**Intrusion Detection System (IDS)**

An intrusion detection system (IDS) is a security software or hardware device used to monitor, detect, and protect networks or systems from malicious activities; it alerts the concerned security personnel immediately upon detecting intrusions. IDS are extremely useful as they monitor the inbound/outbound traffic of the network and check for suspicious activities continuously to detect a network or system security breach.

An IDS gathers and analyzes information from within a computer or a network to identify possible violations of the security policy, including unauthorized access, as well as misuse. An IDS is also referred to as a “packet sniffer,” which intercepts packets traveling via various communication media and protocols, usually TCP/IP. The packets are analysed after they are captured. An IDS evaluates traffic for suspected intrusions and raises an alarm upon detecting such intrusions.

One of the most common places to deploy an IDS is near the firewall. Depending on the traffic to be monitored, an IDS is placed outside/inside the firewall to monitor suspicious traffic originating from outside/inside the network. When 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.

**How an IDS Detects Intrusion**

* **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. Signature-based intrusion detection compares incoming or outgoing network packets with the binary signatures of known attacks using simple pattern-matching techniques to detect intrusions. The more the signatures, the greater are the chances are of the IDS detecting attacks; however, the traffic may incorrectly match with the signatures, thus impeding system performance. New virus attacks such as URSNIF and VIRLOCK have driven the need for multiple signatures for a single attack. Changing a single bit in some attack strings can invalidate a signature generated for that attack. Therefore, entirely new signatures are required to detect a similar attack.
* **Anomaly detection** involves a database of anomalies. An anomaly is detected when an event occurs outside the tolerance threshold of normal traffic. Therefore, any deviation from regular use is an attack. Anomaly detection detects intrusions based on the fixed behavioural characteristics of the users and components in a computer system. In this type of approach, the inability to construct a model thoroughly on a regular network is a concern. These models should be used to check specific networks.
* **Protocol anomaly detection** depends on the anomalies specific to a protocol. It identifies particular flaws in vendors’ deployment of the TCP/IP protocol. Protocols are designed according to RFC specifications, which dictate standard handshakes to permit universal communication. Malicious anomaly signatures are becoming increasingly common. By contrast, the network protocol is well defined and is changing slowly. Therefore, the signature database should frequently be updated to detect attacks. The best way to present alarms is to explain which part of the state system is compromised.

**Types of Intrusion Detection Systems**

* **Network-based IDS (NIDS)** 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. It generates alerts at the IP or application level based on the content. The NIDS identifies the anomalies at the router and host levels. It audits the information contained in the data packets and logs the information of malicious packets. The threat level enables the security team to remain on alert. These mechanisms typically consist of a black box placed on the network in a promiscuous mode, listening for patterns indicative of an intrusion.
* **Host-based IDS (HIDS)** analyzes each system’s behavior. The HIDS can be installed on any system ranging from a desktop PC to a server. The HIDS focuses on the changing aspects of local systems. It is also more platform-centric, with a greater focus on the Windows OS; nevertheless, other HIDS are available for UNIX platforms. These mechanisms usually include auditing events that occur on a specific host.

**Types of IDS Alerts**

* **True Positive (Attack - Alert):** A true positive is a condition that occurs when an event triggers an alarm and causes the IDS to react as if a real attack is in progress. The event may be an actual attack, in which case an attacker attempts to compromise the network.
* **False Positive (No attack - Alert):** A false positive occurs if an event triggers an alarm when no actual attack is in progress. It occurs when an IDS treats regular system activity as an attack. False positives tend to make users insensitive to alarms and weaken their reactions to actual intrusion events.
* **False Negative (Attack - No Alert):** A false negative is a condition that occurs when an IDS fails to react to an actual attack event.
* **True Negative (No attack - No Alert):** A true negative is a condition that occurs when an IDS identifies an activity as acceptable behavior and the activity is acceptable. A true negative means successfully ignoring acceptable behavior.

**IDS Evasion Techniques**

* **Insertion Attack:** Insertion is the process by 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). An IDS blindly trusts and accepts a packet that an end system rejects. 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 gets confused. An attacker exploits this condition and inserts data into the 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 search 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. A straightforward 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 the information in the packet.

* **Evasion attack:** occurs when the IDS discards packets while the host that has to get the packets accepts them. Using this technique, an attacker exploits the host computer. 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 a 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 is 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.

