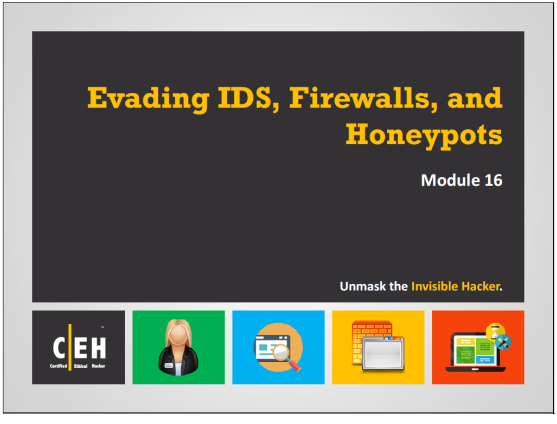
Evading IDS, Firewalls, and Honeypots

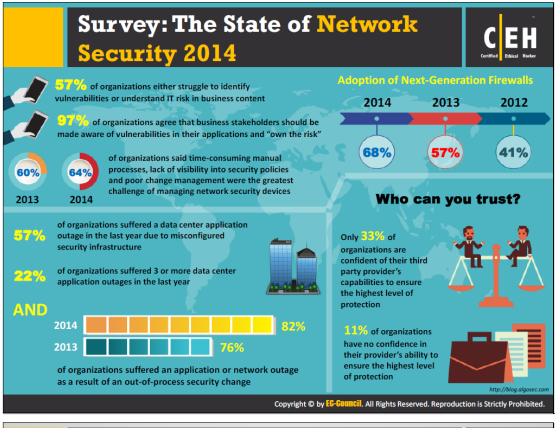
Module 16

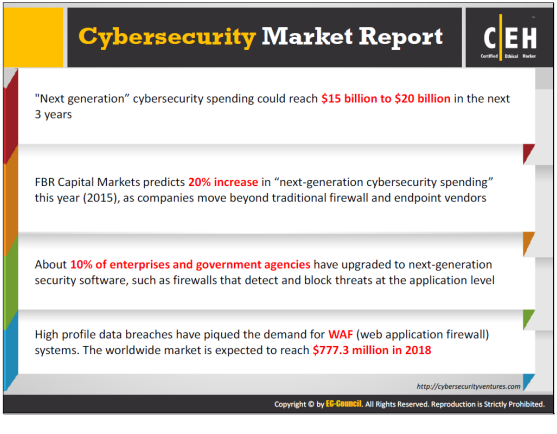


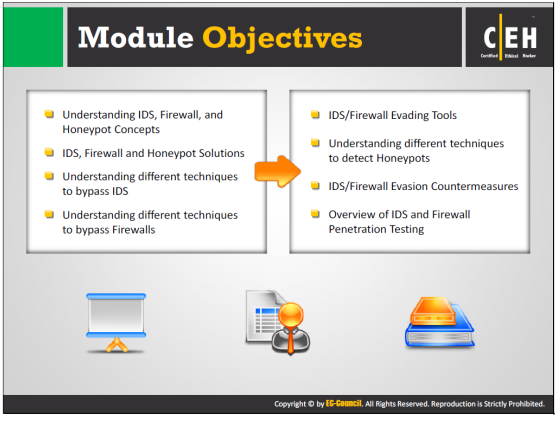
Ethical Hacking and Countermeasures V9

Module 16: Evading IDS, Firewalls, and Honeypots

Exam 312-50

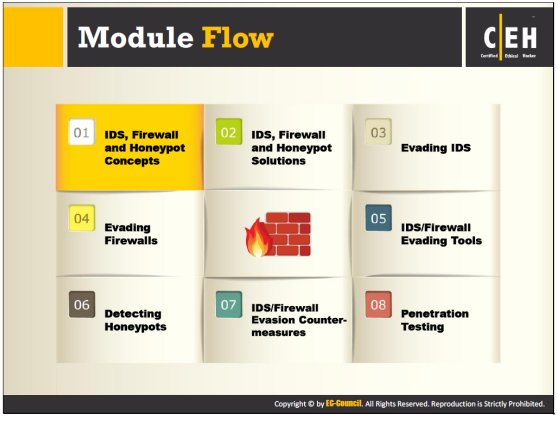




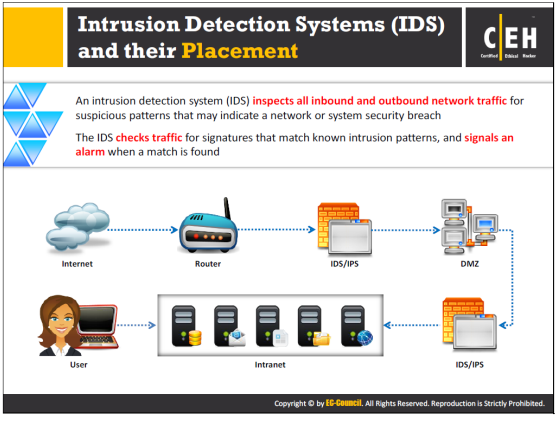


Adoption of Internet use throughout the business world has boosted network usage in general. To protect their networks, organizations are using various network security measures such as firewalls, intrusion detection system (IDS), intrusion prevention system (IPS), and "honeypots." Networks are the most preferred targets of hackers for compromising organizations' security, and attackers continue to find new ways to breach network security and attack these targets.

This module provides a deep insight into various network security technologies, such as IDS, firewalls, and honeypots. It explains the operations of these components as well as the various techniques attackers use to evade them. This module discusses the various tools and techniques used in evading network security, and provides countermeasures necessary to prevent such attacks. It also includes an overview of firewall pen testing an ethical hacker should follow to increase network security.



To understand how an attacker evades the security of firewalls, IDS, and honeypots, the ethical hacker should have an idea about their functions, role, placement, and design implemented to protect an organization's network. This section provides an overview of these basic concepts.



An Intrusion Detection System (IDS) is security software or hardware device used to monitor, detect, and protect networks or system from malicious activities; it alerts the concern security personnel immediately upon detecting intrusions. Intrusion detection systems are highly useful as IDS monitors both inbound/outbound traffic of the network and checks for suspicious activities continuously.

Main Functions of IDS:

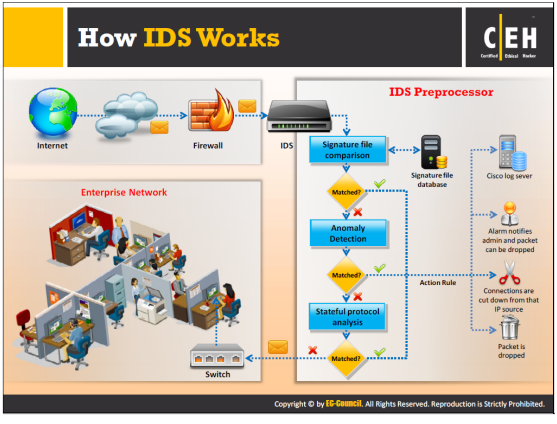
- An intrusion detection system (IDS) gathers and analyzes information from within a computer or a network, to identify the possible violations of security policy, including unauthorized access, as well as misuse.
- An IDS is also referred to as a "packet-sniffer," which intercepts packets traveling along various communication mediums and protocols, usually TCP/IP.
- The packets are analyzed after they are captured.
- An IDS evaluates traffic for suspected intrusions, and signals an alarm when one is found.

Where the IDS do resides in the network?

One of the most common places to deploy IDS is near the firewall. Depending on the traffic to be monitor, IDS is placed outside/inside the firewall to monitor suspicious traffic originated from outside/inside the network. Placed inside, the IDS will be ideal if it is near a DMZ;

however, the best practice is to use a layered defense by deploying one IDS in front of the firewall and another one behind the firewall in the network.

Before deploying the IDS, it is essential to analyze network topology, understand how the traffic flows to and from the resources that an attacker can use to gain access to the network, and identify the critical components that will be possible target by many of the attacks against the network. Even after deciding the position of the IDS in the network, its configuration would maximize the effectiveness of network protection.

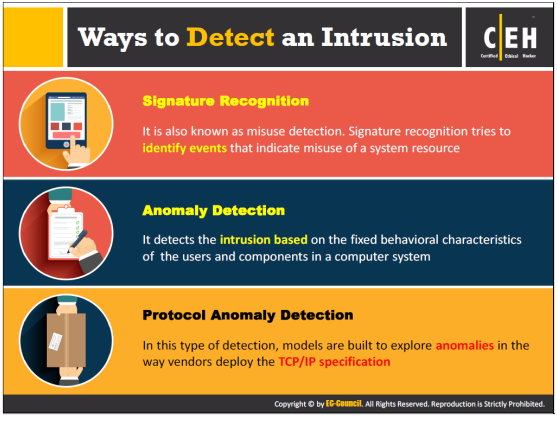


The main purpose of the IDS is to recognize and provide real-time monitoring of intrusions. Additionally, reactive IDSs (and IPs) can intercept, respond to, and/or prevent the intrusions.

An IDS works in the following way:

- IDSs have sensors to detect malicious signatures in data packets, and some advanced IDSs have behavioral activity detection, to determine malicious traffic behavior. Even if the packet signatures do not match perfectly with the signatures in the IDS signature database, the activity detection system can alert administrators about possible attacks.
- If the signature matches, the IDS performs predefined actions such as terminating connection, blocking the IP address, dropping the packet, and/or signaling an alarm to notify the administrator.
- When signature matches, anomaly detection will skip; otherwise, the sensor may analyze traffic patterns for an anomaly.
- When the packet passes all tests, the IDS will forward it into the network.

The administrator must also be able to identify the methods and techniques used by the intruder and the source of attack.



An IDS uses three methods to detect intrusion in the network.

Signature Detection

Signature recognition, also known as misuse detection, tries to identify events that indicate an abuse of a system or network. This technique involves first creating models of possible intrusions and then comparing these models with incoming events to make a detection decision. The signatures for IDS were created on the assumption that the model must detect an attack without disturbing normal system traffic. Attacks—and only attacks—should match the model; otherwise, false alarms could occur.

- Signature-based intrusion detection compares incoming or outgoing network packets with the binary signatures of known attacks, using simple pattern-matching techniques to detect intrusion. Attackers can define a binary signature for a specific portion of the packet, such as TCP flags.
- Signature recognition can detect known attacks. However, there is a possibility that other innocuous packets might also contain the same signature, which will trigger a false positive alert.
- Improper signatures may trigger false alerts. To detect misuse, the number of signatures required is huge. The more the signatures, the greater the chances are of the IDS detecting attacks, although traffic may incorrectly match with the signatures, thus impeding system performance.

- An increase in signature data consumes more network bandwidth. IDS systems compare signatures of data packets against those in the signature database. An increase in the number of signatures in the database could result in the dropping of certain packets.
- New virus attacks such as ADMutate and Nimda create the need for multiple signatures for a single attack. Changing a single bit in some attack strings can invalidate a signature created for that attack. Therefore, it requires creating entirely new signatures to detect the same attack.
- Despite problems with signature-based intrusion detection, such systems are popular and work well when configured correctly and monitored closely.

Anomaly Detection

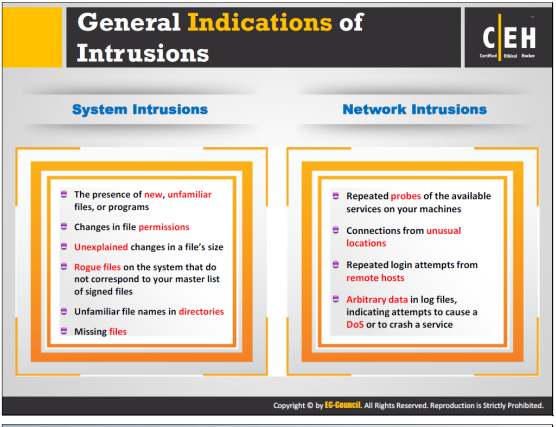
Anomaly detection, or "**not-use detection**," differs from the signature-recognition model. Anomaly detection consists of a database of anomalies. You can detect anomaly when any event occurs outside the tolerance threshold for of normal traffic. Therefore, any deviation from normal use is an attack. Creating a model of normal use is the most difficult task in creating an anomaly detector.

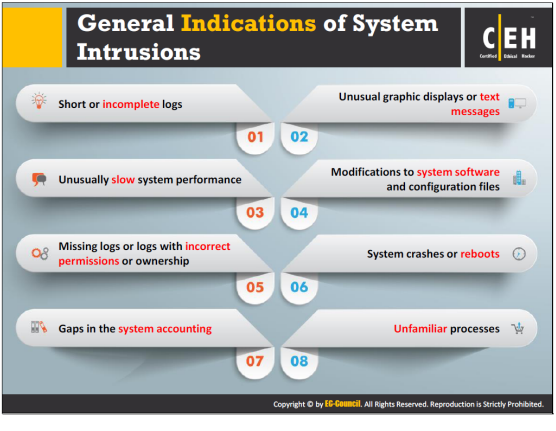
- In the traditional method of anomaly detection, important data are kept for checking variations in network traffic. However, in reality, there is some unpredictability in network traffic, and there are too many statistical variations, thus making these models imprecise. Some events labeled as anomalies might only be irregularities in network usage.
- In this type of approach, the inability to construct a model thoroughly on a normal network is of concern. These models should be used to check on specific networks.

Protocol Anomaly Detection

Protocol anomaly detection depends on the anomalies specific to a protocol. It identifies specific flaws between how vendors deploy the TCP/IP protocol. Protocols designs according to RFC specifications, which dictate standard handshakes to permit universal communication. The protocol anomaly detector can identify new attacks.

- There are new attack methods and exploits that violate protocol standards.
- A malicious anomaly signature is growing greatly. However, the network protocol, in comparison, is well defined and changing slowly. Therefore, the signature database should be updated frequently to detect attacks.
- Protocol anomaly detectors are different from the traditional IDS in how they present alarms.
- The best way to present alarms is to explain which part of the state system compromises. For this, IDS operators have to have a thorough knowledge of protocol design.





Intrusion attempts on networks or file systems can be identified by following some general indicators:

File System Intrusions

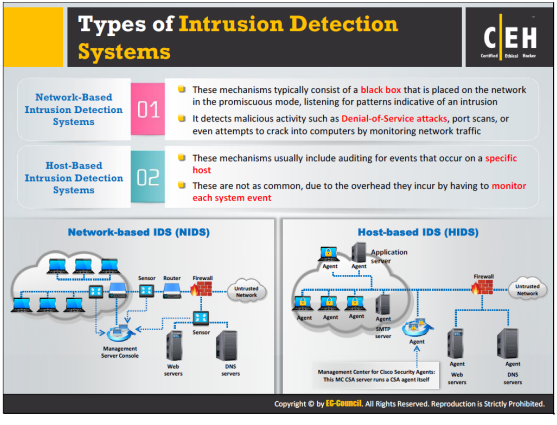
By observing system files, you can identify the presence of an intrusion. System files record the activities of the system. Any modification or deletion in the file attributes or the file itself is a sign that the system was a target of attack:

- If you find new, unknown files/programs on your system, then there is a possibility that your system has intruded. The system can be compromised to the point that it can, in turn, compromise other network systems.
- When an intruder gains access to a system, he or she tries to escalate privileges to gain administrative access. When the intruder obtains Administrator privilege, he/she could change file permissions, for example, from Read-Only to Write.
- Unexplained modifications in file size are also an indication of an attack. Make sure you analyze all of your system files.
- Presence of rogue suid and sgid files on your Linux system that do not match your master list of suid and sgid files could indicate an attack.
- You can identify unfamiliar file names in directories, including executable files with strange extensions and double extensions.
- Missing files are also a sign of a probable intrusion/attack.

Network Intrusions

Similarly, general indications of network intrusions include:

- Sudden increase in bandwidth consumption is an indication of intrusion
- Repeated probes of the available services on your machines
- Connection requests from IPs other than those in the network range, indicating that an unauthenticated user (intruder) is attempting to connect to the network
- You can identify repeated attempts to log in from remote machines
- A sudden influx of log data could indicate attempts at denial-of-service attacks, bandwidth consumption, and distributed denial-of-service attacks



There are two types of intrusion detection systems:

Network-Based Intrusion Detection Systems

Network-based intrusion detection systems (NIDSs) check every packet entering the network for the presence of anomalies and incorrect data. By limiting the firewall to drop large numbers of data packets, the NIDS checks every packet thoroughly. An NIDS captures and inspects all traffic. It generates alerts either at the IP or at the application-level based on the content. NIDSs are more distributed than host-based IDSs. The NIDS identifies the anomalies at the router and host level. It audits the information contained in the data packets, logging information of malicious packets, and assigns a threat level to each risk after receiving the data packets. The threat level enables the security team to be on alert. These mechanisms typically consist of a black box placed on the network in promiscuous mode, listening for patterns indicative of an intrusion.

Host-Based Intrusion Detection Systems

In the host-based system, the IDS analyze each system's behavior. Install Host-Based Intrusion Detection Systems (HIDSs) on any system ranging from a desktop PC to a server. The HIDS is more versatile than the NIDS. In addition to detecting unauthorized insider activity, host-based systems are also effective at detecting unauthorized file modification. HIDSs focuses on the changing aspects of local systems. The HIDS is also more platform-centric, with more focus on the Windows OS, but there are other HIDSs for UNIX platforms. These mechanisms usually

include auditing events that occur on a specific host. These are not as common, because of the overhead they incur by having to monitor each system event.

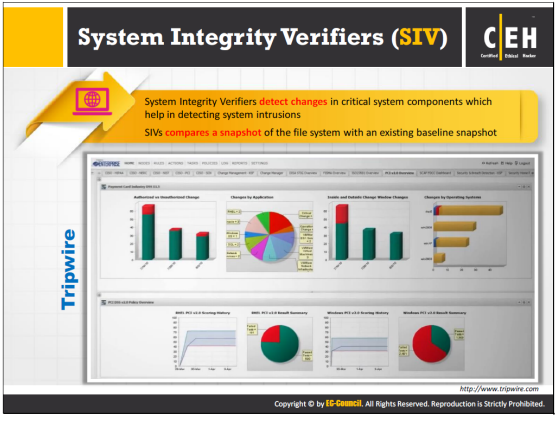
Organization networks also use two other IDS systems.

Log File Monitoring

A log file monitor (LFM) monitors log files created by network services. The LFM IDS searches through the logs and identifies malicious events. In a similar manner to NIDS, these systems look for patterns in the log files that suggest an intrusion. A typical example would be parsers for HTTP server log files that look for intruders who try well-known security holes, such as the "**phf**" attack. LFM tools, like "**Swatch**," for example, are typically programs that parse log files after an event has already occurred, such as failed login attempts.

File Integrity Checking

These mechanisms check for Trojan horses, or modified files, indicating an intruder has already been there. Tripwire is an example of a file integrity checking tool.

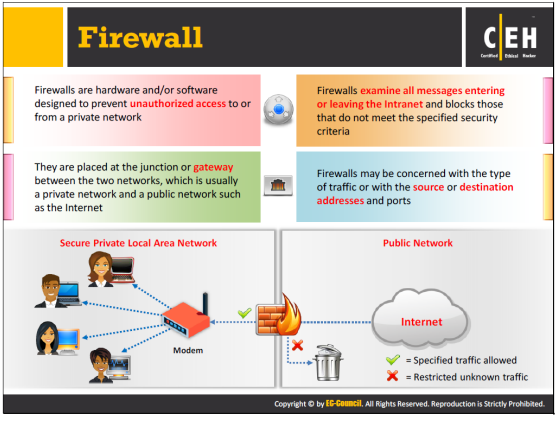


A System Integrity Verifier (SIV) monitors system files to determine whether an intruder has changed them. An integrity monitor watches key system objects for changes. For example, these tools monitor system files and registry keys for changes that might indicate an intrusion. Although they have limited functionality, integrity monitors can add an additional layer of protection to other forms of intrusion detection.

