **IP address directly in browser's address bar** in place of typing the blocked website's domain name to evade the firewall restriction
- **Use proxy servers** that block the actual IP address and display another thereby allowing **access to the blocked website**
- Perform **ICMP tunneling** to tunnel a backdoor application in the data portion of ICMP Echo packets
- Perform **ACK tunneling** using tools such as **AckCmd** to tunnel backdoor application with TCP packets with the ACK bit set

Firewall Penetration Testing (Cont'd)



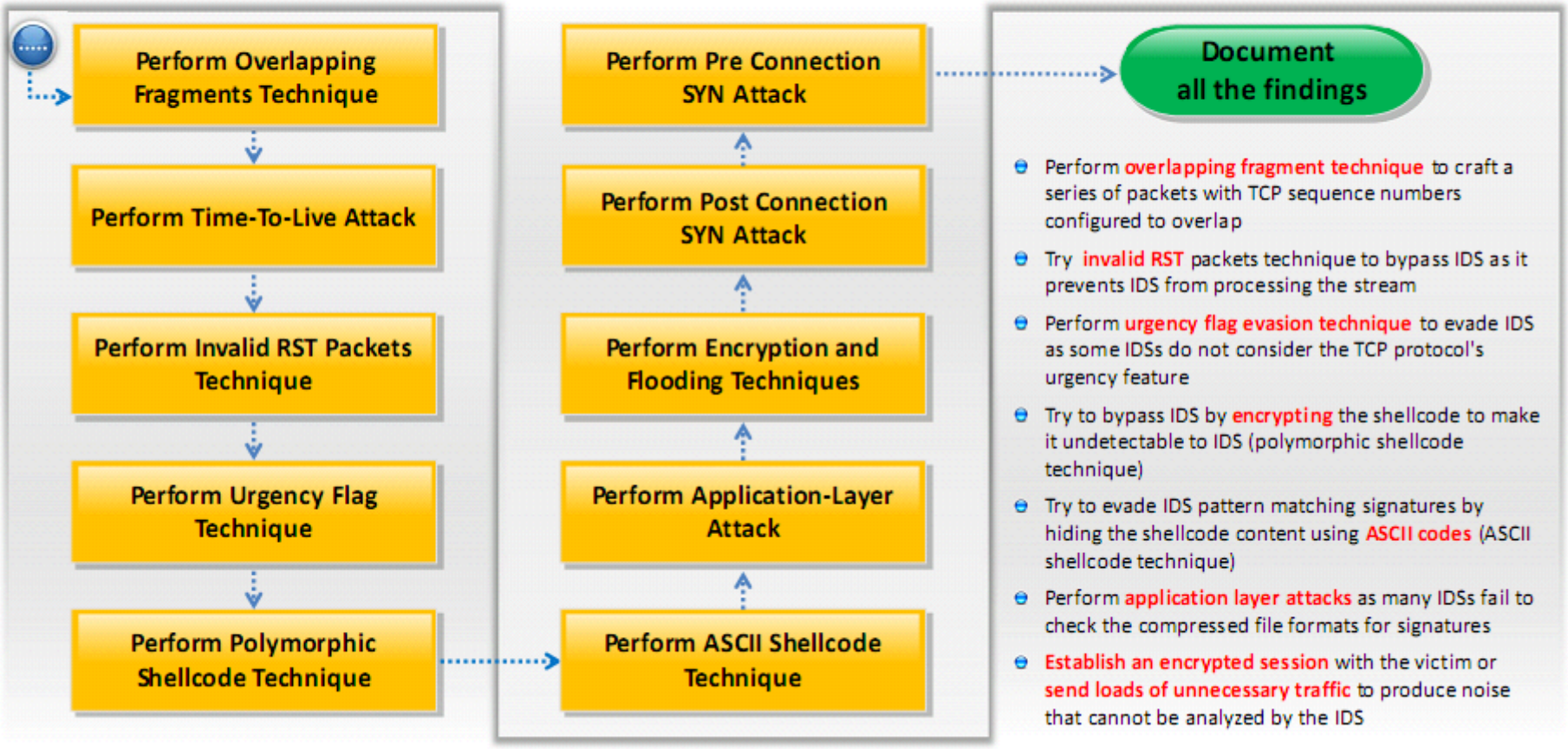
- Perform HTTP tunneling using tools such as **Super Network Tunnel**, **HTTPort**, **HTTHost**, **Tunna**, etc. to tunnel the traffic across TCP port 80
- Perform SSH tunneling using tools such as **Bitvise** to encrypt and tunnel all the traffic from a local machine to a remote machine
- Gain **access to the corporate network** by sniffing the user's traffic and stealing the session ID and cookies
- Perform MITM attack in order to **own corporate DNS server** or to spoof DNS replies to it
- Perform XSS attack to **identify the vulnerabilities** present in the Web Application Firewall

IDS Penetration Testing



- Perform **obfuscating technique** to encode attack packets that IDS would not detect but an IIS web server would decode and become attacked
- Try to bypass IDS by hiding attack traffic in a large volume of **false positive alerts** (false positive generation attack)
- Use **session splicing technique** to bypass IDS by keeping the session active for a longer time than the IDS reassembly time
- Try **Unicode representations** of characters to evade the IDS signature
- Perform **fragmentation attack with** IDS fragmentation reassembly timeout **less and more than** that of the victim

IDS Penetration Testing (Cont'd)



Module Summary

- An intrusion detection system (IDS) is a security software or hardware device which inspects all inbound and outbound network traffic for suspicious patterns that may indicate a network or system security breach
- Firewalls are software and/or hardware-based systems designed to prevent unauthorized access to or from a private network
- A honeypot is an information system resource that is expressly set up to attract and trap people who attempt to penetrate an organization's network
- Attackers can evade IDS and firewalls using various evasion techniques such as session splicing, fragmentation, Time-to-Live, IP address spoofing, ICMP, ACK, HTTP, and SSH tunneling, etc.
- Attackers can determine the presence of honeypots by probing the services running on the system
- Firewall/IDS penetration testing helps in evaluating the Firewall and IDS for ingress and egress traffic filtering capabilities