**Encryption:** Network-based intrusion detection analyzes traffic in the network from the source to the 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 virtual private network (VPN) tunnel, the IDS will not analyze the packets going through these encrypted communications.

**Flooding:** IDS use resources such as memory and processor speed to analyze the traffic going through them. To bypass IDS security, attackers flood IDS resources with noise or fake traffic to exhaust them with having to analyze flooded traffic.

* **Denial-of-Service Attack (DoS):** Attackers monitor and attack the CPU capabilities of the IDS. This is because the IDS needs half of a CPU cycle to read the packets, detect the purpose of their existence, and then compare 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 in carrying out useless work.

The system requires memory to read the packets. The 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 assign all of its memory for meaningless information.

The IDS store activity logs on the disk. The attackers can occupy a significant 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.

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

* **Obfuscating:** Obfuscation means to make code more difficult to understand or read, generally for privacy or security purposes. Obfuscating is an IDS evasion technique used by attackers to encode the attack packet payload in such a way that the destination host can only decode the packet but not the IDS. An attacker manipulates the path referenced in the signature to fool the HIDS. Using Unicode characters, an attacker can encode attack packets that the IDS would not recognize but which an IIS web server can decode. Polymorphic code is another means to circumvent signature-based IDS 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.
* **False Positive Generation:** 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 create a significant amount of alert data that the IDS will log. Attackers construct malicious packets known to trigger alerts within the IDS, forcing it to generate a large number of false reports. Such an attack creates a large amount of log "noise" in an attempt to blend real attacks with fake ones.
* **Session splicing:** It is an IDS evasion technique that exploits how some IDS do not reconstruct sessions before pattern-matching the data. It is a network-level evasion method used to bypass IDS where an attacker splits the attack traffic into an excessive number of packets such that no single packet triggers the IDS. The attacker divides the data in the packets into small portions of a few bytes and evades the string match while delivering the data. The IDS cannot handle an excessive number of small-sized packets and fails to detect the attack signatures. If attackers know what IDS is in use, they could add delays between packets to bypass reassembly checking.

If a packet is not received within a reasonable period, many IDS stop reassembling and handling that stream. If the application under attack keeps a session active for a longer time than that spent by the IDS 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.

* **Unicode Evasion Technique:** Unicode is a character coding system that supports encoding, processing, and displaying of written texts for universal languages to maintain consistency in a computer representation. Several standards, such as Java, LDAP, and XML, require Unicode, and many OS 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, the character “/” can be represented as “%u2215” and “e” as “%u00e9”; in UTF-8, “©” can be represented as “%c2%a9” and “≠” as “%e2%89%a0.”

In the Unicode code space, all the code points are treated differently, but it is possible that there are multiple representations of a single character. There are also code points that alter the previous code points. Moreover, applications or OS may assign the same representation to different code points.

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

Attack-1: The fragment reassembly timeout is 10 s at the IDS and 20 s at the target system, attackers will send the second fragment 15 s after sending the first fragment. In this scenario, the IDS will drop the fragment on receiving the second fragment after its reassembly timeout, but the target host will reassemble the fragments. Attackers will continue sending fragments with intervals of 15 s until the attack payload is reassembled at the target system.

(Just for understanding) Attack-2: 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 s, and the fragmentation reassembly timeout for the host is 30 s. Initially, the attacker sends frag-2 and frag-4 with a false payload referred to 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 four received fragments, the 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 a valid payload, the IDS will have only these two fragments with a valid payload, as the previous fragments will have been reassembled and dropped.