Features:

- Complete visibility and control of all change during continuous monitoring
- Automatic change audit and anomaly detection of threat indicators
- Remove potential human error

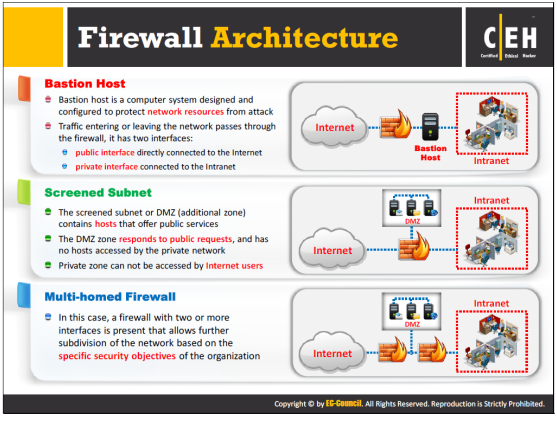
Source: *http://www.tripwire.com*



A firewall is a software- or hardware-based system located at the network gateway that protects the resources of a private network from users on other networks. Firewalls include set of tools that monitor the flow of traffic between networks. A firewall placed at the network level and working closely with a router filters all network packets to determine whether to forward them toward their destinations or not. Always install firewalls away from the rest of the network, so that no incoming request can get direct access to a private network resource. If configured properly, the firewall protects systems on one side of it from systems on the other side of the firewall.

- A firewall is an intrusion detection mechanism that is designed in accordance with each organization's security policy. Its settings can change to make appropriate changes to its functionality.
- Firewalls can configure to restrict incoming traffic to POP and SMTP and to enable email access. Certain firewalls block certain email services to secure against spam.
- A firewall can configure to check inbound traffic at a "check point," where a security audit is performed. It can also act as an active "phone tap" tool for identifying an intruder's attempt to dial into modems in a secured network. Firewall logs consist of logging information that reports to the administrator all attempts to access various services.

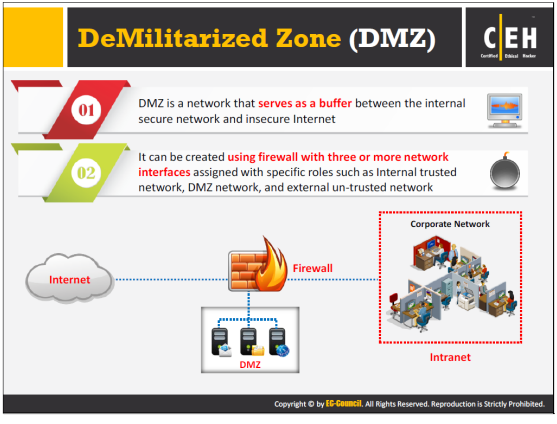
- The firewall verifies the incoming and outgoing traffic against firewall rules, and acts as a router to move data between networks. Firewalls allow or deny access requests made from one side of the firewall to services on the other side of the firewall.
- Identify all the attempts to log into the network for auditing. Unauthorized attempts can be identified by embedding an alarm that is triggered when an unauthorized user attempts to login. Firewalls can filter packets based on address and types of traffic. They identify the source, destination addresses, and port numbers when address filtering, and they identify types of network traffic when protocol filtering. Firewalls can identify the state and attributes of data packets.



Firewall architecture consists of the following elements:

- **Bastion Host:** The bastion host designed for defending the network against attacks. It acts as a mediator between inside and outside networks. A bastion host is a computer system designed and configured to protect network resources from attack.
- **Screened Subnet:** A screened subnet is a protected network created with a two- or three-homed firewall behind a screening firewall, and is a name commonly used to refer to the DMZ. When using a three-homed firewall, connect the first interface to the Internet, the second interface to the DMZ, and the third to the intranet.

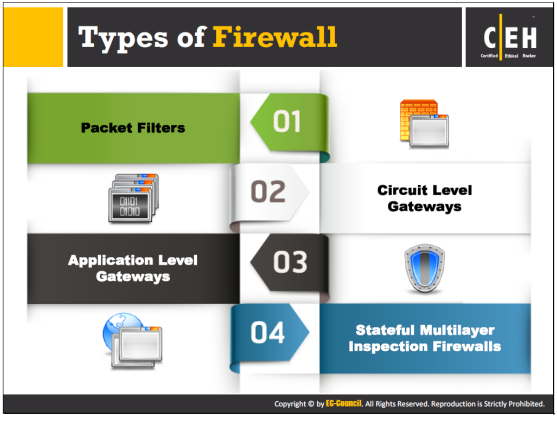
  The advantage of screening a subnet away from the intranet is that public requests can be responded to without allowing traffic into the intranet. A disadvantage with the three-homed firewall is that if it compromised, both the DMZ and intranet can also be compromised. A safer technique is to use multiple firewalls to separate the Internet from the screened subnet (DMZ), and then to separate the DMZ from the intranet.
- **Multi-homed Firewall:** A multi-homed firewall is a node with multiple NICs that connects to two or more networks. Connect each interface to the separate network segments logically and physically. A multi-homed firewall helps in increasing efficiency and reliability of an IP network. In the multi-homed firewall, more than three interfaces are present that allow for further subdividing the systems based on the specific security objectives of the organization. However, the model that adds depth of protection is the back-to-back firewall.



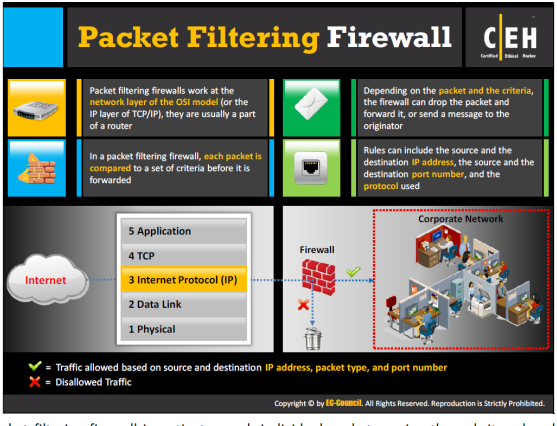
In computer networks, the DMZ is an area that hosts computer(s) or a small sub-network placed as a neutral zone between a particular company's internal network and untrusted external network to prevent outsider access to a company's private data. The DMZ serves as a buffer between the internal secure network and insecure Internet, as it adds a layer of security to the corporate LAN, thus preventing direct access to other parts of the network.

A DMZ is created using a firewall with three or more network interfaces assigned specific roles, such as an internal trusted network, a DMZ network, or an external untrusted network (Internet).

Any service such as mail, web, and FTP that provide access to external users can be placed in the DMZ. Although web servers that communicate with database servers cannot reside in the DMZ—as doing so could give outside users direct access to sensitive information—there are many ways in which the DMZ can be configured, according to specific network topologies and company requirements.



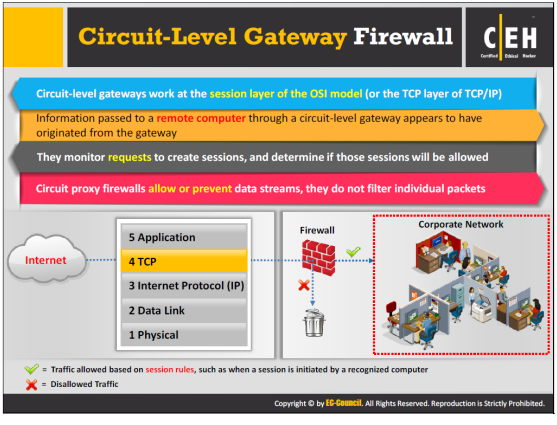
There are different types of firewalls, depending on where the communication is taking place, where traffic is intercepted in the network, the state that it traces, and so on. Taking into account the capabilities of the different types of firewalls offered, it is easy to choose and place an appropriate firewall to meet your security needs in the best possible way. Each type of firewall has its own advantages.



A packet filtering firewall investigates each individual packet passing through it and makes a decision whether to pass the packet or drop it. It works at the Internet Protocol (IP) layer of the TCP/IP model. Packet filter–based firewalls concentrate on individual packets, analyze their header information, and determine which way they need to directed.

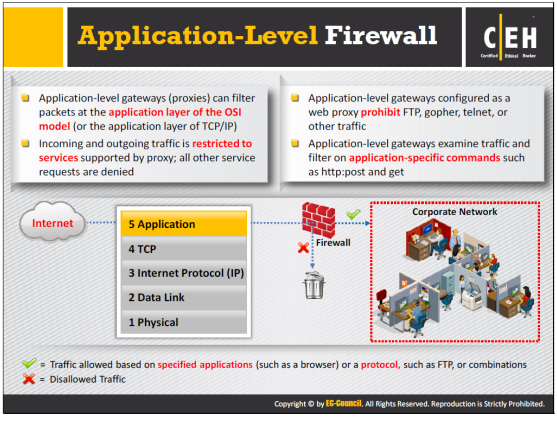
Traditional packet filters make this decision according to the following information in a packet:

- **Source IP address:** Used to check if the packet is coming from a valid source or not. The information about the source IP address can found from the IP header of the packet, which indicates the source system address.
- **Destination IP address:** Checks if the packet is going to the correct destination and check if the destination accepts these types of packets. The information about the destination IP address can found from the IP header of the packet, which has the destination address.
- **Source TCP/UDP port:** This is used to check the source port of the packet
- **Destination TCP/UDP port:** This is used to check the destination port, regarding the services to be allowed and the services to be denied.
- **TCP flag bits:** Used to check whether the packet has a SYN, ACK, or other bits set for the connection to be made.
- **Protocol in use:** Used to check whether the protocol that the packet is carrying should be allowed.
- **Direction:** Used to check whether the packet is entering or leaving the private network.
- **Interface:** Used to check whether or not the packet is coming from an unreliable zone.



A circuit-level gateway firewall works at the session layer of the OSI model or TCP layer of TCP/IP. It forwards data between networks without verifying it, and blocks incoming packets into the host, but allows the traffic to pass through itself. Information passed to remote computers through a circuit-level gateway will appear to have originated from the gateway, as the incoming traffic carries the IP address of the proxy (circuit-level gateway).

A circuit-level gateway gives controlled access between network services and host requests. For detecting whether or not a requested session is valid, it checks TCP handshaking between packets. Circuit-level gateways do not filter individual packets. They are relatively inexpensive and hide the information about the private network that they protect.



Application-based proxy firewalls concentrate on the Application layer rather than just the packets. The need of use of application level firewall arises as huge amount of voice, video, and collaborative traffic accessed at data-link layer and network layer utilized for unauthorized access to internal and external networks.

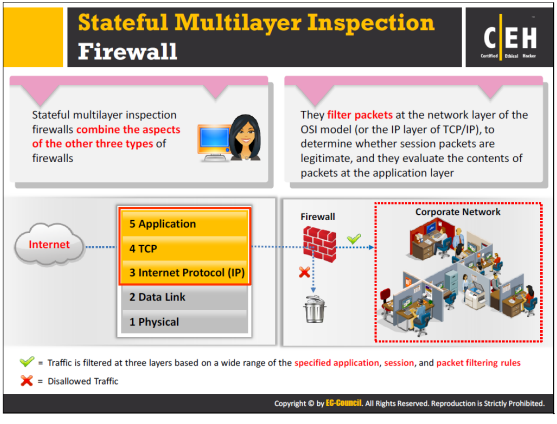
Traditional firewalls are unable to filter such types of traffic. They can inspect, find, and verify malicious traffic that is missed by stateful inspection firewalls to make decisions about whether to allow it access, and improves the overall security of the application layer. For example, worms that send malicious code in legitimate protocols cannot be detected by stateful firewalls, as proxy firewalls concentrate on packet headers at the network layer. However, deep packet inspection firewalls can find such attacks with the help of informative signatures added inside packets.

Some of features of application-level firewalls:

- They analyze the application information to make decisions about whether to permit traffic.
- Being proxy-based, they can permit or deny traffic according to the authenticity of the user or process involved.
- A content-caching proxy optimizes performance by caching frequently accessed information rather than sending new requests to the servers for the same old data.

Application-layer firewalls can function in one of two modes: active or passive.

- **Active application-level firewalls**: They examine all incoming requests, including the actual message that exchanged against known vulnerabilities, such as SQL injection, parameter and cookie tampering, and cross-site scripting. The requests deemed genuine are allowed to pass through them.
- **Passive application-level firewalls**: They work similarly to an IDS, in that they also check all incoming requests against known vulnerabilities, but they do not actively reject or deny those requests if a potential attack is discovered.

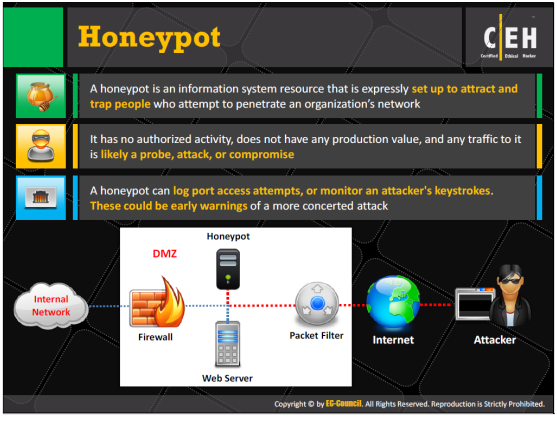


Stateful multilayer inspection firewalls combine the aspects of the other three types of firewalls. They filter packets at the network layer, to determine whether session packets are legitimate, and they evaluate the contents of packets at the application layer.

With the use of stateful packet filtering, you can overcome the limitation of packet firewalls that can only filter on IP address, port, and protocol, and so on. This multilayer firewall can perform deep packet inspection.

Features of the Stateful Multilayer Inspection Firewall:

- This type of firewall can remember the packets that passed through it earlier and make decisions about future packets based on the stated of the conversation.
- These firewalls provide the best of both packet filtering and application-based filtering.
- Cisco PIX firewalls are stateful.
- These firewalls track and log slots or translations.

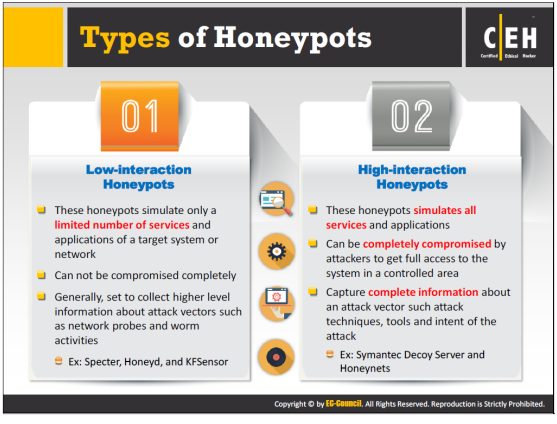


A honeypot is a computer system on the Internet intended to attract and trap people who try unauthorized or illicit utilization of the host system. It is a fake proxy run in an attempt to frame attackers by logging traffic through it, and then sending complaints to victims' ISPs. Whenever there is any interaction with a honeypot, it is most likely to be a malicious activity. Honeypots are unique; they do not solve a specific problem. Instead, they are a highly flexible tool with many different security applications. Some honeypots helps in preventing attacks, others can be used to detect attacks, while others can be used for information gathering and research. It requires require a considerable amount of attention to maintain a honeypot.

To set up a honey pot:

- Install a system on the network with no particular purpose other than to log all attempted access.
- Install an older, unpatched operating system on a network. For example, the default installation of WinNT 4 with IIS 4 can be hacked using several different techniques. A standard intrusion detection system can then be used to log hacks directed against the system and further track what the intruder attempts to do with the system once it is compromised. Install special software designed for this purpose, which will have the advantage of making it appear that the intruder is successful without really allowing him/her access to the network.
- Ensure that the attacker cannot easily delete system data intended to be in the honeypot.

However, any existing system can be "**honeypot-ized**." For example, on WinNT, it is possible to rename the default administrator account and then create a dummy account called "**administrator**" with no password. WinNT allows extensive logging of a person's activities, so this honeypot tracks users who are attempting to gain administrator access and exploit that access.



Honeypots are classified into two types based on their design criteria:

Low-interaction Honeypots

Low-interaction Honeypot emulates the services and programs that would found on an individual's system. If the attacker does something that the emulation does not expect, the honeypot will simply generate an error. They capture limited amounts of information, mainly transactional data and some limited interaction. Some examples are Specter, Honeyd, and KFSensor.

Honeyd is a low-interaction honeypot. It is open source and designed to run primarily on UNIX systems. Honeyd works on the concept of monitoring unused IP space. Anytime it sees a connection attempt to an unused IP, it intercepts the connection and then interacts with the attacker, pretending to be the victim.

By default, Honeyd detects and logs connections to any UDP or TCP port. In addition, the user can configure emulated services to monitor specific ports, such as an emulated FTP server monitoring port 21 (TCP). When an attacker connects to the emulated service, not only does the honeypot detect and log the activity, but also it captures all of the attacker's interaction with the emulated service.

In the case of the emulated FTP server, it potentially captures the attackers' login and password, issued commands, what they were looking for, or tracks their identity. Most

emulated services work the same way. They expect a specific type of behavior, and are programmed to react in a predetermined way.

High-Interaction Honeypots

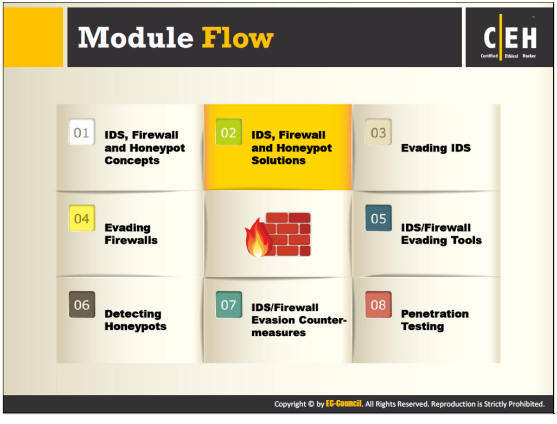
Unlike their low-interaction counterparts, high-interaction honeypots do not emulate anything; they run actual vulnerable services or software on production systems with real operating systems and applications. The honeypot-ized system are more prone to infection, as attack attempts can be carried out on real production systems.

A honeynet is a prime example of a high-interaction honeypot and is neither a product nor a software solution that a user installs. Instead, it is an architecture—an entire network of computers designed to attack.

The idea is to have an architecture that creates a highly controlled network with real computers running real applications, in which all activities are controlled and logged.

"**Bad guys**" find, attack, and break into these systems on their own initiative. When they do, they do not realize they are in a honeynet. All of their activity, from encrypted SSH sessions to email and file uploads, is captured without them knowing it by inserting kernel modules on the victim's systems, capturing all of the attacker's actions.

At the same time, the honeynet controls the attacker's activity. Honeynets do this by using a honeywall gateway, which allows inbound traffic to the victim's systems but controls the outbound traffic using intrusion prevention technologies. This gives the attacker the flexibility to interact with the victim's systems, but prevents the attacker from harming other non-honeynet computers.



The previous section discussed about the functioning, role and placement of IDS, firewalls, and honeypots for securing the networks. There are number of easy to use and feature enriched solutions (hardware, software, or both) available for IDS, firewalls, and honeypots implementation. This section will discuss about some of IDS, firewalls, and honeypots solutions available in the market that simplify their usage.

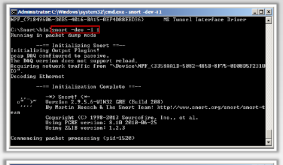
Intrusion Detection Tool: Snort

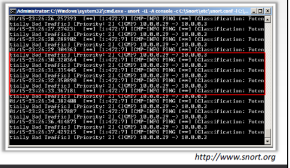
1. Snort is an open source network intrusion detection system, capable of performing real-time traffic analysis and packet logging on IP networks

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, CGI attacks, SMB probes, and OS fingerprinting attempts

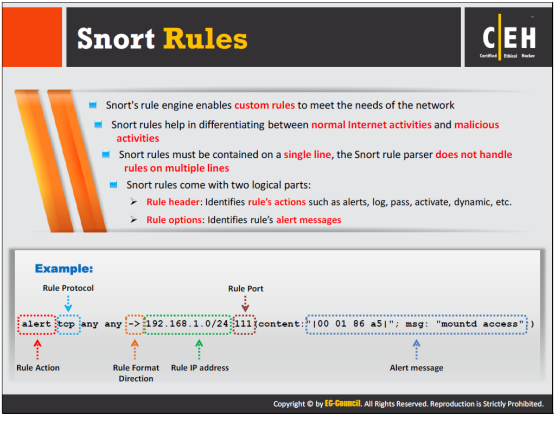
3. It uses a flexible rules language to describe traffic that it should collect or pass, as well as a detection engine that utilizes a modular plug-in architecture

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





http://www.snort.org



Snort uses the popular **libpcap library** (for UNIX/Linux) or **Winpcap** (for Windows), the same library that tcpdump uses to perform its packet sniffing. Attaching snort in promiscuous mode to the network media decodes all the packets passing through the network. It generates alerts according to the content of individual packets and rules defined in the configuration file.