* **Overlapping Fragments:** In this technique, attackers generate a series of tiny fragments with overlapping TCP sequence numbers. For example, the initial fragment consists of 100 bytes of payload with the sequence number of 1, the second fragment includes an overlapping sequence of 96 bytes, and so on. At the time of reassembling the packet, the destination host must know how to assemble the overlapping TCP fragments. Some OS will take the original fragments with a given offset (e.g., Windows W2K/XP/2003) and some OS will take the subsequent fragments with a given offset (e.g., Cisco IOS).
* **Time-To-Live Attack:** 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 the 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 high value such that it can reach its destination under normal circumstances. Different OS use different default initial values for the TTL. Therefore, attackers can guess the number of routers between them and a sending machine, and make assumptions as to what the initial TTL was, thereby guessing which OS a host is running, as a prelude to an attack.
* **Invalid RST Packets:** The TCP uses 16-bit checksums for error checking of the header and data and 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 drops the packet at the receiver's end. The TCP also uses an RST packet to end two-way communications.
* **Urgency Flag:** The urgency flag in the TCP 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, the TCP ignores all data before the urgency pointer, and the data to which the urgency pointer points is processed. If the URG flag is set, the TCP sets the Urgent Pointer field to a 16-bit offset value that points to the last byte of urgent data in the segment. Some IDS do not consider the TCP’s urgency feature and process all the packets in the traffic, whereas the target system processes only the urgent data. When a TCP packet contains both urgent data and normal data then 1-byte data after the urgent data is lost.
* **Polymorphic Shellcode:** Many IDS identify signatures for commonly used strings embedded in the shellcode. Polymorphic shellcode attacks include multiple signatures, making it difficult to detect the signature. Attackers encode the payload using some technique and then place a decoder before the payload. As a result, the shellcode is completely rewritten each time it is sent, thereby evading detection. 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.
* **ASCII Shellcode:** ASCII shellcodes contain only characters from the ASCII standard. Such shellcodes allow attackers to bypass commonly enforced character restrictions within the string input code. They also help attackers bypass IDS pattern matching signatures because they hide strings similarly to polymorphic shellcodes. The IDS pattern matching mechanism does not work efficiently with ASCII values.
* **Application Layer Attack: Media** files such as images, audios, and videos can be compressed so that they can be rapidly transferred as smaller chunks. Attackers find flaws in this compressed data and perform attacks; even the IDS signatures cannot identify the attack code within data thus compressed. Many applications that deal with such media files employ some form of compression to increase the data transfer speed. When you find a flaw in these applications, the entire attack can occur within the compressed data, and the IDS will have no way to check the compressed file format for signatures.
* **Desynchronization**

**Pre-Connection SYN:** This attack is performed by sending an initial SYN before the real connection is established, but with an invalid TCP checksum. The IDS can ignore or accept subsequent SYNs in a connection. If a SYN packet is received after the TCP control block is opened, the IDS resets the appropriate sequence number to match the newly received SYN packet. Attackers send fake SYN packets with a completely invalid sequence number to desynchronize the IDS.

**Post-Connection SYN:** In this technique, attackers attempt to desynchronize the IDS from the actual sequence numbers that the kernel is honouring. Send a post-connection SYN packet in the data stream, which will have divergent sequence numbers but otherwise meet all the necessary criteria to be accepted by the target host. However, the target host will ignore this SYN packet, as it references an already established connection. This attack intends to get the IDS to resynchronize its notion of the sequence numbers to the new SYN packet.

**How to Defend Against IDS Evasion**

* Shut down switch ports associated with known attack hosts.
* Perform an in-depth analysis of ambiguous network traffic for all possible threats.
* Use TCP FIN or Reset (RST) packet to terminate 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 the network topology, nature of network traffic, and number of hosts to monitor.

**Intrusion Detection Tools**

**Snort:** Snort is an open-source network intrusion detection system capable of performing real-time traffic analysis and packet logging on IP networks. It can perform protocol analysis and content searching/matching, and it 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.

**Uses of Snort:**

* Straight packet sniffer such as tcpdump
* Packet logger (useful for network traffic debugging, etc.)
* Network intrusion prevention system

**Snort Rules:** Snort's rule engine allows custom rules to meet the needs of the network. Snort rules  
help in differentiating between normal Internet activities and malicious activities. 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 the promiscuous mode to the network media decodes all the packets passing through the network.

Snort allows users to write their own rules. 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 maintain a hard 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 sufficiently compatible to act immediately and take necessary remedial measures according to the nature of the intrusion.  
Both flexibility and robustness can be achieved using an easy-to-understand and lightweight rule-description language that aids in writing simple Snort rules. Consider the following two primary principles while writing Snort rules:

* No written rule must extend beyond a single line; thus, 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 Classless Inter-Domain Routing (CIDR) block.

**Snort Rules: Rule Actions and IP Protocols**  
The rule header stores a complete set of rules to identify a packet and determines the action to be performed or rule to be applied. It contains information that defines the who, where, and what of a packet, as well as what to do if a packet with all the attributes indicated in the rule should show up. There are five available default actions in Snort: alert, log, pass, activate, and dynamic. Snort supports three IP protocols: TCP, UDP, ICMP

**Snort Rules: Direction Operator and IP Addresses**

* **Direction Operator**This operator indicates the direction of interest for the traffic; traffic can flow either in a single direction or bidirectionally.