Snort allows users to write their own rules. However, each of these Snort rules must describe the following:

- Any violation of the security policy of the company that might be a threat to the security of the company's network and other valuable information
- All well-known and common attempts to exploit the vulnerabilities in the company's network
- The conditions in which a user thinks that a network packet(s) is unusual (i.e., if the identity of the packet is not authentic)

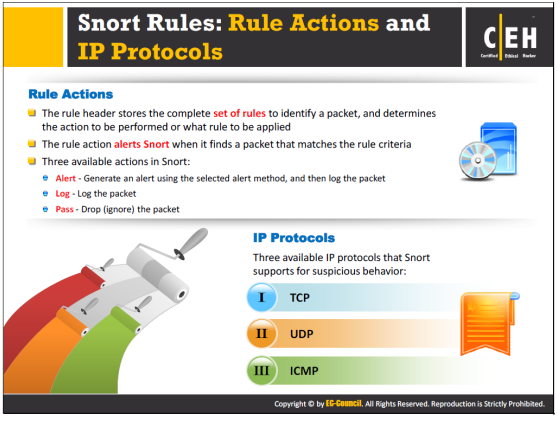
Snort rules, written for both protocol analysis and content searching and matching, should be robust and flexible. The rules should be "**robust**": the system should keep a rigid check on the activities taking place on the network and notify the administrator of any potential intrusion attempt. The rules should be "**flexible**": the system must be compatible enough to act immediately and take necessary remedial measures, according to the nature of the intrusion.

You can achieve both flexibility and robustness using an easy-to-understand and lightweight rule-description language that aids in writing simple Snort rules. Consider two basic principles while writing Snort rules:

- No written rule must extend beyond a single line, so rules should be short, precise, and easy-to-understand.
- Each rule should be divided into two logical sections:
  - The rule header
  - The rule options

The rule header contains the rule's action, the protocol, the source and destination IP addresses, the source and destination port information, and the **CIDR (Classless Inter-Domain Routing) block**.

The rule option section includes alert messages, in addition to information about inspected part of the packet, to determine whether to take rule action.



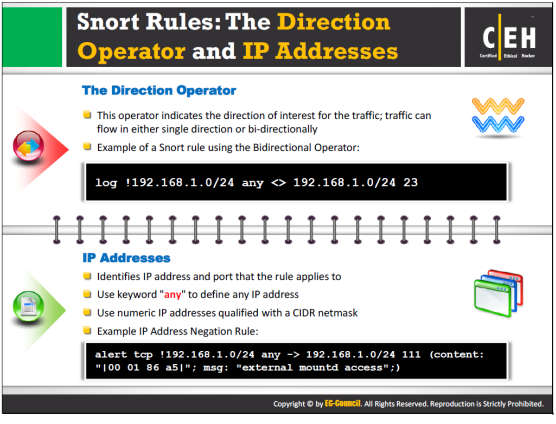
The rule header contains information that defines the who, where, and what of a packet, as well as what to do in the event that a packet with all the attributes indicated in the rule should show up. The first item in a rule is the rule action, which tells Snort "**what to do**" when it finds a packet that matches the rule criteria. There are five available default actions in Snort: alert, log, pass, activate, and dynamic. In addition, if you are running Snort in inline mode, you have additional options, which include drop and reject.

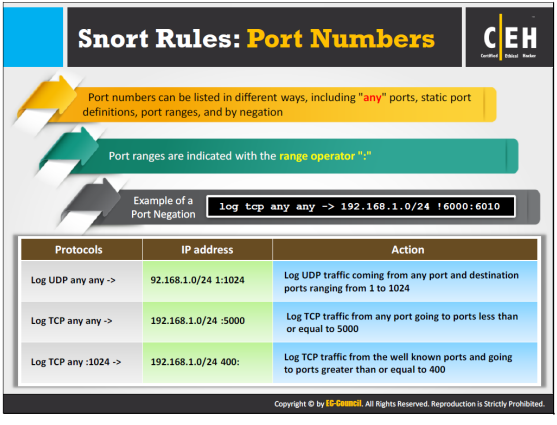
The Internet protocol (IP) sends data from one system to another via the Internet. The IP supports unique addressing for every computer on a network. Organize data on the Internet protocol network into packets. Each packet contains message data, source, destination, and more.

Three available IP protocols that Snort supports for suspicious behavior:

- **TCP:** Transmission control protocol is a part of the Internet Protocol. It is used to connect two different hosts and exchanges data between them.
- **UDP:** User Datagram Protocol, used for broadcasting messages over a network.
- **ICMP:** The Internet Control Message protocol is a part of the Internet protocol. Operating systems use ICMP in a network to send error messages, for example.

Source: *http://manual.snort.org*



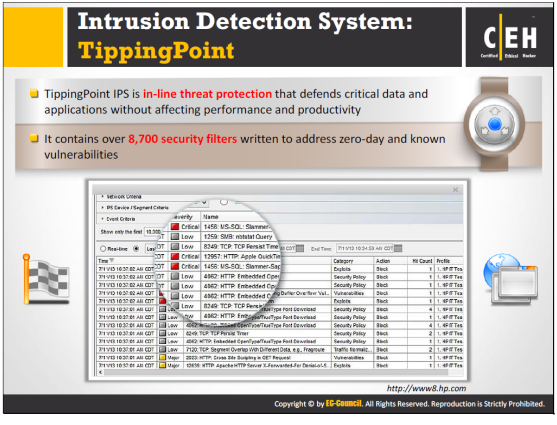


The direction operator "**-$>$**" indicates the orientation, or direction, of the traffic to which the rule applies. Consider an IP address and port number on the left side of the direction operator as the traffic coming from the source host, and the address and port information on the right side of the operator as the destination host. There is also a bidirectional operator, indicated with a "$<>$" operator. This tells Snort to consider the address/port pairs in either the source or the destination orientation, and is handy for recording/analyzing both sides of a conversation, such as telnet or POP3 sessions. Also, note that there is no "$<$-" operator. In Snort versions prior to 1.8.7, the direction operator did not have proper error checking, so many people used an invalid token. The reason the "$<$-" does not exist is so that rules always read consistently.

The next fields in a Snort rule specifies the source and destination IP addresses and ports of the packet, as well as the direction in which the packet is traveling. Snort can accept a single IP address or a list of addresses. When specifying a list of IP address, you should separate each one with a comma and then enclose the list within square brackets, like this:

[192.168.1.1,192.168.1.45,10.1.1.24]

When doing this, be careful not to use any whitespace. You can also specify ranges of IP addresses using CIDR notation, or even include CIDR ranges within lists. Snort also allows you to apply the logical NOT operator ("!") to an IP address or CIDR range to specify that the rule should match all but that address or range of addresses. For example, an easy modification to the initial example is to make it alert on any traffic that originates outside of the local net with the negation operator.

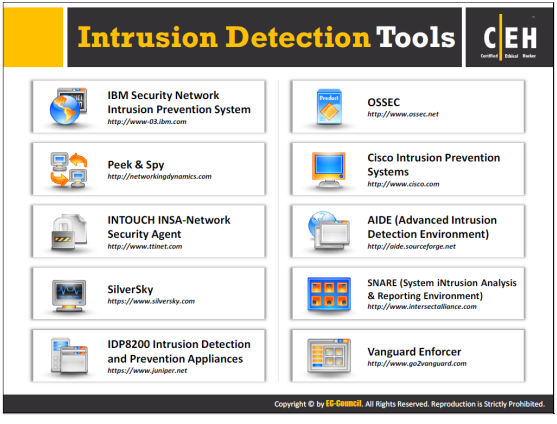


TippingPoint IPS consists of both inbound/outbound traffic inspection, as well as application-level security capabilities.

Features:

- Pre-built, real-time reports that display big-picture analyses on traffic, top applications, and filtered attack events
- Permits to see, control, and leverage the rules, shared services, and profiles of all the firewall devices throughout the network
- Comprises of in-line, bump-in-the-wire intrusion prevention system with layer 2 fallback capabilities
- Gives an overview of current performance for all HP systems in the network, including launch capabilities into targeted management applications by using monitors
- Delivers fully customizable dashboard and management console
- Offers up to 20 GB of protection with less than 40 microseconds of network latency

Source: http://www8.hp.com



Intrusion detection tools detect anomalies. These tools, when run on a dedicated workstation, read all network packets, reconstruct user sessions, and scan for possible intrusions by looking for attack signatures and network traffic statistical anomalies. In addition, these tools give real-time, zero-day protection from network attacks and malicious traffic, and prevent malware, spyware, port scans, viruses, and DoS and DDoS from compromising hosts. Below is a list of widely used IDS systems that you can use to prevent intrusion in your network.

IBM Security Network Intrusion Prevention System

Source: *http://www-03.ibm.com*

IBM® Security Network Intrusion Prevention System appliances stop constantly evolving threats before they affect your business. This means providing both high levels of protection and performance, while lowering the overall cost and complexity associated with deploying and managing a large number of point solutions.

Peek & Spy

Source: *http://networkingdynamics.com*

PEEK & SPY lets a privileged user see exactly what is on another user's terminal and then permits him to either take control of that terminal to fix the problem from his own or let the user have control while he gives any needed instructions. If the PEEK & SPY user chooses to fix it himself, you can display input on the user's screen to show him/her how to fix it. Where PEEK informs users that they may have watched, SPY does not. In addition, SPY gives system

managers documented proof of security breaches and provides a tool to lock out unauthorized users.

INTOUCH INSA–Network Security Agent

Source: *http://www.ttinet.com*

INTOUCH INSA - Network Security Agent scans all user activity on your networks, seven days a week, 24 hours a day. Whether the intrusion is from the outside (firewall failure) or from the inside (unauthorized insider activity). With INTOUCH INSA–Network Security Agent, the Network manager and Network Security Officer have a tool that allows the automated tracking and logging of unauthorized or suspicious activity.

SilverSky

Source: *https://www.silversky.com*

Intrusion Detection and Prevention (IDS/IPS) systems analyze complex network traffic in real-time and proactively block malicious internal traffic and sophisticated attacks that might not be prevented with firewalls alone. SilverSky reduces the costs and complexity of managing IDS/IPS Systems while improving your ability to respond to evolving threats.

IDP8200 Intrusion Detection and Prevention Appliances

Source: *https://www.juniper.net*

The IDP8200 Intrusion Detection and Prevention Appliances is the ideal network intrusion detection and application security management solution for large enterprise networks and service providers that require the highest throughput levels and reliability, and outstanding quality of service.

OSSEC

Source: *http://www.ossec.net*

OSSEC is an Open Source Host-based Intrusion Detection System that performs log analysis, file integrity checking, policy monitoring, rootkit detection, real-time alerting, and active response. It covers PCI DSS sections 11.5 and 10.5.5.

Cisco Intrusion Prevention Systems

Source: *http://www.cisco.com*

Cisco IOS Intrusion Prevention System (IPS) is an inline, deep-packet inspection-based solution that enables Cisco IOS Software to effectively mitigate a wide range of network attacks. Although it is common practice to defend against attacks by inspecting traffic at data centers and corporate headquarters, distributing the network level defense to stop malicious traffic close to its entry point at branch or telecommuter offices is also critical.

AIDE (Advanced Intrusion Detection Environment)

Source: *http://aide.sourceforge.net*

AIDE (Advanced Intrusion Detection Environment) is a file and directory integrity checker. It creates a database from the regular expression rules that it finds from the config file(s). Initialization of this database helps in verifying the integrity of the files. It has several message

digest algorithms to check the integrity of the file. You can also check the inconsistencies of all usual file attributes.

SNARE (System iNtrusion Analysis & Reporting Environment)

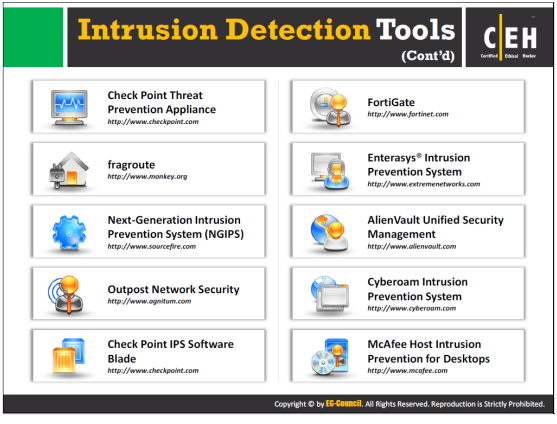
Source: *http://www.intersectalliance.com*

SNARE (System iNtrusion Analysis & Reporting Environment) consists of the centrally installed Snare Server and individual device-based Snare Agents. The Snare Server's role is to give your system administrator all the tools needed to define, gather, index, track, report on and store all relevant IT network security events input from Snare and open source agents. Snare Agents examine all IT events at their source. SSNARE (System iNtrusion Analysis and Reporting Environment) is a series of log collection, forwarding, filtering agents that facilitate centralized analysis of audit log data.

Vanguard Enforcer

Source: *http://www.go2vanguard.com*

Vanguard Enforcer provides real-time intrusion protection, detection and management solutions for the z/OS mainframe that prevent human error and deliberate attacks. By providing 24/7 protections for critical information and resources hosted on the mainframes, Vanguard Enforcer guarantees that z/OS and RACF® security standards, profiles, rules and settings should not compromised. In less than two seconds, the software can automatically detect and notify personnel when threat events on the mainframe and network occur, and then respond to deviations from the security baseline with corrective actions that reassert the approved security policy.



In addition to the previously described intrusion detection tools, a few more tools used for detecting intrusions are:

Check Point Threat Prevention Appliance

Source: *http://www.checkpoint.com*

Check Point Threat Prevention Appliance prevents advanced threats and malware attacks and enables an organization to control access to millions of web sites easily and confidently. Protections include stopping application-specific attacks, botnets, targeted attacks, APTs, and zero-day threats.

fragroute

Source: *http://www.monkey.org*

fragroute intercepts, modifies, and rewrites egress traffic destined for a specified host. It features a simple ruleset language to delay, duplicate, drop, fragment, overlap, print, reorder, segment, and source-route. It helps in testing intrusion detection systems, firewalls, and basic TCP/IP stack behavior.

Next-Generation Intrusion Prevention System (NGIPS)

Source: *http://www.sourcefire.com*

The Sourcefire Next-Generation IPS provides threat protection integrating real-time contextual awareness, full-stack visibility, and intelligent security automation to deliver security

effectiveness, performance, and low total cost of ownership. Passive intrusion detection (IDS) mode notifies of suspicious network traffic and behavior while inline IPS mode blocks threats. The NGIPS solution further expanded to add Application Control/URL Filtering and Advanced Malware Protection.

Outpost Network Security

Source: *http://www.agnitum.com*

Outpost Network Security helps small and medium business (SMB) organizations to protect against modern security challenges and address the problem of productivity waste. It safeguards local networks against external attacks and internal sabotage, keeps endpoints clean of malware, prevents disclosure of inside information, polices employee Internet access, and shifts the task of deploying and managing protection away from the busy workforce.

Check Point IPS Software Blade

Source: *http://www.checkpoint.com*

The IPS Software Blade delivers complete and proactive intrusion prevention—all with the deployment and management advantages of next-generation firewall solution.

FortiGate

Source: *http://www.fortinet.com*

Fortinet Intrusion Prevention System (IPS) technology protects networks from both known and unknown threats, blocking attacks that might otherwise take advantage of network vulnerabilities and unpatched systems. Fortinet understands that your enterprise or service provider network is supporting many different applications, protocols and operating systems at the same time. This diverse infrastructure can complicate maintenance and patching of servers and network devices, resulting in delays and systems that are vulnerable to evolving threats.

Enterasys® Intrusion Prevention System

Source: *http://www.extremenetworks.com*

IPS provides exceptional functionality by locating, containing, and removing the source of the attack from the network. It ensures the confidentiality, integrity, and availability of business-critical resources with Intrusion Prevention capabilities.

AlienVault Unified Security Management

Source: *http://www.alienvault.com*

AlienVault Unified Security Management™ provides you complete security visibility by delivering three types of intrusion detection system (IDS) software that includes network intrusion detection (NIDS), host-based intrusion detection (HIDS), and wireless intrusion detection (WIDS), combined with all of the essential security capabilities built-in and continuous threat intelligence updates from AlienVault Labs.

Cyberoam Intrusion Prevention System

Source: *http://www.cyberoam.com*

Cyberoam Intrusion Prevention System protects against network and application-level attacks, securing organizations against intrusion attempts, malware, Trojans, DoS, and DDoS attacks, malicious code transmission, backdoor activity and blended threats.

McAfee Host Intrusion Prevention for Desktops

Source: *http://www.mcafee.com*

McAfee Host Intrusion Prevention for Desktops safeguards your business against complex security threats that may unintentionally introduced or allowed by desktops and laptops. Host Intrusion Prevention for Desktops is easy to deploy, configure, and manage.

There are also Intrusion detection tools available for mobile devices that can help you detect and prevent any attempt of intrusion.

Wi-Fi Intrusion Detection

Source: *https://play.google.com*

Wi-Fi Intrusion Detection tool allows one to find intruders on a Wi-Fi network.

Wi-Fi Intruder Detector Pro

Source: *https://play.google.com*

Wi-Fi Intruder Detector Pro helps to find security leaks in the Wi-Fi network internet connection. It allows one to find any intruder who is accessing the network, Wi-Fi, or Internet connection without your consent.

WiFi Inspector

Source: *https://play.google.com*

Wi-Fi Inspector allows you to find all the devices connected to the network (both wired and Wi-Fi, whether consoles, TVs, pcs, tablets, phones, etc.), giving relevant data such as IP address, manufacturer, device name and Mac Address. It also allows saving a list of known devices with custom name and finds intruders in a short period.

ZoneAlarm PRO Firewall blocks attackers and intruders from accessing your system. It prevents identity theft by guarding your personal data. It even erases your tracks allowing you to surf the web in complete privacy. Furthermore, it locks out attackers, blocks intrusions, and makes your PC invisible online. In addition, it filters out annoying and potentially dangerous email.

Features:

- Two-way firewall that monitors and blocks inbound as well as outbound traffic
- Allows users to browse the web privately
- Identity protection services helps to prevent identity theft by guarding crucial data of the users. It also offers PC protection and data encryption
- Through Do Not Track, it stops data collecting companies from tracking the online users
- Online Backup to backs up files and restores the data in the event of loss, theft, accidental deletion or disk failure
- Privacy and Security Toolbar provides Site Check, Do Not Track, Facebook Privacy Scan, private browsing, etc.

Source: *http://www.zonealarm.com*

Comodo Firewall Software alerts when any application tries to connect Internet. It will also list all the active connections including details such as protocol, source, and destination.

This tool offers various levels of firewall security such as:

- **Firewall only**: For enterprise strength network firewall.
- **Firewall with Optimum Proactive Defense**: Provides optimal network security by adding protection against the methods commonly used by malware in order to bypass the firewall.
- **Firewall with Maximum Proactive Defense**: Provides maximum security including leak protection and defense against malware threats.

Source: *http://personalfirewall.comodo.com*

Firewalls provide essential protection to the computers against viruses, privacy threats, objectionable content, hackers, and malicious software when connected to the Internet. A firewall monitors running applications that access the network. It analyzes downloads and warns you if downloading a malicious file, stops it from infecting your PC. A few of the firewalls that provide system protection are:

Cisco ASA 1000V Cloud Firewall

Source: *http://www.cisco.com*

Cisco ASA 1000V Cloud Firewall provides consistent, enterprise-class security for private and public clouds. The ASA 1000V employs mainstream, Adaptive Security Appliance (ASA) technology, optimized for highly secure multi-tenant virtual and cloud infrastructure at the edge. This helps to enable consistency across physical, virtual, and cloud infrastructures.

Check Point Firewall Software Blade

Source: *http://www.checkpoint.com*

The Check Point Firewall Software Blade incorporates all of the power and capability of the revolutionary FireWall-1 solution for strongest level of gateway security while adding user identity awareness to provide granular event awareness and policy enforcement.

eScan Enterprise Edition

Source: *http://www.escanav.com*

The new eScan Enterprise Edition (with Hybrid Network Support) strengthens the cyber-security capabilities of businesses by providing multi-layered protection against complex threats and securing critical business information effectively without constraining business growth.

Jetico Personal Firewall

Source: *http://www.jetico.com*

Jetico Personal Firewall software protects computers against hackers and malicious software when connected to the Internet. It offers detailed and configurable event logs, detailed and configurable reports, and the option of viewing and editing firewall configuration.

Outpost Security Suite

Source: *http://free.agnitum.com*

Outpost Security Suite Pro delivers comprehensive, multilayered protection against all types of Internet threat with minimal impact on PC performance. It ensures that confronted threats at every stage of potential propagation routes—from initial contact to potential data theft or system compromise.

Novell BorderManager

Source: *http://www.novell.com*

Novell® BorderManager® is a premier firewall and VPN technologies product that enables secure identity management solutions. With its directory-integrated features, you can control, accelerate, and monitor your users' Internet activities. Novell BorderManager leverages identity-based access control and forward proxies that help you safeguard your network against undesirable Internet content while maintaining exceptional performance levels. Novell BorderManager also integrates IPSec-based VPN services, firewall to protect your network, and ensures that your users are productive. The Novell BorderManager VPN client software allows a workstation to communicate securely over the Internet to a network protected by a Novell VPN server.

Untangle NG Firewall

Source: *https://www.untangle.com*

NG Firewall is a next-generation platform for deploying network-based applications. It unites these applications around a common GUI, database and reporting. NG Firewall's applications inspect network traffic simultaneously, greatly reducing the resource requirements of each individual application.

Sonicwall

Source: *http://www.sonicwall.com*

The Dell SonicWALL family of firewalls tightly integrates intrusion prevention, malware protection, and application intelligence and control with real-time visualization. The Dell

SonicWALL Reassembly-Free Deep Packet Inspection engine scans 100% of traffic and massively scales to meet the needs of the most high-performance networks.

Online Armor

Source: *http://www.online-armor.com*

Emsisoft Online Armor Premium not only monitors network traffic, but also monitors critical internal system actions. It protects the user by reporting any suspicious programs like malware, Trojans, phishing attacks, and spyware that tries to record keyboard sequences. Online Armor lists all autoruns and classifies them by their security status (trusted, not trusted, and unknown).

FortiGate-5101C

Source: *http://www.fortinet.com*

The FortiGate-5101C security system is a high-performance Advanced Telecommunications Computing Architecture (ATCA) compliant. You can install FortiGate security system in any ATCA chassis that can provide sufficient power and cooling.

The firewalls discussed previously used for securing personal computers and networks. Likewise, there are firewalls that can secure your mobile device.

Android Firewall

Source: *https://play.google.com*

This tool uses iptables to limit data usage and add security to Android.

Features:

- Hollow UI for all users and navigation drawer
- Ability to export and import rules to files
- Supports Tasker/Locale, roaming, LAN, VPN, SD card, tethering, etc.
- Multi-user support for tablets
- Ability to log both rejected and accepted packets
- Application shortcuts for quickly enabling/disabling the firewall and swapping profiles

Firewall iP

Source: *http://cydia.saurik.com*

Firewall iP is a security tool that allows blocking of outgoing connections (TCP and UDP) selectively.

Features:

- Hooks into applications and warns if the app wants to establish a connection to a host
- Blocks specific ports, analytic providers, and data collectors
- Options for allowing/denying the connection once/always or for allowing/denying all connections for the application
- Blocks unneeded content such as ads
- Shows the host name for the connection
- Provides the Whois information
- Block connections of app when the user is on the cellular network

Besides the firewalls discussed above, there many others that can help you secure your mobile devices from internet threats, including those described below.

Mobiwol: NoRoot Firewall

Source: *http://www.mobiwol.com*

Mobiwol is an Android Firewall that monitors and controls the data connections initiated by the apps. With Mobiwol, you can prevent data leakage, manage data limits and increase battery uptime. Mobiwol No Root Firewall allows the apps you trust and limit or block those you do not.

DroidWall

Source: *https://code.google.com*

This tool is a front-end application for the iptables generated based on Linux firewall. It allows the user to restrict which applications permitted to access data networks such as 2G/3G and/or Wi-Fi.

AFWall+

Source: *https://github.com*

Android Firewall+ is a GUI-based advanced iptables editor for Android and provides control over which Android apps allowed to access the network.

Firewall Plus

Source: *http://squariolabs.com*

Firewall Plus allows the user to restrict which apps can access the network both on 3G/4G networks and on Wi-Fi. Advanced users will also be able to define custom iptable rules.

Root Firewall

Source: *http://www.rootuninstaller.com*

Root Firewall is an Android firewall that can block ads; prevent data over-billing, save battery life and protects privacy. This tool requires a rooted Android device.

Android Firewall Gold

Source: *https://play.google.com*

Android Firewall Gold is used to secure devices against intrusion, malware, and backdoors. It closes unused ports to help extend battery life. Allows advanced users to define custom iptables rule manually. To use this app, the Android device must be rooted first. It allows the user to restrict which apps can access the network.

Droid Firewall

Source: *https://play.google.com*

Droid Firewall is a tool that prevents other apps from sharing your personal data and information like contacts and messages, saves mobile data, and blocks suspicious outgoing activities such as apps that collect personal data.

Privacy Shield

Source: *http://www.snoopwall.com*

Snoopwall is a tool that picks up where antivirus technology fails, strengths existing firewalls, complements in-built device security measures and monitors apps and device ports for malicious activities. Privacy Shield is a product of snoopwall used to defend online banking and personal identity from the preying eyes of eavesdropping.

aFirewall

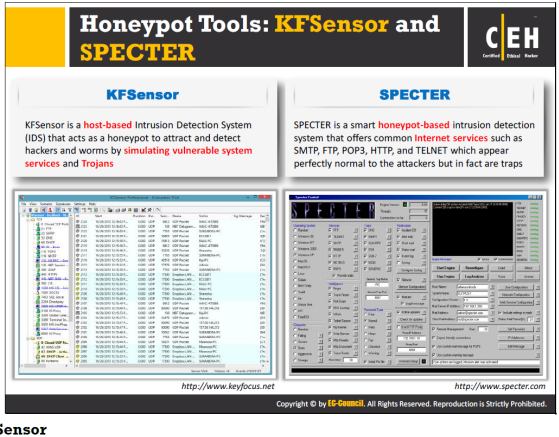
Source: *http://afirewall.wordpress.com*

aFirewall allows configuring multi rules to block different call or message at different time. It can backup blocked log and protected log to Gmail based on BackupToEmail application.

NoRoot Firewall

Source: *https://play.google.com*

NoRoot Firewall acts as a firewall for Android devices without root. It protects your personal information from sending to the Internet and notifies when an app is trying to access the Internet. It allows the user to create filter rules based on IP addresses, host names or domain names. One can allow or deny only specific connection of an app. This tool uses minimal permissions, no location, and no phone number.



KFSensor

Source: *http://www.keyfocus.net*

KFSensor is a **host-based honeypot Intrusion Detection System (IDS)**. By acting as a decoy server, it can divert attacks from critical systems and provide a higher level of information than can be achieved by using firewalls and NIDS alone.

You can use KFSensor in a Windows-based corporate environment and contains many innovative and unique features such as remote management, a Snort-compatible signature engine, and emulations of Windows networking protocols.

Features:

- Signature attack identification
- Detects Windows networking attacks
- Remote Administration
- Detects unknown threats
- Security in-depth
- Real time detection
- Advanced server simulation
- Extendable architecture
- No false positives and low overhead

SPECTER

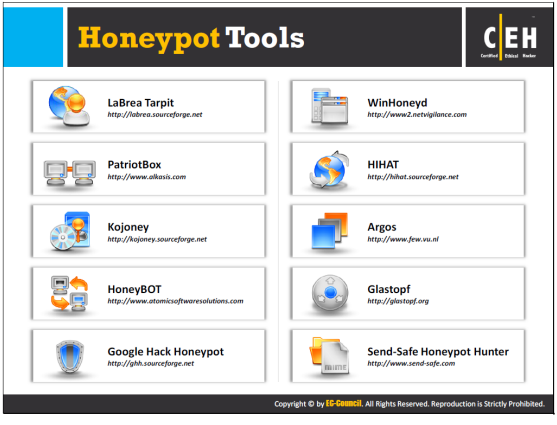
Source: *http://www.specter.com*

SPECTER is a honeypot or deception system. It simulates a complete system and provides an interesting target to lure hackers away from production systems. It offers common Internet services such as SMTP, FTP, POP3, HTTP, and TELNET, which appear perfectly normal to attackers. However, it is a trap for an attacker by messing them so that he leaves some traces knowing that they had connected to a decoy system that does none of the things it appears to do; but instead, it logs everything and notifies the appropriate people.

Furthermore, SPECTER automatically investigates attackers while they are still trying to break in. It provides massive amounts of decoy content and it generates decoy programs that cannot leave hidden marks on the attacker's computer. Automated weekly online updates of the honeypot's content and vulnerability databases allow the honeypot to change constantly without user interaction.

Advantages:

- Suspicious interest in the network, and computers, can be detected immediately
- Administrators are notified of hostile activity when it happens, so that they can immediately look at the problem and take action
- The system is very easy to set up and configure while providing sophisticated features. Fully automated online updates of the honeypot's content and vulnerability databases allow the honeypot to change constantly without user interaction
- There cannot be false alerts, as a legitimate user cannot connect to the honeypot
- Specter runs on 14 different operating systems



Honeypots are the security tools that give the security community an opportunity to monitor attackers' tricks and exploits by logging their every activity, so that they can respond to these exploits quickly without attacker's actually misusing and compromising systems. Below is a list of honeypot tools:

LaBrea Tarpit

Source: *http://labrea.sourceforge.net*

LaBrea takes over unused IP addresses, and creates virtual servers that are attractive to worms, hackers, and other denizens of the Internet. The program answers connection attempts in such a way that the machine at the other end gets "stuck," sometimes for a very long time.

PatriotBox

Source: *http://www.alkasis.com*

PatriotBox HoneyPot Server is an early detection solution for cost-effective threat prioritization. By alerting network management of intrusion attempts, it offers comprehensive attack detection, manageable deployment, elimination of false positives, invisible monitoring, and reporting, trending and policy-based attack migration. It helps to reduce spam on the Internet and simulates an Open Relay Mail server.

Kojoney

Source: *http://kojoney.sourceforge.net*

Kojoney is a low-level interaction honeypot that emulates an SSH server. The developer has written the daemon in Python using the Twisted Conch libraries.

HoneyBOT

Source: *http://www.atomicsoftwaresolutions.com*

HoneyBOT is a medium interaction honeypot for windows. A honeypot creates a safe environment to capture and interact with unsolicited and often malicious traffic on a network. It is a solution ideal for network security research or as part of an early warning IDS. The logging capability of a honeypot is far greater than any other network security tool and captures raw packet level data even including the keystrokes and mistakes made by hackers.

Google Hack Honeypot

Source: *http://ghh.sourceforge.net*

Google Hack Honeypot is the reaction to a new type of malicious web traffic: search engine hackers. It designed to provide reconnaissance against attackers that use search engines as a hacking tool against your resources. GHH implements honeypot theory to provide additional security to your web presence.

WinHoneyd

Source: *http://www2.netvigilance.com*

WinHoneyd is an open-source low-interaction honeypot developed by Niels Provos, and based on Honeyd for Unix/Linux platform. It is a Windows command-line program. It is able to simulate big network structures with different operating systems on a single host. It can also simulate hosts running different services.

HIHAT

Source: *http://hihat.sourceforge.net*

The High Interaction Honeypot Analysis Toolkit (HIHAT) allows transforming arbitrary PHP applications into web-based high-interaction Honeypots. Furthermore, a graphical user interface supports the process of monitoring the Honeypot and analyzing the acquired data. It automatically scans for known attacks and detects SLQ injections, (remote) file inclusions, cross-site scripting (XSS), and download attempts for malicious files (e.g., with WGET or CURL, Command-Injections).

Argos

Source: *http://www.few.vu.nl*

Argos is a full and secure system emulator designed for use in honeypots. It depends on Qemu, an open source emulator that uses dynamic translation to achieve a good emulation speed. Argos extends Qemu to enable it to detect remote attempts to compromise the emulated guest operating system. Using dynamic taint analysis, it tracks network data throughout execution

and detects any attempts to use them in an illegal way. When it detects the attack, it logs the memory footprint of the attack.

Glastopf

Source: *http://glastopf.org*

Glastopf is a Honeypot, which emulates thousands of vulnerabilities to gather data from attacks targeting web applications. The principle behind it is very simple: provide the correct response to the attacker exploiting the web application.

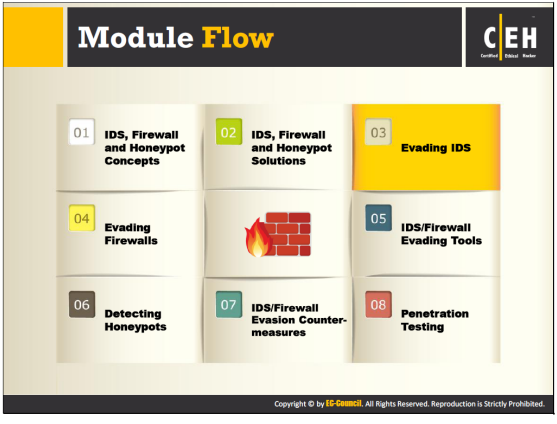
Send-Safe Honeypot Hunter

Source: *http://www.send-safe.com*

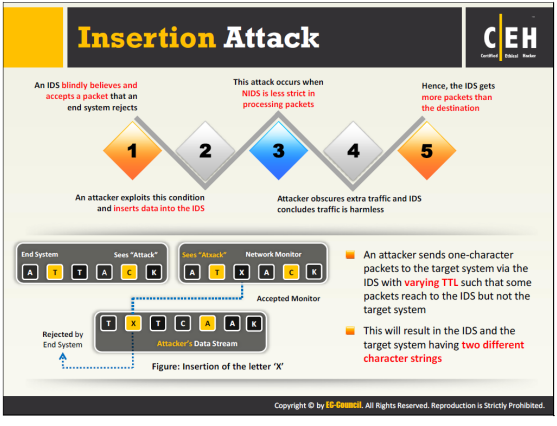
Send-Safe Honeypot Hunter is a tool designed for checking lists of HTTPS and SOCKS proxies for honeypots.

Network administrators can deploy honeypots on all kinds of mobile devices, (e.g., smartphones, tablets) to provide a quick assessment on the potential security state of a network. To unlock the full functionality of HosTaGe, users need to have a rooted Android device with Portbinder installed. Portbinder allows binding of privileged ports (i.e., < 1024) to allow some services to be emulated.

Source: *http://www.tk.informatik.tu-darmstadt.de*



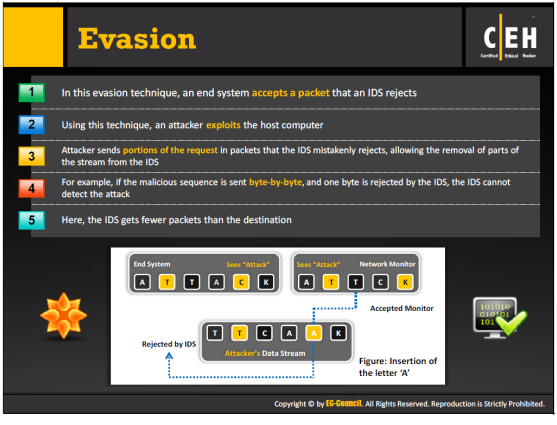
Previous sections helped to understand about Intrusion detection Systems (IDS), their roles and functions, how they protect your network from intruders, and the number of IDS solutions available. Even though IDS secures any attempts of breaking the network security, the attackers can still try to evade the IDS. This section explains about various ways attackers use to evade the IDS.



Insertion is the process in which the attacker confuses the IDS by forcing it to read invalid packets (i.e., the system may not accept the packet addressed to it). If a packet is malformed or if it does not reach its actual destination, the packet is invalid. If the IDS reads an invalid packet, it becomes confused.

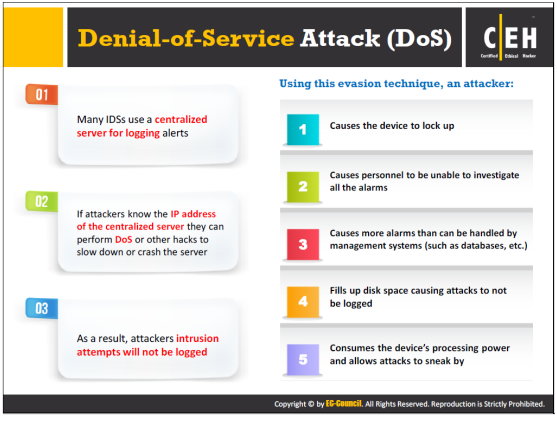
To understand how insertion becomes a problem for a network IDS, it is important to understand how the IDS detects attacks. It employs pattern-matching algorithms to look for specific patterns of data in a packet or stream of packets. For example, it might look for the "phf" string in an HTTP request to discover a PHF **Common Gateway Interface (CGI)** attack. An attacker who can insert packets into the IDS can prevent pattern matching from working. For instance, an attacker can send the string "phf" to a web server, attempting to exploit the CGI vulnerability, but force the IDS to read "phoneyf" (by "inserting" the string "oney") instead. One simple insertion attack involves intentionally corrupting the IP checksum. Every packet transmitted on an IP network has a checksum that verifies the corrupted packets. IP checksums are 16-bit numbers, computed by examining information in the packet. If the checksum on an IP packet does not match the actual packet, the addressed host will not accept it, while the IDS might consider it as part of the effective stream.

For example, the attacker can send packets whose time-to-live (TTL) fields are crafted to reach the IDS but not the target computers. An attacker confronts the IDS with a stream of one-character packets (the attacker-originated data stream), in which one of the characters (the letter "X") will be accepted only by the IDS. As a result, the IDS and the end system reconstruct two different strings.



An "evasion" attack occurs when the IDS discards packets while the host that has to get the packets accepts them. Evasion attacks devastates to the accuracy of the IDS. An evasion attack at the IP layer allows an attacker to attempt arbitrary attacks against hosts on a network, 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 ID system's view. For example, if the attacker sends malicious sequence byte by byte, and if the IDS rejects only one byte, it cannot detect the attack. Here, the IDS gets fewer packets than the destination.

One example of an evasion attack occurs when an attacker opens a TCP connection with a data packet. Before any TCP connection can be used, it must be "**opened**" with a handshake between the two endpoints of the connection. An important fact about TCP is that the handshake packets can themselves bear data. The IDS that does not accept the data in these packets is vulnerable to an evasion attack.



Multiple types of DoS attack will work against IDS systems. The attacker identifies a point of network processing that requires the allocation of a resource, causing a condition to occur that consumes all of that resource. The resources affected by the attacker are CPU cycles, memory, disk space, and network bandwidth. Attackers monitor and attack the CPU capabilities of the IDS. This is because IDS needs half of the CPU cycle to read the packets, detecting what the purpose of their existence is, and then comparing them with some location in the saved network state. An attacker can verify the most computationally expensive network processing operations and then compel the IDS to spend all its time carrying out useless work.

An IDS requires memory for a variety of things. For generating a match for the patterns, save the TCP connections, maintain reassembly queues, and generate the buffers of the data. In the initial phase, the system requires memory so that it can read the packets. System will allocate the memory for network processing operations. An attacker can verify the processing operations that require the IDS to allocate memory and force the IDS to allocate all of its memory for meaningless information.

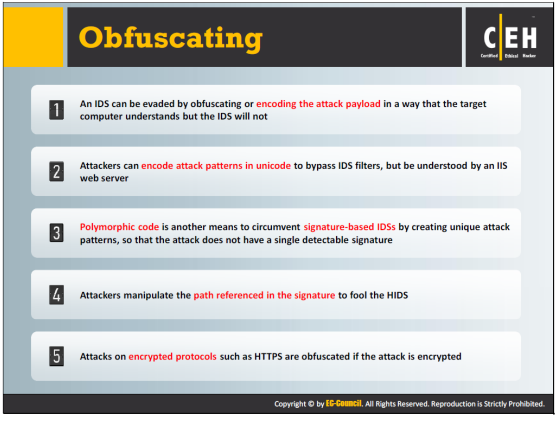
In certain circumstances, the IDS store activity logs on the disk. The stored events occupy most of the disk space. Most computers have limited disk space. The attackers can occupy a major part of the disk space on the IDS by creating and storing a large number of useless events. This renders the IDS useless in terms of storing real events.

Network IDS systems record the activity on the networks they monitor. They are competent because networks are hardly ever used to their full capacity; few monitoring systems can cope with an extremely busy network.

The IDS system, unlike an end system, must read everyone's packets, not just those sent specifically to it. An attacker can overload the network with meaningless information and prevent the IDS system from keeping up with what is actually happening on the network.