Example of a Snort rule using the Bidirectional Operator: log !192.168.1.0/24 any <> 192.168.1.0/24 23

* **IP Addresses**

Identify the IP address and port that the rule applies to

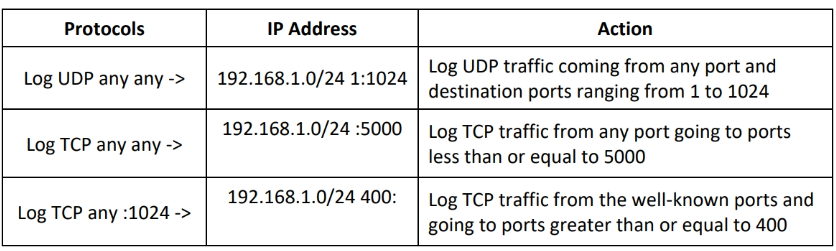
Use keyword "any" to define the IP address

Use numeric IP addresses qualified with a CIDR netmask

Example of IP Address Negation Rule:  
alert tcp !192.168.1.0/24 any -> 192.168.1.0/24 111 (content: "|00 01 86 a5|"; msg: "external mountd access";

Snort Rules: Port Numbers  
Port numbers can be listed in different ways, including the use of "any" ports, static port definitions, port ranges, and by negation. Port ranges are indicated by the range operator ":." 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 by “$<>$”.

Example of port negotiation:

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

**Intrusion Prevention System (IPS)**

Intrusion prevention systems (IPS) are considered as active IDS, as they are capable of not only detecting intrusions but also preventing them. IPS are continuous monitoring systems that often sit behind firewalls as an additional layer of protection.

Some of the actions that an IPS is meant to perform are as follows:

* Generate alerts if any abnormal traffic is detected in the network
* Continuously record real-time logs of network activities 
* Block and filter malicious traffic
* Detect and eliminate threats quickly, as it is placed inline in the operational network
* Identify threats accurately without generating false positives

Classification of IPS:

* Host-based IPS
* Network-based IPS

Advantages of IPS over IDS:

* Unlike IDS, IPS can block as well as drop illegal packets in the network
* IPS can be used to monitor activities occurring in a single organization
* IPS can prevent the occurrence of direct attacks in the network by controlling the amount of network traffic.

**Firewall**

A firewall is a software-or hardware-based system located at the network gateway that protects the resources of a private network from unauthorized access by users on other networks. They are placed at the junction or gateway between two networks, usually a private network and a public network such as the Internet. Firewalls examine all the messages entering or leaving the intranet and block those that do not meet the specified security criteria. Firewalls may be concerned with the type of traffic or with the source or destination addresses and ports.

**Firewall Architecture**

* **Bastion Host:** The bastion host is 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 attacks. Traffic entering or leaving the network passes through the firewall. It has two interfaces:

Public interface directly connected to the Internet

Private interface connected to the intranet

* **Screened Subnet:** A screened subnet (DMZ) is a protected network created with a two-or three-homed firewall behind a screening firewall, and it is a term that is commonly used to refer to the DMZ. When using a three-homed firewall, connect the first interface to the Internet, the second to the DMZ, and the third to the intranet. The DMZ responds to public requests and has no hosts accessed by the private network. Internet users cannot access the private zone. 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 of the three-homed firewall is that if it is compromised, both the DMZ and the intranet could also be compromised.
* Multi-homed Firewall: A multi-homed firewall is a node with multiple NICs that connects to two or more networks. It connects each interface to separate network segments logically and physically. A multi-homed firewall helps in increasing the efficiency and reliability of an IP network. The multi-homed firewall has more than three interfaces that allow for further subdividing the systems based on the specific security objectives of the organization.

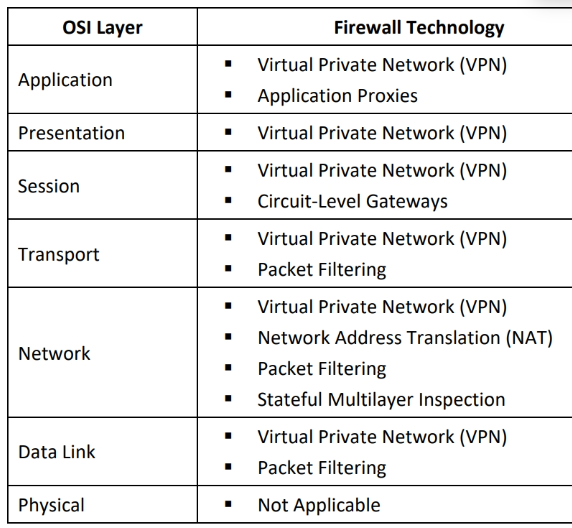
**Demilitarized Zone (DMZ)**

In computer networks, the demilitarized zone (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 an untrusted external network to prevent outsider access to a company’s private data. The DMZ serves as a buffer between the secure internal network and the 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 that are assigned specific roles, such as an internal trusted network, a DMZ network, or an external untrusted network (Internet). Any service such as email, web, or FTP that provides access to external users can be placed in the DMZ.