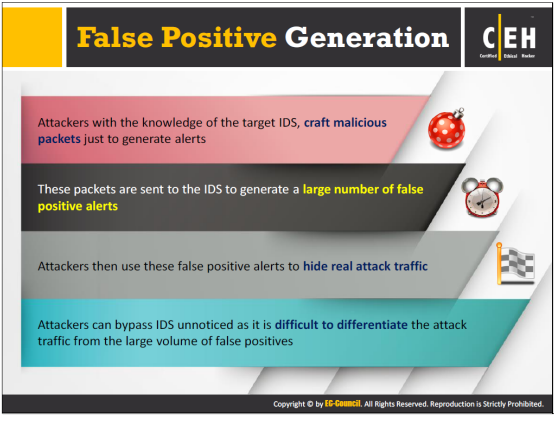
Many IDSes today employ central logging servers that are used exclusively to store IDS alert logs. The central server's function is to centralize alert data so that it viewed as a whole rather than on a system-by-system basis.

However, if attackers know the central log server's IP address, they could slow it down or even crash it using a DoS attack. After shutting down the server, attacks could go unnoticed because the alert data is now no longer logged.

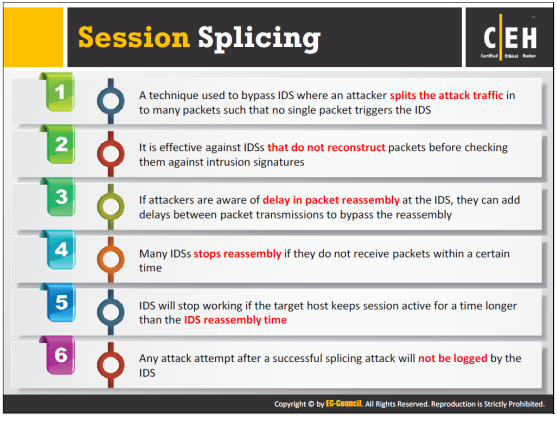


Obfuscation means to make code harder to understand or read, generally for privacy or security purposes. A tool called an obfuscator converts a straightforward program into one that works the same way but is much harder to understand.

A threat can evade an IDS by obfuscating or encoding the attack payload in a way that the target computer will reverse but the IDS will not. An attacker manipulates the path referenced in the signature to fool the HIDS. Using the Unicode character, an attacker could encode attack packets that the IDS would not recognize, but an IIS web server would decode. Polymorphic code is another means to circumvent signature-based IDSes by creating unique attack patterns, so that the attack does not have a single detectable signature. Attackers perform obfuscated attacks on encrypted protocols such as HTTPS.

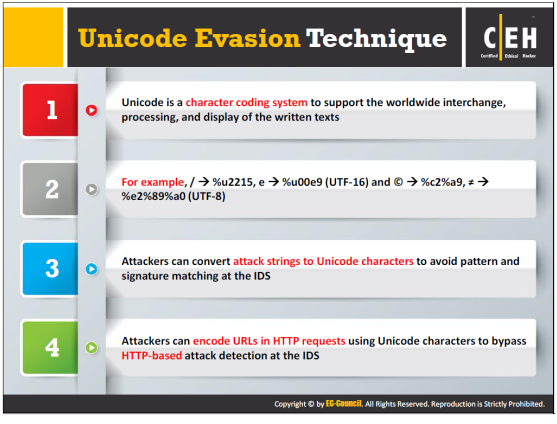


This mode does not attack the target; instead, it does something relatively normal. In this mode, the IDS generates an alarm when no condition is present to warrant one. Another attack similar to the DoS method is to generate a large amount of alert data that the IDS will log. Attackers construct packets known to trigger alerts within the IDS, forcing it to generate a large number of false reports. This type of attack creates a great deal of log "noise" in an attempt to blend real attacks with the false. Attackers know all too well that when looking at log data, it can be very difficult to differentiate between legitimate attacks and false positives. If attackers have knowledge of the IDS system, they can even generate false positives specific to that IDS.



Session splicing is an IDS evasion technique that exploits how some IDSs do not reconstruct sessions before pattern-matching the data. It is a network-level evasion method that divides the string across several packets. The attacker divides the data in the packets into small portions of bytes and while delivering the data evades the string match. Attackers uses this technique to deliver the data into several small sized packets. The IDS cannot handle too many small sized packets and fails to detect the attack signatures. If attackers know what IDS system is in use, they could add delays between packets to bypass reassembly checking.

Many IDS reassemble communication streams, so if a packet not received within a reasonable period, many IDSs stop reassembling and handling that stream. If the application under attack keeps a session active longer than an IDS will spend on reassembling it, the IDS will stop. As a result, any session after the IDS stops reassembling the sessions will be susceptible to malicious data theft by attackers. Attackers can use different tools such as **Nessus** and **Whisker** for session-splicing attacks.



Unicode is a character coding system that supports encoding, processing, and displaying of written texts for worldwide languages to maintain consistency in computer representation. Several standards such as Java, LDAP, and XML require Unicode, and many operating systems and applications support it. Attackers can implement an attack by different character encodings known as "**code points**" in the Unicode code space,. The most commonly used character encodings are Unicode Transformation Format (UTF)-8 and UTF-16.

For Example: In UTF-16, character "/" can be represented as "%u2215," "e" as "%u00e9," and in UTF- 8, "©"as "%c2%a9" and "≠" as "%e2%89%a0."

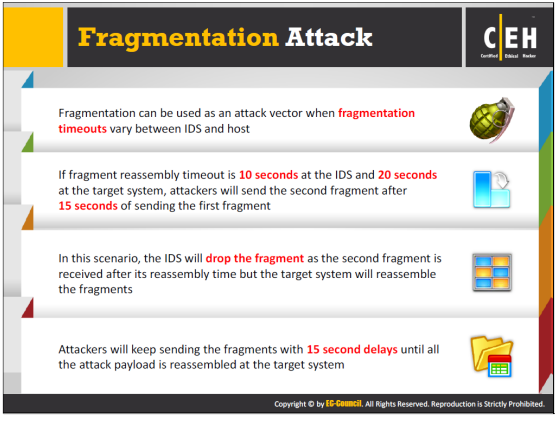
Problems with Unicode:

In the Unicode code space, all the code points treated differently but it is possible that there could be multiple representations of a single character. There are also code points that alter the previous code points. Moreover, applications or operating systems may assign the same representation to different code points. Because of this complexity, some IDS systems handle Unicode improperly as Unicode allows multiple interpretations of the same characters.

For example, "\" represents 5C, C19C, and E0819C, which makes writing pattern matching signatures very difficult. Taking this as an advantage, attackers can convert attack strings to Unicode characters to avoid pattern and signature matching in the IDS. Attackers can also encode URLs in HTTP requests using Unicode characters to bypass HTTP-based attack detection at the IDS.

Example for how Unicode affects IDS:

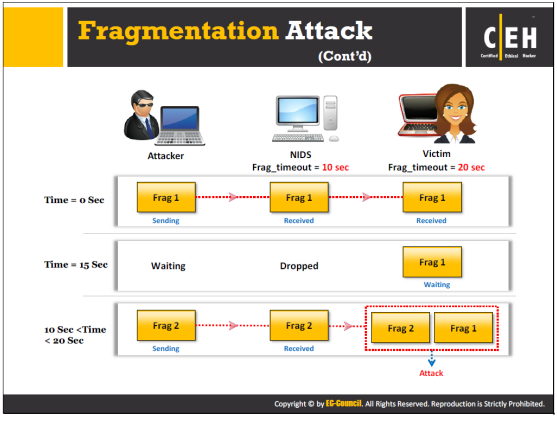
- Microsoft IIS 4.0/5.0 Directory Traversal vulnerability released in October 2000 by Rain Forrest Puppy
- IIS verifies directory traversal before it is decoded to UTF-8 and this vulnerability of IIS improperly checks directory listings that were Unicode encoded within the URL request
- For instance, if the attacker tries to escape out of web root and submits a URL like http://iis/../../page/system32/cmd.exe, IIS removes the extra "../.." and generates an error
- Instead, the directory traversal would not remove, if the attacker encodes "**../..**" portion of the attack with UTF-8 so that "..%C4%6A.." is sent. The operating system decodes the UTF-8 and proceeds further
- This allowed remote attackers to view files on the IIS server that they normally would not be permitted to see



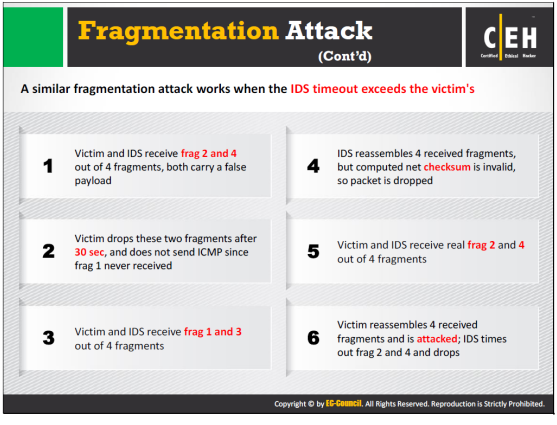
IP packets must follow standard **Maximum Transmission Unit (MTU)** size while traveling across the network. If the packet size is exceeded, it is split into multiple fragments ("fragmentation"). The IP header contains a fragment ID, fragment offset, fragment length, fragments flags, and others besides the original data. In a network, the flow of packets is irregular, so systems need to keep fragments around, wait for future fragments, and then reassemble them in order. Through the process of fragmenting and reassembling, attackers can send malicious packets over the network to exploit and attack the systems. To avoid detection by an IDS, attackers may utilize fragmentation using the fragment reassembly timeout, which varies from system to system.

Attack Scenario-1:

If, for example, the 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 on receiving the second fragment after its reassembly time out, but the target host will reassemble the fragments. Attackers will continue sending fragments with intervals of 15 seconds until the attack payload reassembles at the target system. Thus, the victim will reassemble the fragments and receive the attack code, whereas the IDS will not make any noise or generate alerts as the IDS drops the fragments of packets.



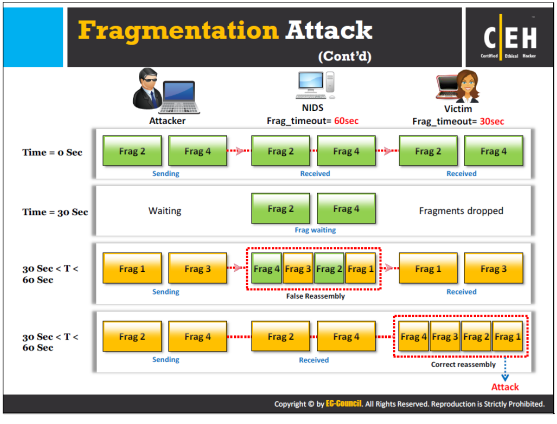
The figure illustrates the scenario (Attack Scenario-1) discussed in previous page. Attacker will successfully perform fragmentation attack on a host. Attacker plays with the order and time of fragments in order to send the those fragments to victim machine and will successful when the NIDS fragmentation re-assembly timeout is less than the victim's fragmentation reassembly timeout.



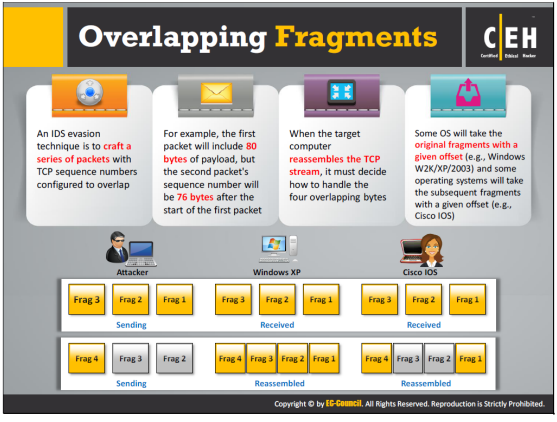
Attack Scenario-2:

Sometimes, IDS fragmentation reassembly timeout is more than fragmentation reassembly timeout of a host. In this scenario, consider that the attacker has fragmented the attack packet into four fragments: frag-1, frag-2, frag-3 and frag-4. Here, the IDS fragmentation reassembly timeout is 60 sec and the fragmentation reassembly timeout for the host is 30 sec.

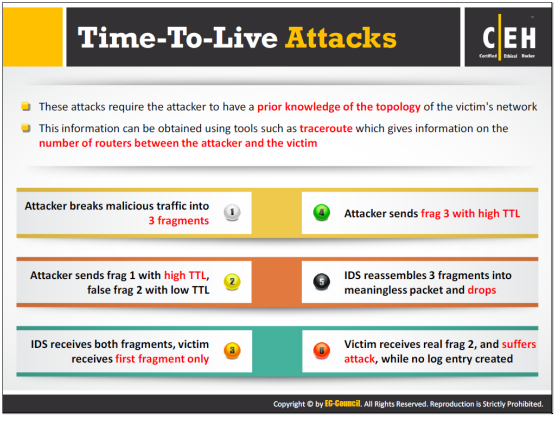
Initially, attacker sends frag-2 and frag-4 with a false payload referred as frag-2' and frag-4', which are received by both the IDS and the victim. The attacker waits until the fragments' reassembly timeout occurs at the victim's system. In this attack, the victim has not received frag-1, so it will drop the fragments without generating an ICMP error message. The attacker then sends a packet (frag-1, frag-3) with a legitimate payload. Now, the victim has only frag-1 and frag-3, whereas the IDS has frag-1, frag-2', frag-3, and frag-4'. Here, frag-2' and frag-4' have false payloads. With the received four fragments, IDS will perform a TCP reassembly but drop the packet, as the computed checksum for frag-2' and frag-4' will be invalid. If the attacker now sends frag-2 and frag-4 again with valid payload, the IDS will have only these two fragments with a valid payload, as the previous fragments will have reassembled and dropped. The victim will have all fragments (frag-1, frag-3, frag-2, frag-4)—with valid payloads that will reassemble—and read the packet as an attack.



The figure illustrates the scenario (Attack Scenario-2) discussed above. The attacker sends the false payload that will falsely reassemble fragments at IDS yet successfully perform fragmentation attack on a host when the NIDS fragmentation reassembly timeout exceeds the victim's fragmentation reassembly timeout.

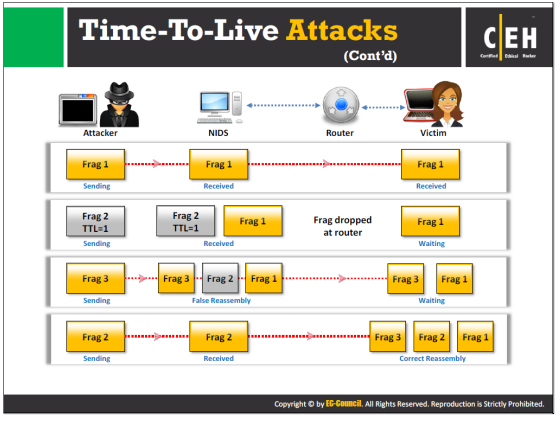


Consider a scenario in which the attacker carries out this attack by breaking the packet into four fragments, sending frag-1, frag-2, and frag-3 first, accepted by both operating systems. Then the attacker sends frag-2', frag-3', and frag-4. Here, the payloads of frag-2' and frag-3' are different from those of frag-2 and frag-3, respectively, but the fragment offset and its length, along with other fields in the IP-header, remain the same. In such a scenario, an operating system such as Windows XP will reassemble frag-1, frag-2, frag-3, and frag-4, whereas an OS such as Cisco IOS will reassemble frag-1, frag-2', frag-3', and frag-4.

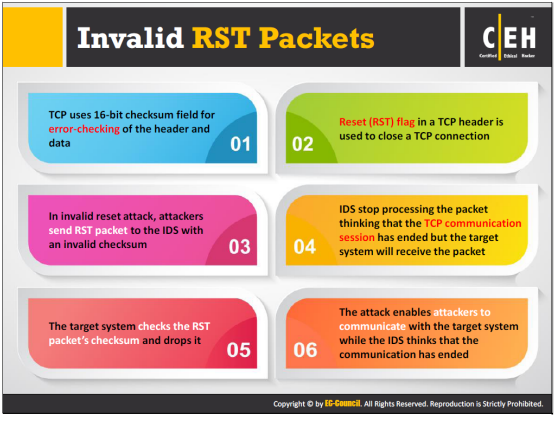


Each IP packet has a field called **Time to Live (TTL)**, which indicates how many hops the packet can take before a network node discards it. Each router along a data path decrements this value by 1. When TTL reaches 0, the packet is dropped, and an ICMP alert notification is sent to the sender. Typically, when a host sends a packet, it sets the TTL to a value high enough that it can reach its destination under normal circumstances. Different operating systems use different default initial values for the TTL. Because of this, attackers can guess the number of routers between them and a sending machine, and make assumptions on what the initial TTL was, thereby guessing which OS a host is running, as prelude to an attack. To prevent such detection, **SmartDefense** can change the TTL field of all packets (or all outgoing packets) to a given number.

Consider a scenario in which a router is present between the IDS and a victim. Attackers need to acquire this information prior to launching the time-to-live attack by breaking the malicious data packet into three fragments. The attacker sends fragment 1 with a large TTL value, which is received by both the IDS and the victim, and then sends second fragment (frag-2') with the TTL value of 1 and a false payload. The IDS receives this fragment, whereas the router (situated between the IDS and the victim) discards it as the TTL value reduces to 0. At this stage, the IDS has only fragment 2, as it has already performed a reassembly and the stream has flushed. The attacker finally sends the second fragment with a valid payload, and the victim performs a reassembly on fragments 1, 2, and 3 and gets the attack. The attacker then sends fragment 3 with a valid TTL, which makes the IDS perform a TCP reassembly on fragments 1, 2', and 3 while the victim waits for the second fragment.

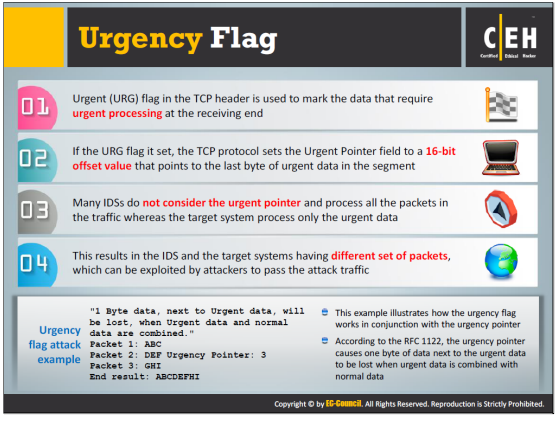


In this scenario, it is assumed that the attacker has prior knowledge about the topology of the target network (i.e., how many routers there are between attacker and victim machines). The attacker fragments the packet and sends frag 1 with the TTL set to a higher value. It is then received by the victim and the IDS. Then, the attacker sends frag-2' with false payload and a TTL value of 1, which is received by the IDS; however, the victim will not receive it, because the router discards it, because the TTL value is reduced to 0. Next, the attacker sends frag-3 with a correct payload and a higher TTL value, which enables it to reach the IDS and the victim. After receiving frag-3, the IDS perform a TCP reassembly on fragments 1, 2', and 3, and the victim waits for frag-2. Finally, the attacker sends frag-2 with a valid payload. The victim, after receiving frag-2 reassembles fragments 1, 2, and 3 and gets the attack code embedded in a false payload. Here, the IDS has only frag-2, as it already has reassembled the fragments and the stream has cleared.

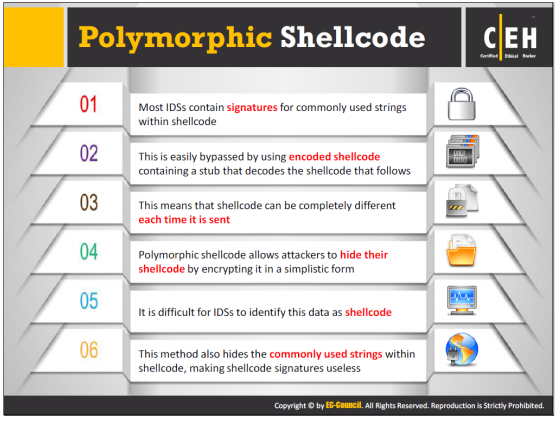


The TCP protocol uses checksums to ensure that communication is reliable. It adds a checksum to every transmitted segment that is checked at the receiving end. When a checksum differs from the checksum expected by the receiving host, the TCP protocol drops the packet at the receiver's end. The TCP protocol also uses an RST packet to end two-way communications. Attackers can use this feature to elude detection by sending RST packets with an invalid checksum, which causes the IDS to stop processing the stream because the IDS thinks the communication session has ended. However, the end host sees this packet and verifies the checksum value, then drops the packet if it is invalid.

Some IDS systems might interpret this packet as an actual termination of the communication and stop reassembling the communication. Such instances allow attackers to continue to communicate with the end host while confusing the IDS because the end host accepts the packets that follow the RST packet with an invalid checksum value.

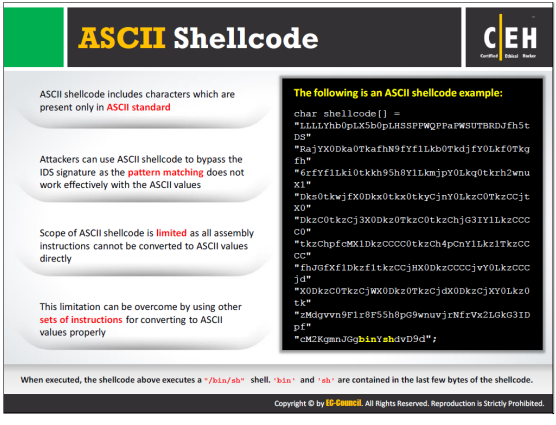


The urgency flag in the TCP protocol marks data as urgent. TCP uses an urgency pointer that points to the beginning of urgent data within a packet. When the user sets the urgency flag, TCP protocol ignores all data before the urgency pointer, and the data to which the urgency pointer points is processed. Some IDSes do not take into account the TCP protocol's urgency feature, which could allow attackers to evade the IDS, as seen in other evasion techniques. Attackers can place garbage data before the urgency. The pointer and the IDS read that data without consideration for the end host's urgency flag handling. This means the IDSs have more data than the end host actually processes.



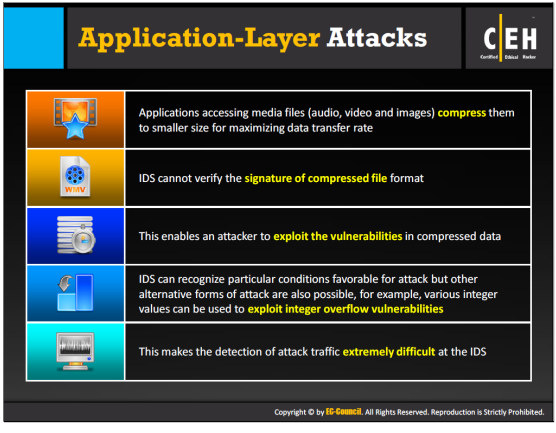
A signature-based network intrusion detection system (NIDS) identifies an attack by matching attack signatures with incoming and outgoing data packets. Most IDSs contain signatures for commonly used shellcode strings that can be bypassed by using encoded shellcodes.

With polymorphic shellcodes, attackers hide their shellcode (attack code) by encrypting it with an unknown encryption algorithm and including the decryption code as part of the attack packet. To carry out polymorphic shellcode attacks, they use an existing buffer-overflow exploit and set the "return" memory address on the overflowed stack to the entrance point of the decryption code. This makes it difficult for the IDS to identify it as shellcode. Therefore, when attackers modify/transform their attacks in this way, the NIDS cannot identify them. This method also hides the commonly used shellcode strings, thus making the shellcode signatures unusable.



ASCII shellcode contains only characters from the ASCII standard. This form of shellcode allows attackers to bypass commonly enforced character restrictions within string input code. It also helps attackers bypass IDS pattern matching signatures, because shellcode hides strings in a similar way to polymorphic shellcode.

Using ASCII for shellcode is very restrictive, in that it limits what the shellcode can do under some circumstances, as not all assembly instructions convert directly to ASCII values. This restriction bypasses using other instructions, or a combination of instructions, that convert to ASCII character representation, which serves the same purpose as those instructions that improperly convert.



Media files such as images, audios, and videos can be compressed so that they rapidly transfer as smaller chunks. Attackers find flaws in this compressed data and perform attacks; even the IDS signatures cannot identify attack code within data thus compressed.

Many applications that deal with such media files employ some form of compression, resulting in an increase in data transfer speed. When you find a flaw in these applications, the entire attack can occur within compressed data, and the IDS can have no way to check the compressed file format for signatures. Many IDSs look for specific conditions that allow for an attack. However, there are times when the attack can take many different forms. For example, attackers can exploit the integer overflow vulnerabilities using several different integer values. This fact, combined with compressed data, makes signature detection extremely difficult.

Desynchronization – Pre-Connection SYN

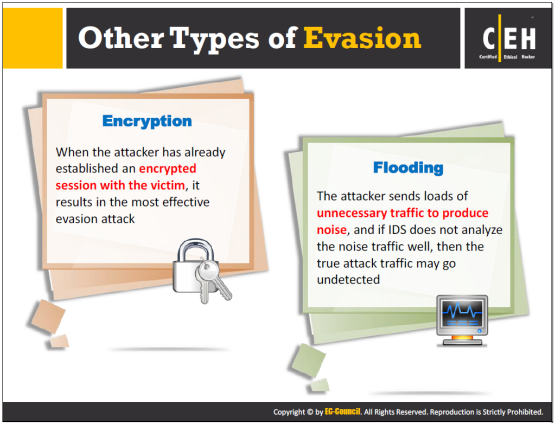
01 If a SYN packet is received **after the TCP control block is opened**, the IDS resets the appropriate sequence number to match that of the newly received SYN packet

02 Attackers send **fake SYN packets** with a completely invalid sequence number to desynchronize the IDS

03 This **stops IDS** from monitoring all, legitimate and attack, traffic

Desynchronization – Post-Connection SYN

1. For this technique, attempt to **desynchronize the IDS** from the actual sequence numbers that the kernel is honoring
2. Send a **post connection SYN packet** in the data stream, which will have **divergent sequence** numbers, but otherwise meet all of the necessary criteria to be accepted by the target host
3. However, the target host will ignore this **SYN packet**, as it references an already established connection
4. The intent of this attack is to get the IDS to **resynchronize** its notion of the sequence numbers to the new SYN packet
5. It will then ignore any data that is a **legitimate part of the original stream**, because it will be awaiting a different sequence number
6. Once succeeded 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



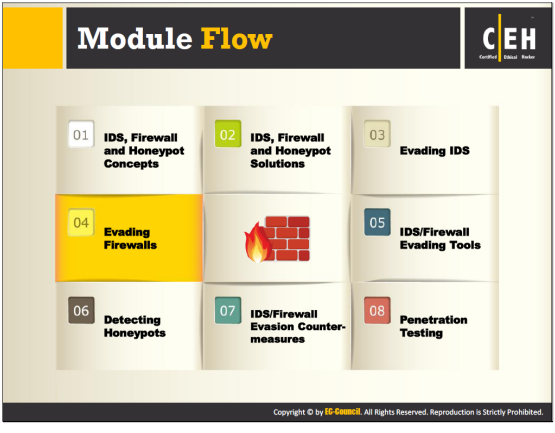
There are two more types of evasion:

Encryption

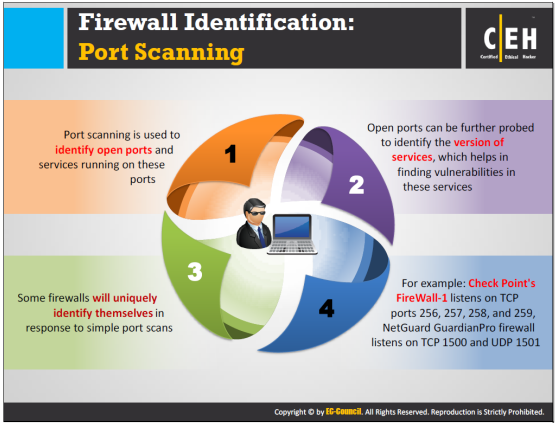
Network-based intrusion detection analyzes traffic in the network from source to destination. If an attacker succeeds in establishing an encrypted session with his/her target host using a secure shell (SSH), secure socket layer (SSL), or a virtual private network (VPN) tunnel, the IDS will not analyze the packets going through these encrypted communications. Thus, he/she can send the malicious traffic using this secure channel, thus evading IDS security.

Flooding

IDSs make use of resources such as memory and processor speed to analyze the traffic going through them. To bypass IDS security, attackers flood IDS's resources with noise or fake traffic to exhaust them with having to analyze flooded traffic. Once such attacks succeed, attackers then send malicious traffic toward the target system behind the IDS, which offers little or no intervention. Thus, true attack traffic might go undetected.



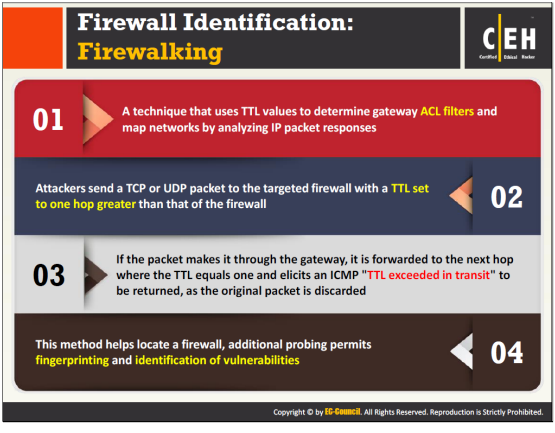
The previous section explains how attackers use various techniques to bypass IDSs. Similarly, they can also use various tricks and techniques to bypass firewalls. This section discusses the different techniques attackers use to bypass firewall security.



Ports are places from which computers send or accepts information from network resources. Finding open ports is an attacker's first step toward access to the target system. To do so, the attacker systematically scans the target's ports, to identify possible vulnerabilities. Attackers sometimes use automated port-scanning utilities to do so, many of which are available.

How Attackers Scan Ports

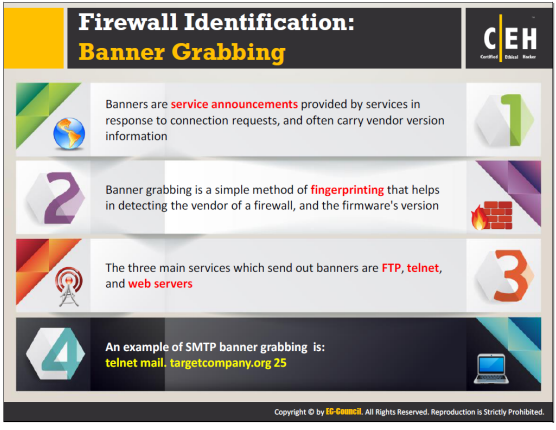
Port-scanning consists of sending messages to each port, one at a time. The kind of response received indicates whether the system is using the port, and leaving it open to the discovery of weaknesses. Some firewalls will uniquely identify themselves using simple port scans. For example, Check Point's FireWall-1 listens on TCP ports 256, 257, 258, and 259 and Microsoft's Proxy Server usually listens on TCP ports 1080 and 1745.



Firewalking is a method used to collect information about remote networks behind firewalls. It probes ACLs on packet filtering routers/firewalls using the same method as tracerouting. Firewalking involves sending TCP or UDP packets into the firewall with TTL value is one hop greater than the targeted firewall. If the packet makes it through the gateway, the system forwards it to the next hop, where the TTL equals one and prompts an ICMP error message at the point of rejection with a "TTL exceeded in transit" message. Using this method, possible access to the firewall can be determined if successive probe packets are sent.

Firewalk is a well-known application used for firewalking. It has two phases: a network discovery phase and a scanning phase. It requires three hosts:

- **Firewalking host:** The firewalking host is the system outside the target network, from which the data packets are sent to the destination host to gain more information about the target network.
- **Gateway host:** The gateway host is the suspected firewall system on the target network, through which the data packet passes on its way to the target network.
- **Destination host:** The destination host is the target system on the target network to which the data packets are addressed.



Banners are announcements generated by services in response to access attempts. They identify which service is running on the system. Attackers use banner grabbing to fingerprint services and thereby discover what services are running on firewalls. The three main services that send out banners are FTP, Telnet, and web servers.

A firewall does not block banner grabbing, because the connection between the attacker's system and the target system looks legitimate. An example of SMTP banner grabbing is telnet mail.targetcompany.org 25.

The syntax is "`<service name > <service running > <port number>`"

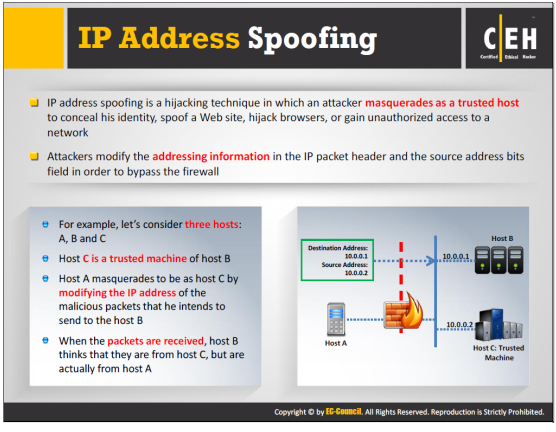
Banner grabbing used for specifying banners and application information. For example, when the user opens a telnet connection to a known port on the target server and presses Enter a few times, if required, it displays the following result:

C:\>telnet www.corleone.com 80

HTTP/1.0 400 Bad Request

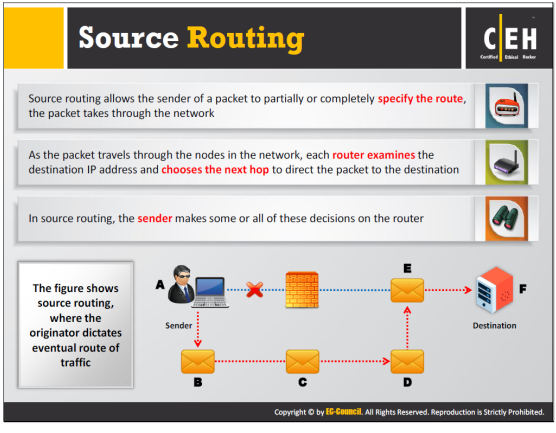
Server: Netscape – Commerce/1.12

This system works with many other common applications that respond on a set port. The information generated through banner grabbing can enhance the attacker's efforts to further compromise the system. With information about the version and the vendor of the web server, the attacker can further concentrate on employing platform-specific exploit techniques. Services on ports of such as FTP, Telnet, and web servers should not kept open, as they are vulnerable to banner grabbing.



Most of the firewalls filter packets based on the source IP address. These firewalls examine the source IP address and decides whether the packet is coming from legitimate source or illegitimate source. The IDS filters packets from illegitimate sources. Attackers use the IP spoofing technique to bypass such firewalls.

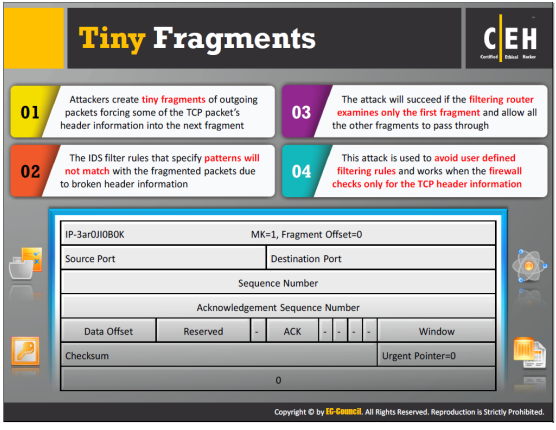
IP address spoofing or IP spoofing is a technique in which the attacker makes use of someone's IP address to hide his/her identity. In IP spoofing, the attacker creates IP packets by using a forged IP address and gains access over the system or network without authorization. The attacker spoofs the messages; therefore, the destination host feels that it has come from a reliable source. Thus, the attacker succeeds in impersonating others' identities with the help of IP spoofing. Hackers generally use this technique to avoid detection during spamming and various other activities.

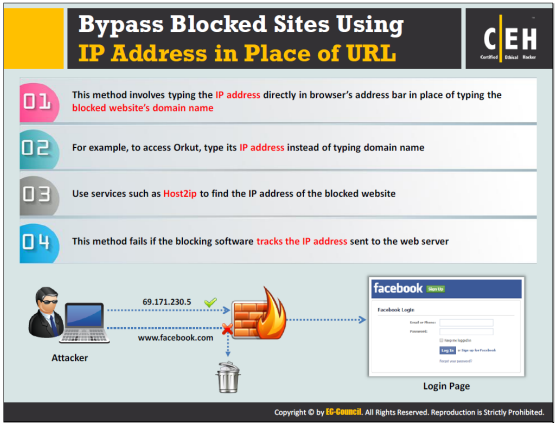


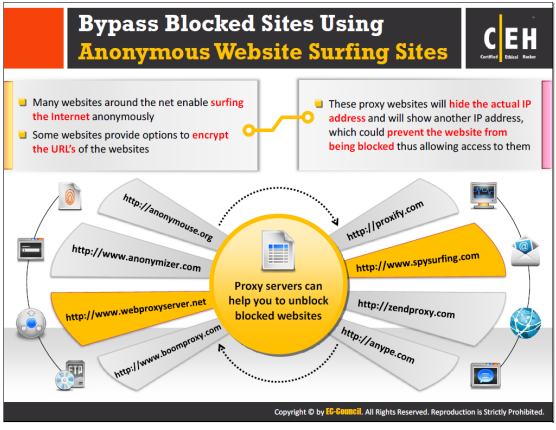
Using this technique, the sender of the packet designates the route that a packet should take through the network, in such a way that the designated route should bypass the firewall node. Using this technique, the attacker can evade firewall restrictions.

When these packets travel through the network nodes, each router will check the IP address of the destination and choose the next node to which to forward them. In source routing, the sender makes some or all of these decisions on the router.

Source routing takes two approaches: loose source routing, and strict source routing. In loose source routing, the sender specifies on or more stages the packet must go through, whereas in strict source routing, the sender specifies the exact route the packet must go through.







Anonymous web-surfing sites help you to browse the Internet anonymously and unblock blocked sites (i.e., evade firewall restrictions). By using these sites, you can surf restricted sites anonymously, without using your IP address. There are a number of anonymous web-surfing sites available, some of which provide options to encrypt website URLs.

Here is a list of eight proxy servers that can help you to access blocked websites:

Anonymous Website Surfing Site 1: (*http://anonymouse.org*)

This service allows you to surf the web without revealing any personal information.

Anonymous Website Surfing Site 2: (*http://www.anonymizer.com*)

Anonymizer Universal keeps your online activities safe, private, and secure.

Anonymous Website Surfing Site 3: (*http://www.webproxyserver.net*)

Webproxyserver.net is an SSL or secured proxy server that helps user to change their IP address online and protect your identity. It has the capability to bypass restrictions thus giving more advantage to the user.

Anonymous Website Surfing Site 4: (*http://www.boomproxy.com*)

It is a Free Proxy Browser that secures the anonymous sites against a variety of internet threats. Its aim is to maximize the anonymous virtual browsing session on a reliable proxy system platform.

Anonymous Website Surfing Site 5: (*http://proxify.com*)

Proxify is an anonymous proxy service which allows anyone to surf the Web privately and securely. Through Proxify, you can use websites but they cannot uniquely identify or track you. Proxify hides your IP address and its encrypted connection prevents monitoring of your network traffic.

Anonymous Website Surfing Site 6: (*http://www.spysurfing.com*)

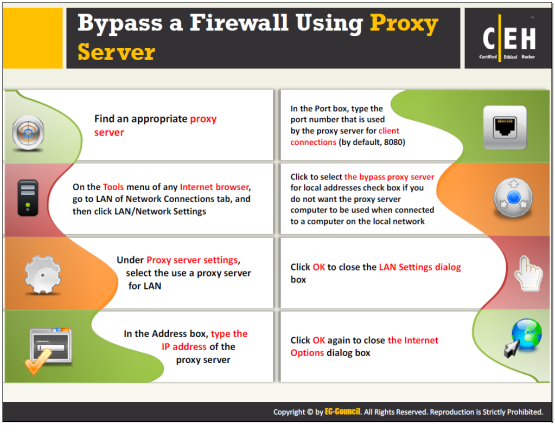
SpySurfing is a FREE anonymous web based proxy service. It ensures your privacy by letting you browse the Web as an anonymous user. Web sites often track, log, and analyze your IP address, geographical information, web browser, and other personal information. Spysurfing helps you avoid giving out this information by hiding your personal information such as your IP address from web sites.