**Classification of Firewalls**

* **Hardware Firewalls:** A hardware firewall is a dedicated firewall device placed on the perimeter of the network. It is an integral part of the network setup and is also built into broadband routers or used as a standalone product. A hardware firewall helps to protect systems on the local network and performs effectively with little or no configuration. It employs the technique of packet filtering. It reads the header of a packet to find out the source and destination addresses, and compares them with a set of predefined and/or user-created rules that determine whether it should forward or drop the packet. Examples of hardware firewalls include Cisco ASA and FortiGate.
* **Software Firewalls:** A software firewall is similar to a filter. It sits between a regular application and the networking components of the OS. It is more useful for individual home users and it is suitable for mobile users who need digital security when working outside the corporate network. Further, it is easy to install on an individual’s PC, notebook, or workgroup server. It helps protect your system from outside attempts at unauthorized access and provides protection against everyday Trojans and email worms. It includes privacy controls, web filtering, and more. A software firewall implants itself in the critical area of the application/network path. It analyzes the data flow against the rule set. Examples of software firewalls include those produced by Norton, McAfee, and Kaspersky.

**Types of Firewall Technologies**



* **Packet Filtering Firewall:** In a packet filtering firewall, each packet is compared with a set of criteria before it is forwarded. Depending on the packet and the criteria, the firewall can drop the packet and transmit it or send a message to the originator. The rules can include the source and the destination IP address, source and destination port number, and the protocol used. It works at the internet layer of the TCP/IP model or the network layer of the OSI model.

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

* **Source IP address:** Used to check whether the packet is coming from a valid source.
* **Destination IP address:** Checks if the packet is going to the correct destination and if the destination accepts these types of packets.
* **Source TCP/UDP port:** Used to check the source port of the packet
* **Destination TCP/UDP port:** Used to monitor 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 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 the packet is coming from an unreliable zone.
* **Circuit-Level Gateway Firewall:** It forwards data between networks without verification and blocks incoming packets from 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 to network services and host requests. To determine whether a requested session is valid, it checks the TCP handshake between packets. Circuit proxy firewalls allow or prevent data streams; they do not filter individual packets.
* **Application-Level Firewall:** Application-based proxy firewalls focus on the application layer rather than just the packets. Application-level gateways (proxies) can filter packets at the application layer of the OSI model (or the application layer of TCP/IP). Incoming and outgoing traffic is restricted to services supported by the proxy; all other service requests are denied.
* **Stateful Multilayer Inspection Firewall:** This firewall combines the best features of three: packet filtering, circuit-level gateways, and application-level firewalls. They filter packets at the network layer of the OSI model (or the internet layer of the TCP/IP model) to determine whether session packets are legitimate, and they evaluate the contents of the packets at the application layer. This type of firewall can remember the packets that passed through it earlier and make decisions about future packets accordingly. Cisco PIX firewalls are stateful. These firewalls track and log slots or translations.
* **Application Proxy:** An application-level proxy works as a proxy server and filters connections for specific services. It filters connections based on the services and protocols when acting as a proxy. It is a type of server that acts as an interface between the user workstation and the Internet. It correlates with the gateway server and separates the enterprise network from the Internet. It receives a request from a user to provide the Internet service and responds to the original request only. A proxy service is an application or program that helps forward user requests (for example, FTP or Telnet) to the actual services. A proxy is also known as an application-level gateway, as it renews the connections and act as a gateway to the services. Transparency is the main advantage of proxy services.
* **Network address translation (NAT):** It separates IP addresses into two sets and enables the LAN to use these addresses for internal and external traffic. The NAT helps hide an internal network layout and force connections to go through a choke point. It also works with a router, and similarly to packet filtering, it will also modify the packets that the router sends simultaneously. When the internal machine forwards the packet to the external machine, the NAT modifies the source address of the packet to make it appear as if it is coming from a valid address. When the external machine sends the packet to the internal machine, the NAT modifies the destination address to turn the visible address into the correct internal address. The NAT can also change the source and destination port numbers.
* **Virtual Private Network:** A virtual private network (VPN) is a network that provides secure access to the private network through the Internet. VPNs are used for connecting wide area networks (WAN). They allow computers on one network to connect to computers on another network. They are used for the secure transmission of sensitive information over an untrusted network via encapsulation and encryption.