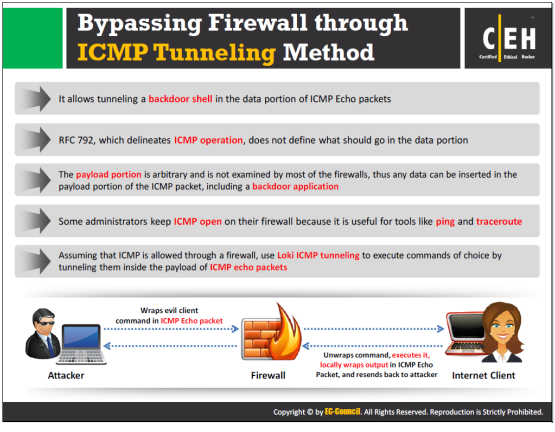
Anonymous Website Surfing Site 7: (*http://zendproxy.com*)

It is a Web Proxy for unblocking. An online web proxy is a computer system or website that acts as the middle person between user and the Internet. A web proxy allows users to surf the Internet anonymously as the online anonymizer hides the IP address of the user. By hiding the IP address of the user, the online anonymous proxy strips the IP address from the user's request to view that content and is then able to view it without restrictions.

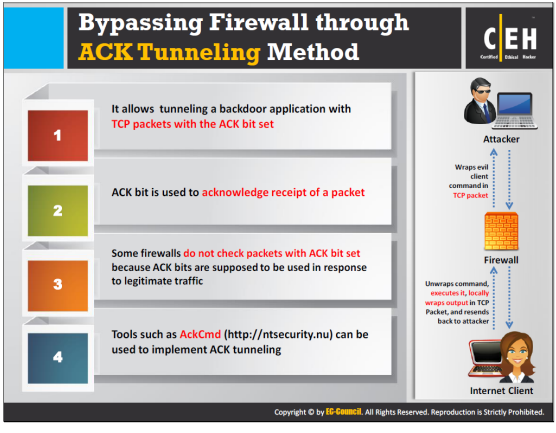
Anonymous Website Surfing Site 8: (*http://anype.com*)

An anonymous surfing system offers anonymity from its servers to web page you open. Using this system, all data at first transmitted to its servers, and then it requests your page. Response data also goes through its servers.

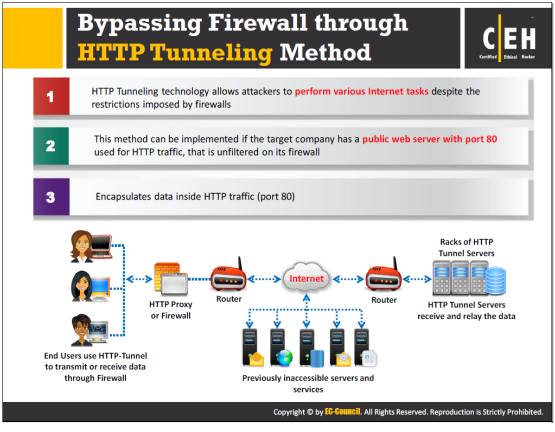




ICMP protocol is used to send error message to the client. As it is required service for network communication, therefore user often enables this service on their networks. Moreover, it does not cause a big threat from the security perspective. Attacker takes the advantage of enabled ICMP protocol on network and performs ICMP tunneling to send his/her malicious data into the target network. ICMP Tunnel provides attackers with full access to target networks.

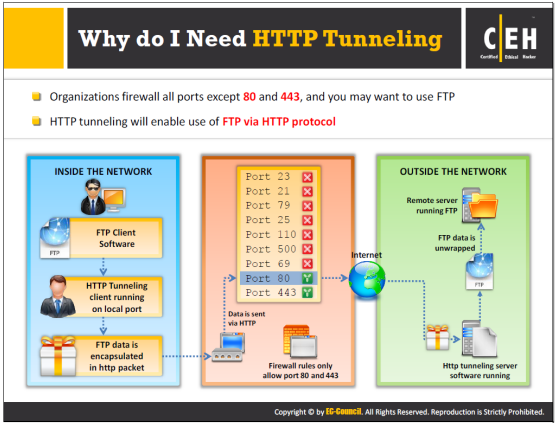


Ordinary packet filtering firewalls defined their rule sets based on SYN packet when TCP level communication is going to establish. This is because such firewall assumes that only SYN packet is coming from the client and thus has possibility of containing malicious code in SYN packet. These firewall ignores the possibility that attacker can also inject malicious code in ACK packet. As ACK packets are sent after establishing a session, ACK traffic is considered legitimate. Another reason why filtering of ACK packets is ignored is to lessen the workload of firewalls, as there can be many ACK packets for one SYN packet. ACK tunneling allows tunneling a backdoor application with TCP packets with the ACK bit set. The ACK bit acknowledges the receipt of a packet. As stated earlier, some firewalls do not check packets with the ACK bit set, because ACK bits are supposed to be used in response to legitimate traffic that has already been allowed to pass through. Attackers use this as an advantage in ACK tunneling. Tools such as **AckCmd (http://ntsecurity.nu)** use ACK tunneling.



This method can be implemented if the target company has a public web server with port 80 used for HTTP traffic and is unfiltered by its firewall. Many firewalls do not examine the payload of an HTTP packet to confirm that it is legitimate, thus it is possible to tunnel traffic via TCP port 80.

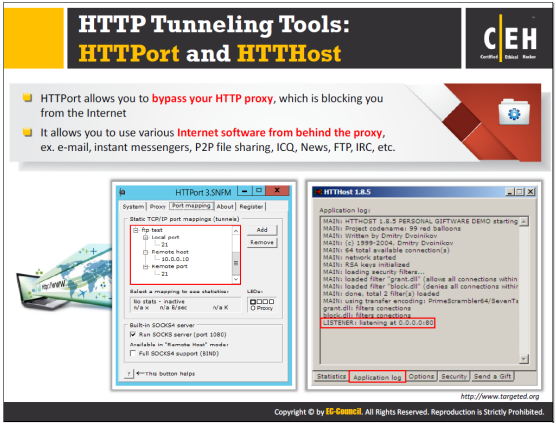
Tools such as **HTTPTunnel (http://www.nocrew.org)** use this technique of tunneling traffic across TCP port 80. HTTPTunnel is a client/server application, the client application is htc, and the server is hts. Upload the server onto the target system, and redirect it through TCP port 80.



HTTP Tunneling is used in scenarios in which network users are granted restricted connectivity through a firewall or proxy; in such conditions, some applications may also lack native communications support. These restrictions include:

- Blocking of TCP/IP ports, traffic initiated from outside the network, and, network protocols except a few commonly used protocols, etc.
- Access to surf denied websites
- Post in forums anonymously by hiding the IP address
- To use application such as chatting through ICQ or IRC, instant messengers, games, browsers, etc.
- Sharing of confidential resource over HTTP securely
- Downloading files with filtered extensions and/or with malicious code

For instance, consider that organization firewalls restrict users to access all ports except 80 and 443, and a user may want to use FTP. HTTP tunneling enables FTP use via HTTP protocol. HTTP Tunnel creates a bidirectional virtual data connection tunneled in HTTP traffic. It works with the help of FTP client software to perform protocol encapsulation by enclosing data packets of one protocol such as SOAP or JRMP within HTTP packets on, for example, local port 80. These packets then are sent through the firewall or proxy server as normal Internet traffic, which is then directed to HTTP Tunneling server software located outside the network. Upon receiving the packets, this server then unwraps FTP data and redirects the packet to the remote FTP server.

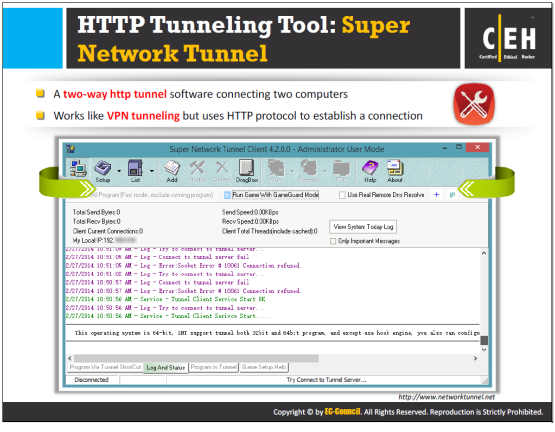


HTTPort allows users to bypass the HTTP proxy, which blocks Internet access to e-mail, instant messengers, P2P file sharing, ICQ, News, FTP, IRC, and so on. Here, the Internet software is configured so that it connects to a local PC as if it is the required remote server. HTTPort then intercepts that connection and runs it through a tunnel through the proxy. HTTPort can work on devices such as proxies or firewalls that allow HTTP traffic. Thus, the HTTPort provides access to websites and Internet apps. HTTPort performs tunneling using one of two modes: SSL/CONNECT mode, and remote host.

In SSL/CONNECT mode, HTTPort can make a tunnel through a proxy all by itself. It requires that the proxy should support a certain HTTP feature, specifically CONNECT HTTP. Most proxies have this method disabled by default. SSL/CONNECT mode is much faster, but encryption cannot be used and the proxy can track all actions.

The remote host method is capable of tunneling through any proxy. HTTPort uses a special server software called HTTHost, which is installed outside the proxy-blocked network. It is a web server, and thus when HTTPort is tunneling, it sends a series of HTTP requests to the HTTHost. The proxy responds as if the user is surfing a website and thus allows the user to do so. HTTHost in turn performs its half of the tunneling and communicates with the target servers. This mode is much slower, but works in most cases and features strong data encryption that makes proxy logging useless.

Source: *http://www.targeted.org*

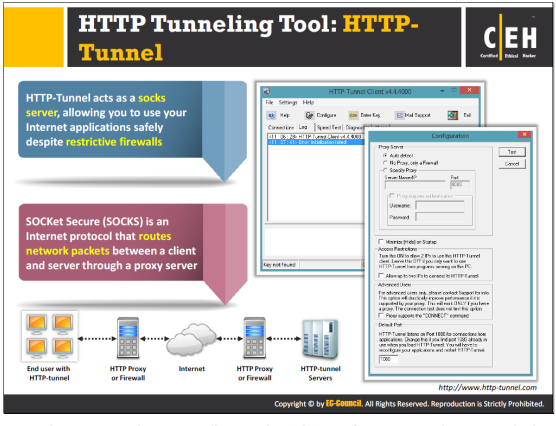


Super Network Tunnel is two-way HTTP tunneling software that connects two computers utilizing HTTP-Tunnel Client and HTTP-Tunnel Server. It works like VPN tunneling but uses HTTP protocol to establish a connection for accessing Internet without monitoring and gives an extra layer of protection against attackers, spyware, identity theft, and so on. It can bypass any firewall to surf the web, use IM applications, games, and so on. Super network tunnel integrates SocksCap function along with bidirectional HTTP tunnelling and remote control to simplify the configuration.

This tool allows HTTP, HTTPS, and SOCKS tunneling of any TCP communication between any client-server systems. The TCP traffic sent from the client to the server via standard HTTP POST requests, which allows penetrating through firewalls, proxy servers, and so on where HTTP traffic generally passes.

The client side of a tunnel is the Super Network Tunnel client app, which listens on a particular TCP port for incoming requests. Once the request comes, the program creates an HTTP/HTTPS tunnel to the server and sends data through it. The server side is a Super Network Tunnel Server, which simply forwards the data to the intended recipient app running on the server computer or LAN. Both client and server sides support multiple tunnels and multiple connections through the same tunnel at the same time.

Source: *http://www.networktunnel.net*



HTTP-Tunnel acts as a socks server, allowing the safe use of Internet applications, which will go un-monitored despite restrictive firewalls; it encrypts all Internet traffic to provide additional security against attackers, spyware, ID theft, and so on. SOCKet Secure (SOCKS) is an Internet protocol that routes network packets between a client and server through a proxy server. HTTP-Tunnel technology allows users to perform various Internet tasks despite the restrictions imposed by firewalls. This is made possible by sending data via HTTP on port 80.

The various applications utilized to implement HTTP-Tunnel technology are:

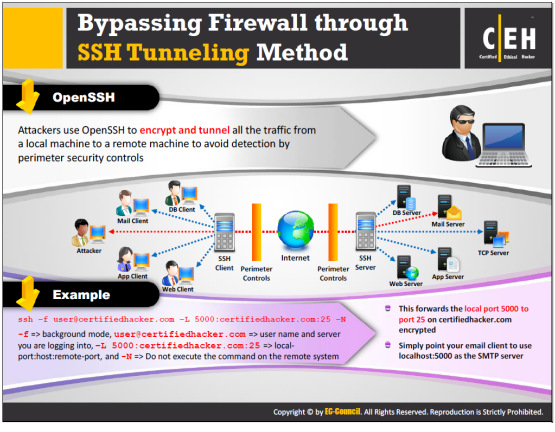
HTTP-Tunnel Client: An application that runs in the system tray acting as a SOCKS server, managing all data transmissions between the computer and the network.

HTTP-Tunnel Server: A customizable server software solution for both personal and corporate networks.

HTTP-Tunnel ActiveX Control: Allows developers to incorporate HTTP-Tunnel technology into their software applications.

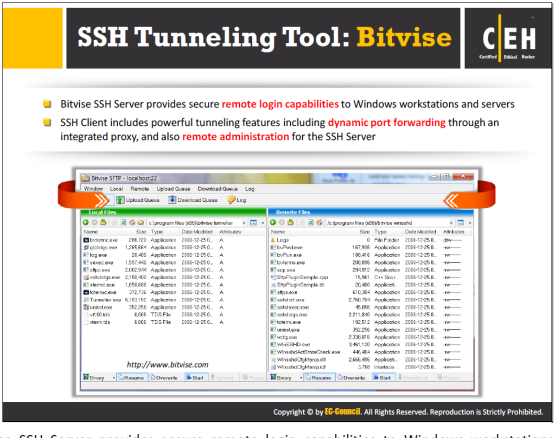
Corporate Messenger (VCM): It is a secure and server less instant messaging application, suitable for corporate intranets.

Source: *http://www.http-tunnel.com*



SSH protocol tunneling involves sending unencrypted network traffic through an SSH tunnel. For example, suppose you want to transfer files on an unencrypted FTP protocol, but the FTP protocol is blocked on the target firewall. The unencrypted data can be sent over encrypted SSH protocol using SSH tunneling. Attackers make use of this technique to bypass firewall restrictions. They connect to external SSH servers and create SSH tunnels to port 80 on the remote server, thereby bypassing firewall restrictions.

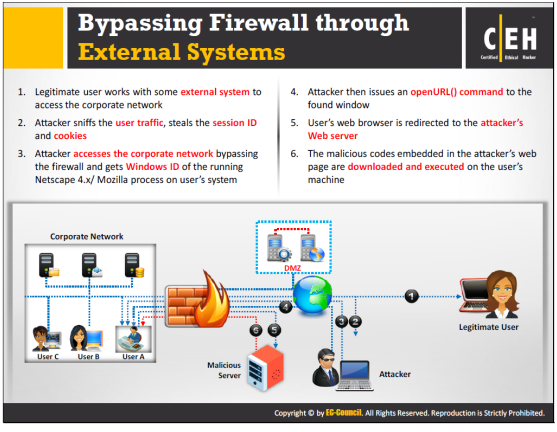
Attackers makes use of OpenSSH (OpenBSD Secure Shell) to encrypt and tunnel all traffic from a local machine to a remote machine to avoid detection by perimeter security controls. OpenSSH is a set of computer programs that provides encrypted communication sessions over a computer network using the SSH protocol.



Bitvise SSH Server provides secure remote login capabilities to Windows workstations and servers by encrypting data during transmission. It is ideal for remote administration of Windows servers; for advanced users who wish to access their home machine from work, or their work machine from home; and for a wide spectrum of advanced tasks, such as establishing a VPN using the SSH TCP/IP tunneling feature or providing a secure file depository using SFTP.

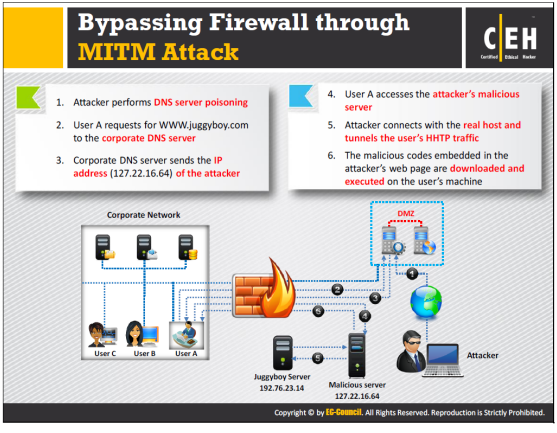
Bitvise SSH Client for Windows includes terminal emulation, graphical as well as command-line SFTP support, an FTP-to-SFTP bridge, tunneling features—including dynamic port forwarding through integrated proxy—and remote administration for SSH Server.

Source: http://www.bitvise.com

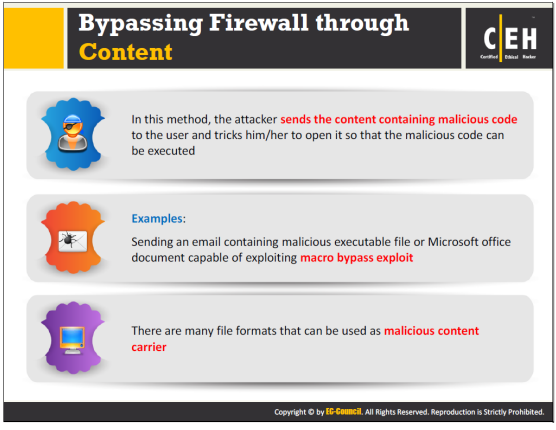


Attackers can bypass firewall restrictions of target networks from any external system that can access the internal network. This external system can be:

- Home machine of employee
- Machine that does remote administration of target network
- Machine from company's network but located at different place



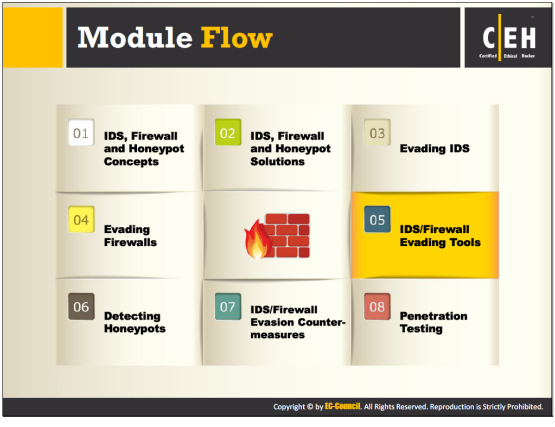
Most security administrators concentrate on the possibility of an external or internal network bypassing their firewall while ignoring the fact that firewalls can be bypassed using MITM attacks on DNS servers. In MITM attacks, attackers make use of DNS servers and routing techniques to bypass firewall restrictions. They may either take over the corporate DNS server or spoof DNS responses to perform the MITM firewall attack.



Attackers can also target WWW/FTP servers and embed Trojan horse files as software installation files, mobile phone software, and so on to lure users to access them. There are many file formats for text, multimedia, and graphics content that can be used to carry malicious content.

Commonly used file formats for carrying malicious contents are:

- EXE,COM,BAT,PS, PDF CDR (Corel Draw)
- DVB,DWG (AutoCad)
- SMM (AMI Pro)
- DOC,DOT,CNV,ASD (MS Word)
- XLS,XLB,XLT (MS Excel)
- ADP, MDA,MDB,MDE,MDN,MDZ (MS Access)
- VSD (Visio)
- MPP,MPT (MS Project)
- PPT,PPS,POT (MS PowerPoint)
- MSG,OTM (MS Outlook)



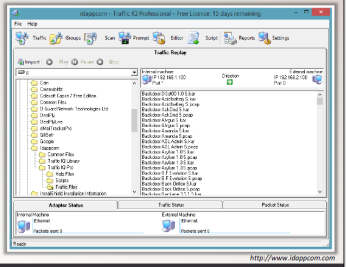
During Firewall evasion, attackers use various security-auditing tools that assess firewall behavior. This section features and enlists some of these tools that help attackers bypass firewall restrictions. They automate the process of bypassing firewall rules while increasing effectiveness and consuming less time.