All VPNs that run over the Internet adopt the following principles:

* Encrypts the traffic
* Checks for integrity protection
* Encapsulates new packets, which are sent across the Internet to some destination that reverses the encapsulation
* Checks the integrity
* Decrypts the traffic eventually

**Firewall Limitations**

* Firewalls can restrict users from accessing valuable services such as FTP, Telnet, NIS, etc., and they sometimes restrict Internet access as well.
* The firewall cannot prevent internal attacks (backdoor) in a network, e.g., a disgruntled employee who cooperates with the external attacker.
* The firewall focuses its security at a single point, which makes other systems within the network prone to security attacks.
* A bottleneck could occur if all the connections pass through the firewall.
* The firewall cannot protect the network from social engineering and data-driven attacks whereby the attacker sends malicious links and emails to employees inside the network.

**Firewall Evasion/Bypassing Techniques**

* **Firewall Identification**
* **Port Scanning:** It is used to identify open ports and the services running on these ports. Finding open ports is an attacker’s first step toward gaining access to the target system. To do so, the attacker systematically scans the target’s ports to identify the versions of services, which helps in finding vulnerabilities in these services. 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:** It is a method of collecting information about remote networks behind firewalls. It is a technique that uses TTL values to determine gateway ACL filters and map networks by analyzing the IP packet response. It probes ACLs on packet filtering routers/firewalls using the same method as tracerouting. Firewalking involves sending TCP or UDP packets into the firewall where the 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.
* **Banner Grabbing:** Banners are service announcements provided by services in response to connection requests, and they often carry vendor version information. Banner grabbing is a simple method of fingerprinting that helps in detecting the vendor of a firewall and the firmware version. It identifies the service running on the system. A firewall does not block banner grabbing because the connection between the attacker’s system and the target system appears legitimate.

The syntax is “<service name > <service running > <port number>”, telnet mail.targetcompany.org 25

* **IP Adress Spoofing:** IP address spoofing is a hijacking technique in which an attacker masquerades as a trusted host to conceal his identity, spoof a website, hijack browsers, or gain unauthorized access to a network. In IP spoofing, the attacker creates IP packets by using a forged IP address and gains access to the system or network without authorization. Attackers modify the addressing information in the IP packet header and the source address bits field to bypass the firewall.
* **Source Routing:** Using this technique, the sender of the packet designates the route (partially or entirely) that a packet should take through the network such that the designated route should bypass the firewall node. Source routing is categorized into two approaches: loose source routing and strict source routing. In loose source routing, the sender specifies one or more stages that the packet must go through, whereas in strict source routing, the sender specifies the exact route the packet must go through.
* **Tiny Fragments:** Attackers create tiny fragments of outgoing packets, forcing some of the TCP packet’s header information into the next fragment. The IDS filter rules that specify patterns will not match with the fragmented packets owing to the broken header information. The attack will succeed if the filtering router examines only the first fragment and allows all the other fragments to pass through.
* **Bypass Blocked Sites Using an IP Address in Place of a URL:** This method involves typing a blocked website’s IP address directly in the browser’s address bar instead of the domain name. This method fails if the blocking software tracks the IP address sent to the web server.
* **Bypass Blocked Sites Using Anonymous Website Surfing Sites:** Anonymous web-surfing sites help 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 revealing your IP address.
* **Bypass a Firewall using Proxy Servers:**

**Steps:**

* In the Tools menu of any Internet browser, go to “Proxy Settings,” and in the Internet Properties dialog box under Connections tab, click “LAN settings”
* Under LAN Settings, click on the “Use a proxy server for your LAN” checkbox
* In the Address box, type the IP address of the proxy server
* In the Port box, type the port number that is used by the proxy server for client connections (by default, 8080)
* Click to select the “Bypass proxy server for local addresses”
* **Bypassing Firewalls through the ICMP Tunneling Method:** The ICMP protocol is used to send an error message to the client. As it is a required service for network communication, users often enable this service on their networks. Moreover, it does not entail a significant threat from the security perspective. It allows tunneling of a backdoor shell in the data portion of ICMP Echo packets. RFC 792, which delineates ICMP operation, does not define what should go in the data portion. The payload portion is arbitrary and is not examined by most firewalls. Thus, any data can be inserted in the payload portion of the ICMP packet, including a backdoor application.
* **Bypassing Firewalls through the ACK Tunneling method:** Ordinary packet filtering firewalls define their rule sets based on the SYN packet when TCP level communication is to be established. This is because such a firewall assumes that only the SYN packet is coming from the client and is thus likely to contain malicious code in the SYN packet. These firewalls ignore the possibility that the attacker can also inject malicious code in the ACK packet. As ACK packets are sent after establishing a session, ACK traffic is considered legitimate. In addition, the filtering of ACK packets is ignored to reduce the workload of firewalls, as there can be many ACK packets for one SYN packet. ACK tunneling allows tunneling of a backdoor application with TCP packets with the ACK bit set. The ACK bit acknowledges the receipt of a packet.
* **Bypassing Firewalls through the HTTP Tunneling Method:** HTTP tunneling allows attackers to perform various Internet tasks despite the restrictions imposed by firewalls. This method can be implemented if the target company has a public web server in which port 80 is used for HTTP traffic that is unfiltered by its firewall. The attacker encapsulates data inside HTTP traffic (via port 80).

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 the HTTP protocol. The 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, e.g., local port 80.

* **Bypassing Firewalls through the SSH Tunneling Method: 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 the encrypted SSH protocol using SSH tunneling. Attackers use 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.

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

* **Bypassing Firewalls through the DNS Tunneling Method:** DNS operates using UDP, and it has a 255-byte limit on outbound queries. Moreover, it allows only alphanumeric characters and hyphens. Such small size constraints on external queries allow DNS to be used as an ideal choice to perform data exfiltration by various malicious entities. Since corrupt or malicious data can be secretly embedded into the DNS protocol packets, even DNSSEC cannot detect the abnormality in DNS tunneling.
* **Bypassing Firewall through External Syatems:**

**External Machine can be:**

* A machine that conducts remote administration of the target network
* A machine from the company’s network but located at a different place
* A home machine of and employee

**Steps:**

* Legitimate user works with some external system to access the corporate network
* Attacker sniffs the user traffic and steals the session ID and cookies
* Attacker accesses the corporate network by bypassing the firewall and gets the Windows ID of the running Mozilla process on the user’s system
* Attacker then issues an OpenURL() command to the found window
* User’s web browser is redirected to the attacker’s web server
* The malicious code embedded in the attacker’s web page is downloaded and executed on the user’s machine
* **Bypassing Firewall through MITM Attacks: In** MITM attacks, attackers use 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.

**Steps:**

* Attacker performs DNS server poisoning
* User A requests for www.certifiedhacker.com from the corporate DNS server
* Corporate DNS server sends the IP address (127.22.16.64) of the attacker
* User A accesses the attacker’s malicious server
* Attacker connects to the real host and tunnels the user’s HTTP traffic
* The malicious code embedded in the attacker’s web page is downloaded and executed on the user’s machine
* **Bypassing Firewalls through Content:** In this method, the attacker sends content containing malicious code to the user and tricks him/her into opening it so that the malicious code can be executed. For example, an attacker can send an email containing a malicious executable file or Microsoft office document capable of exploiting a macro bypass exploit. 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 into accessing them.

**Commonly used file formats:** 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)

* **Bypassing the WAF suing an XSS Atack:** XSS attack exploits vulnerabilities that occur while processing the input parameters of end users and the server responses in a web application. Attackers take advantage of these vulnerabilities to inject malicious HTML code into the victim website to bypass the WAF.
* **Using ASCII values to bypass the WAF**

consider the following XSS payload

**<script>alert("XSS")</script>**

When the above JavaScript code is executed, the WAF filters escape single quotes, double magic quotes, etc. Hence, the above payload is filtered by the WAF. To bypass the WAF, we need to convert the above payload into its equivalent ASCII values and then execute it. The JavaScript will automatically convert the ASCII values back into the original characters.