IDS/Firewall Evasion Tool: Traffic IQ Professional

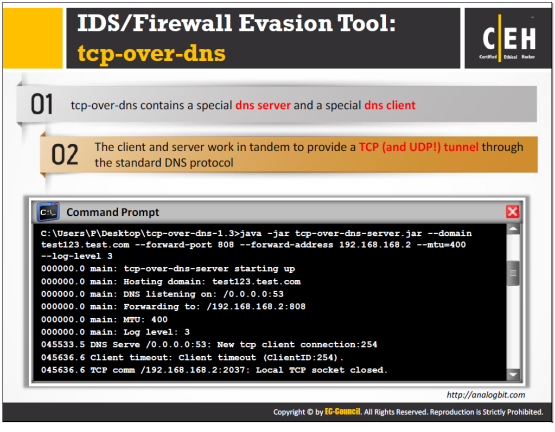
Traffic IQ Professional enables security professionals to **audit and validate the behavior of security devices** by generating the **standard application traffic or attack traffic** between two virtual machines

Traffic IQ Professional can be used to **assess**, **audit**, and **test the behavioral characteristics** of any non-proxy packet-filtering device including:

- Application firewall systems — 01
- Intrusion detection systems — 02
- Intrusion prevention systems — 03
- Routers and switches — 04



http://www.idappcom.com



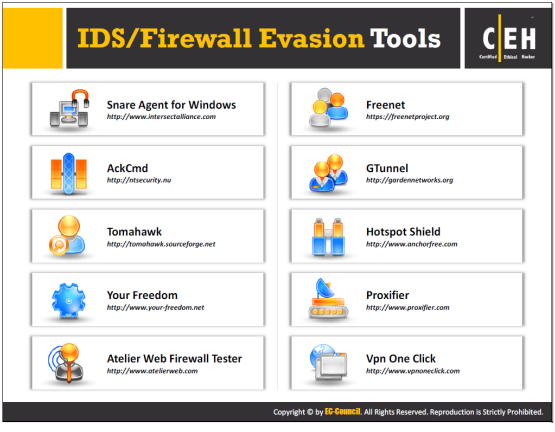
IDS/Firewall Evasion Tool: tcp-over-dns

tcp-over-dns is similar to the defunct NSTX dns tunneling software. The purpose of this software is to succeed where NSTX failed. All NSTX tunnels disconnect within tens of seconds in real-world situations. tcp-over-dns is quite robust while at the same time providing acceptable bandwidth speeds.

It features include:

- Windows, Linux, Solaris compatibility
- Sliding window packet transfers for increased speed and reliability
- Runtime selective LZMA compression
- TCP and UDP traffic tunneling

Source: http://analogbit.com



Other than those discussed in above pages, there are many utilities available, with the help of which attacker can try to bypass firewall rules. Below is the list of firewall evasion tools:

Snare Agent for Windows

Source: *http://www.intersectalliance.com*

Snare for Windows interacts with the underlying Windows Eventlog subsystem to facilitate remote, real-time transfer of event log information. It captures all windows event logs, from the Security, Application and System logs, as well as the DNS, File Replication Service, and Active Directory logs.

AckCmd

Source: *http://ntsecurity.nu*

AckCmd is a backdoor client/server combination that allows you to open a remote command prompt to another system (running the server part of AckCmd). It communicates using only TCP ACK segments. This way the client component is able to contact directly the server component through a firewall, in some cases (static packet filters).

Tomahawk

Source: *http://tomahawk.sourceforge.net*

Tomahawk is a command-line tool for testing network-based intrusion prevention systems (NIPS). It can test the blocking capabilities of NIPS by replaying attacks embedded in packet

traces. It can test the throughput of NIPS using the most realistic mix of protocols possible: one obtained by taking a sample of traffic from the network and replaying it. It can also test the connections-per-second rating of NIPS.

Your Freedom

Source: *http://www.your-freedom.net*

Your Freedom is a VPN tunneling, firewall and proxy bypassing, and anti-censorship solution. It services makes accessible what is normally inaccessible and hides your network address from others.

Atelier Web Firewall Tester

Source: *http://www.atelierweb.com*

AWFT probes the protection provided by your personal firewall software using six different tests. Each test uses a different technique for gaining access to the outside world.

Freenet

Source: *https://freenetproject.org*

Freenet allows you to anonymously share files, browse and publish "freesites" (web sites accessible only through Freenet), and chat on forums, without fear of censorship.

GTunnel

Source: *http://gardennetworks.org*

GTunnel is works as a local HTTP or SOCKS proxy server. After setting proxy to GTunnel in web browser or other Internet applications, the traffic will go through GTunnel and our server farm before it reaches its destination.

Hotspot Shield

Source: *http://www.anchorfree.com*

Hotspot Shield VPN bypasses internet censorship and restrictions. It instantly unblock YouTube, unblock Facebook, or unblock websites, and get access to geo-restricted sites and VOIP applications.

Proxifier

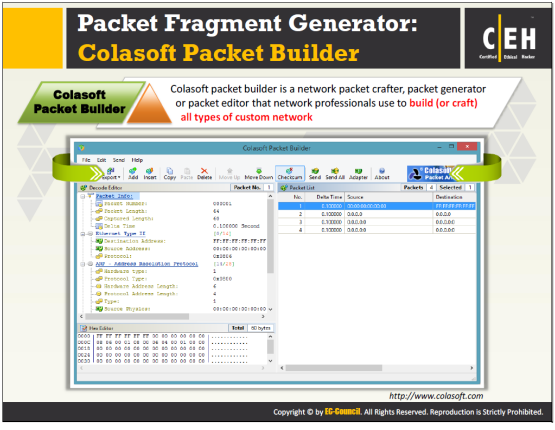
Source: *http://www.proxifier.com*

Proxifier allows you to access the Internet from a restricted network through a proxy server gateway and bypass firewall restrictions.

Vpn One Click

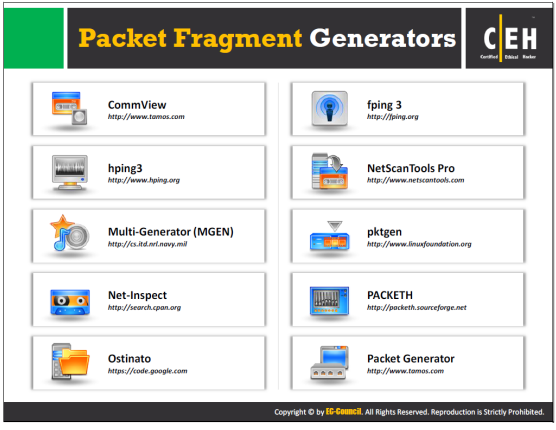
Source: *http://www.vpnoneclick.com*

Vpn One Click is a VPN service that enables you to connect anonymously to any public/shared network. It allows you to send and receive data through an encrypted point-to-point connection with full security, privacy, and functionality (Twitter, Facebook, YouTube, Skype, VoIP, etc.) by connecting to a VPN Server.



Colasoft Packet Builder is used to create custom network packets and fragmenting packets. Attackers use this tool to create custom malicious packets and fragment them in such a way that firewalls will not detect them. They can create custom network packets such as Ethernet Packet, ARP Packet, IP Packet, TCP Packet, and UDP Packet. You can use this tool to check your network's protection against attacks and intruders.

Source: http://www.colasoft.com



As with Colasoft Packet Builder, there are various Packet fragment generators that attackers use to perform fragmentation attacks on firewalls to bypass them. A few such generators are:

CommView

Source: *http://www.tamos.com*

CommView is a network monitor and analyzer that captures every packet on the wire to display important information such as a list of packets and network connections, vital statistics, protocol distribution charts, and so on. You can examine, save, filter, import, and export captured packets, and view decodings down to the lowest layer, with full analysis of over 70 widespread protocols. CommView includes a VoIP analyzer for in-depth analysis, recording.

hping3

Source: *http://www.hping.org*

hping3 is a command-line oriented TCP/IP packet assembler/analyzer that is fully scriptable and uses the TCL language. Packets can be received and sent via a binary or string representation describing them. hping3 can generate arbitrary TCP/IP packets, supporting all the IP and TCP options published.

Multi-Generator (MGEN)

Source: *http://cs.itd.nrl.navy.mil*

Multi-Generator (MGEN) is used to perform IP network performance tests and measurements using TCP and UDP/IP traffic.

Its toolset generates real-time traffic patterns so that the network can be loaded in a variety of ways. Attackers use this utility to perform packet fragmenting.

Net-Inspect

Source: *http://search.cpan.org*

Net-Inspect connects various layers of network inspection to analyze data. It works much like Wireshark. Net-Inspect helps attackers carry out fragmentation attacks on firewalls/IDSs.

Ostinato

Source: *https://code.google.com*

Ostinato is network packet crafter/traffic generator and analyzer with a friendly GUI. It has the ability to construct and sends packets of several streams with different protocols at different rates.

fping 3

Source: *http://fping.org*

fping 3 is a program used to send ICMP echo probes to network hosts. Setting the interval for sending ICMP request packets to a larger value can be extremely useful in avoiding intrusion prevention and intrusion detection systems. fping 3 also helps attackers bypass firewall limits.

NetScanTools Pro

Source: *http://www.netscantools.com*

NetScanTools Pro packet generator allows you to construct or build a TCP, UDP, ICMP, CDP (Cisco® Discovery Protocol), ARP/RARP, or RAW packet, and sends one or more packets to a target IPv4 address. This tool helps attackers bypass firewall rules.

Pktgen

Source: *http://www.linuxfoundation.org*

Linux packet generator is a tool to generate packets at very high speed in the kernel, which can be organized with an interface IP, MAC address, inter-packet delay, packet size, port numbers, and so on.

PACKETH

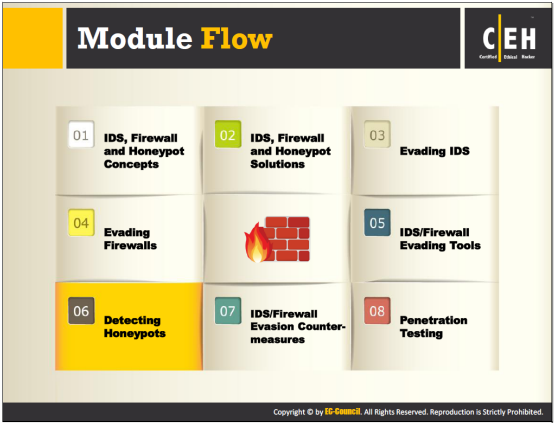
Source: *http://packeth.sourceforge.net*

PACKETH is an ethernet packet generator tool that allows you to create and send any possible packet or sequence of packets on over ethernet. It is very simple to use, powerful, and supports many adjustments of parameters while sending sequences of packets.

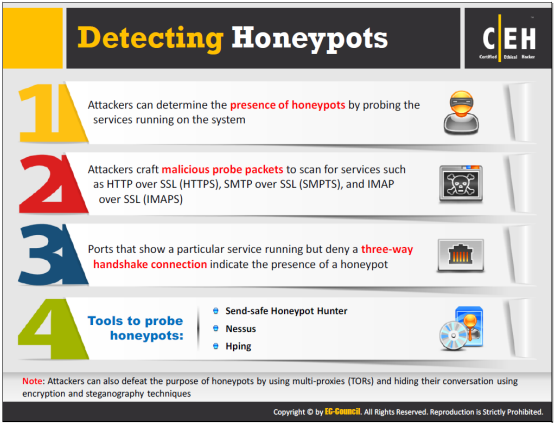
Packet Generator

Source: *http://www.tamos.com*

Packet Generator allows you to edit and send packets via your network card. It is used to send multiple packets at once.



Honeypots are traps set to detect, deflect, or counteract unauthorized intrusion attempts. While attempting to break into the target network, attackers perform honeypot detection using various tools and techniques. This section discusses these tools and the ways in which they are used.



A honeypot is an Internet system designed especially for diverting attackers by tricking or attracting them during attempts to gain unauthorized access to information systems. Attackers use honeypot detection systems or methods to identify the honeypots installed on the target network. Once they detect honeypots, attackers try to bypass them so that they can focus on targeting the actual network.

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:

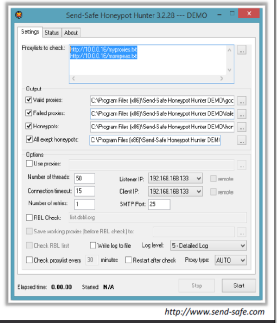
01 Checks lists of **HTTPS**, **SOCKS4**, and **SOCKS5 proxies** with any ports

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

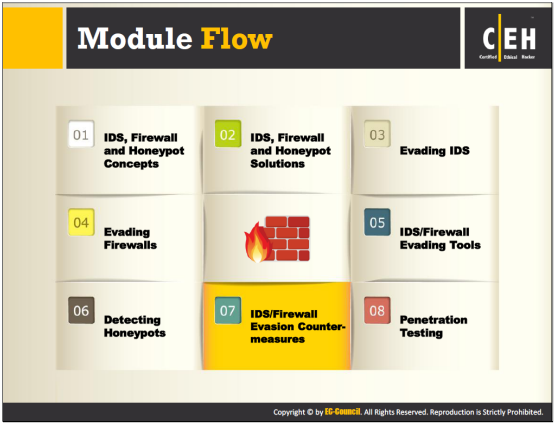
03 Can upload **"Valid proxies"** and **"All except honeypots"** files to FTP

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



The previous sections discussed various tools and techniques attackers use to bypass network security perimeters such as IDSs, firewalls, and honeypots to enter a target network. To avoid attacks on network security perimeters, it is necessary to deploy and configure them securely. This section discusses various countermeasures and best practices for hardening these network security perimeters.

Countermeasures

Shut down switch ports associated with the known attack hosts

Perform an **in-depth analysis** of ambiguous network traffic for all possible threats

Reset (RST) malicious TCP sessions

Look for the **nop opcode** other than 0x90 to defend against the polymorphic shellcode problem

Train users to identify attack patterns and **regularly update/patch** all the systems and network devices

Deploy IDS after a **thorough analysis** of network topology, nature of network traffic, and the number of host to monitor

Countermeasures (Cont'd)

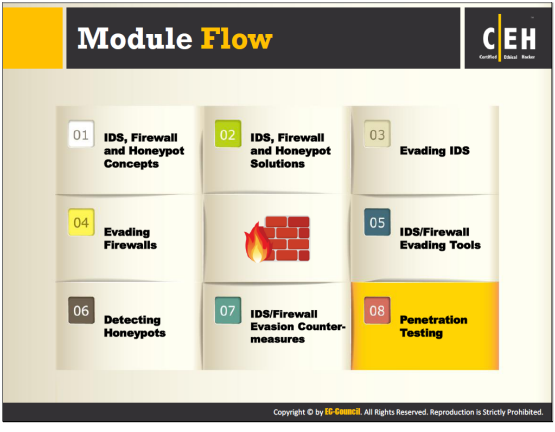
Use a **traffic normalizer** to remove potential ambiguity from the packet stream before it reaches to the IDS

Ensure that IDSs **normalize fragmented packets** and allow those packets to be reassembled in the proper order

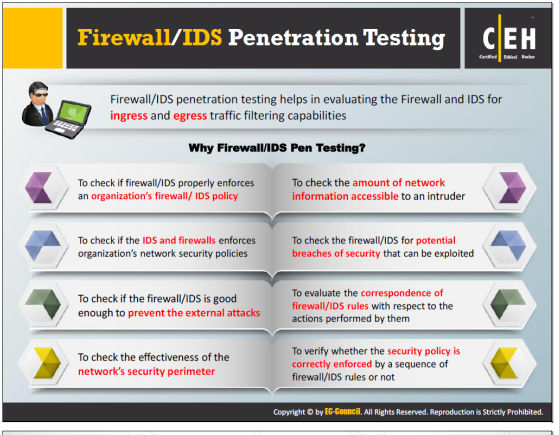
Define DNS server for client resolver in routers or similar network devices

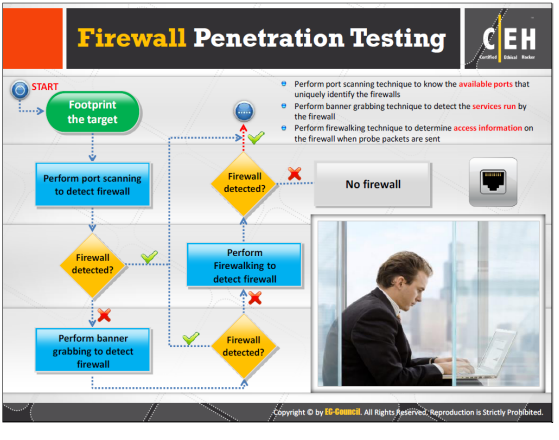
Harden the security of all communication devices such as modems, routers, switches, etc.

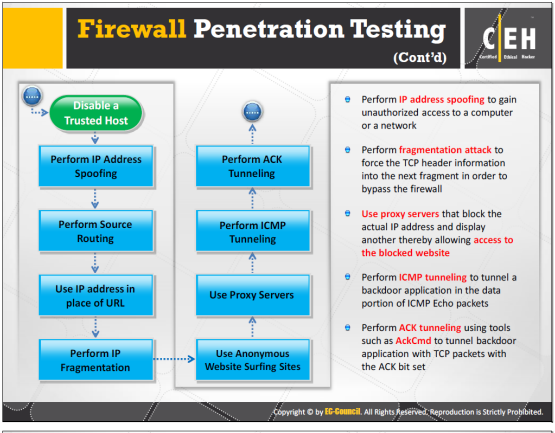
If possible, block **ICMP TTL expired** packets at the external interface level and change the **TTL field to a large value**, ensuring that the end host always receives the packets

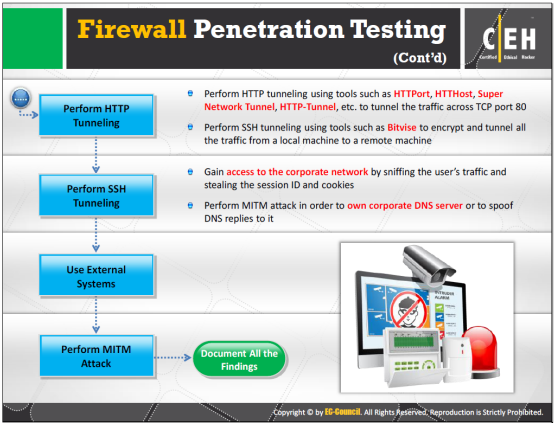


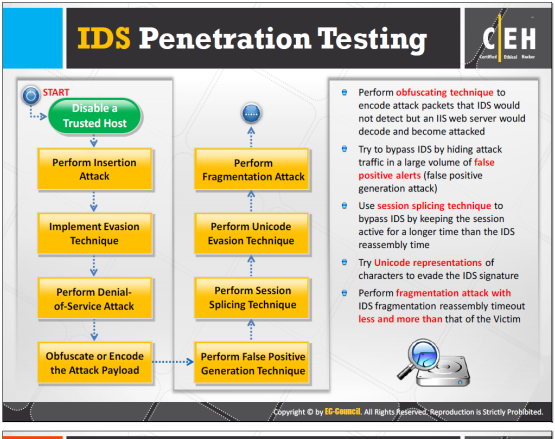
Penetration testing is the process of analyzing a system to determine its weaknesses. Conduct penetration tests on network security perimeters to ensure that they can withstand attackers' bypassing attempts. Penetration testing involves simulating all the possible attacks on network security perimeters in an attempt to bypass them. This section describes and explains the steps required to perform an IDS/firewall/honeypot penetration test.

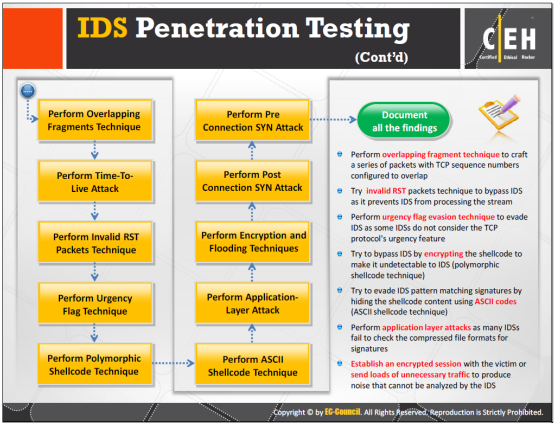


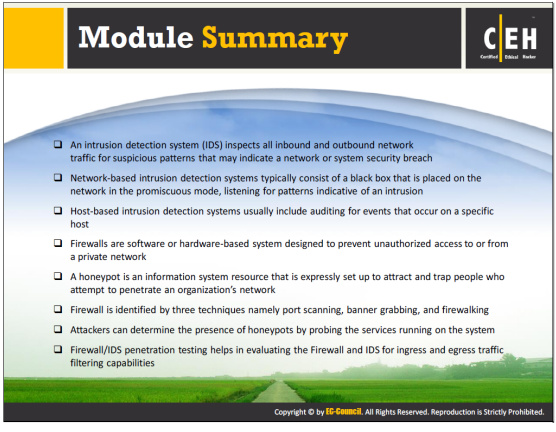












The module discussed how attackers try to evade network security components and various ways to prevent such incidents. The next module discusses **Cloud Computing** and the ways to secure it.