Consider the XSS payload given below:

**XSS Payload:alert("XSS")**

The equivalent ASCII values are:  
**String.fromCharCode(97, 108, 101, 114, 116, 40, 34, 88, 83, 83, 34, 41)**

The above values are inserted into the XSS payload:  
**<script>String.fromCharCode(97, 108, 101, 114, 116, 40, 34, 88, 83, 83, 34, 41)</script>**

The above payload bypasses the WAF filters successfully.

* **Using Hex Encoding to bypass the WAF**In this technique, the entire XSS payload is replaced with Hex values to bypass WAF. Attackers use online websites such as http://www.convertstring.com/EncodeDecode/HexEncode to convert the XSS payload into equivalent Hex values.

consider the following XSS payload

**<script>alert("XSS")</script>**

The encoded value for the XSS payload is  
**%3C%73%63%69%72%70%74%3E%61%6C%65%72%74%28%22%58%53%53%22%29%3C%2F%73%6 3%72%69%70%74%3E**The above payload bypasses the WAF filters successfully.

* **Using Obfuscation to bypass the WAF**  
  Attackers use the obfuscation technique to bypass the WAF. In this technique, attackers use a combination of upper-and lower-case letters in the XSS payload. For example, consider the following XSS payload:

**<script>alert("XSS")</script>**

Using obfuscation, the above payload is replaced with

**<sCRiPt>aLeRT("XSS")</sCriPT>**

**The above payload bypasses the WAF successfully**

**How to Defend Against Firewall Evasion**

* The firewall should be configured such that IP address of an intruder should be filtered out.
* Set the firewall rule set to deny all traffic and enable only the services required.
* If possible, create a unique user ID to run the firewall services instead of running the services using the administrator or root ID.
* Configure a remote syslog server and adopt strict measures to protect it from malicious users.
* Monitor firewall logs at regular intervals and investigate all suspicious log entries found.
* By default, disable all FTP connections to or from the network.
* Catalog and review all inbound and outbound traffic allowed through the firewall.

**Honeypots**

A honeypot is a computer system on the Internet intended to attract and trap those who attempt unauthorized or illicit utilization of the host system to penetrate an organization’s network. It is a fake proxy run to frame attackers by logging traffic through it and then sending complaints to the victims’ ISPs. It has no authorized activity or production value, and any traffic to it is likely a probe, attack, or compromise. Whenever there is any interaction with a honeypot, it is most likely to be malicious. Honeypots are unique; they do not solve a specific problem. Instead, they are a highly flexible tools with many different security applications.

**Types of Honeypots**

Based on design

* **Low-interaction Honeypots:** Low-interaction honeypots emulate only a limited number of services and applications of a target system or network. If the attacker does something that the emulation does not expect, the honeypot will simply generate an error. They capture limited amounts of information, i.e., mainly transactional data, and some limited interactions. These honeypots cannot be compromised completely. KFSensor is a low-interaction honeypot used to attract and identify penetrations. It implements vulnerable system services and Trojans to attract hackers. This honeypot can be used to monitor all TCP, UDP, and ICMP ports and services. KFSensor identifies and raises alerts about port scanning and DoS attacks. A honeytrap is a low-interaction honeypot used to observe attacks against TCP and UDP services. It runs as a daemon and starts server processes dynamically on requested ports. Attackers are tricked into sending responses to the honeytrap server process.
* **Medium-interaction Honeypots:** Medium-interaction honeypots simulate a real OS as well as applications and services of a target network. They provide greater misconception of an OS than low-interaction honeypots. Therefore, it is possible to log and analyze more complex attacks. They can only respond to preconfigured commands; therefore, the risk of intrusion increases. The main disadvantage of medium-interaction honeypots is that the attacker can quickly discover that the system behavior is abnormal. Kojoney2 is a medium-interaction honeypot that emulates a real SSH environment. This honeypot listens on port 21 for incoming SSH connections. If a connection request is initiated, Kojoney2 will verify users against an internal list of fake users. Usually, the connections are accepted by granting access to the SSH shell.
* High-Interaction Honeypots: Unlike their low-and medium-interaction counterparts, high-interaction honeypots do not emulate anything; they run actual vulnerable services or software on production systems with real OS and applications. These honeypots simulate all services and applications of a target network. They can be completely compromised by attackers to gain full access to the system in a controlled area. A honeynet is a prime example of a high-interaction honeypot. It 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.
* **Pure Honeypot:** Pure honeypots emulate the real production network of a target organization. They cause attackers to devote their time and resources toward attacking the critical production system of the company.

Based on deployment strategies

* **Production Honeypots:** Production honeypots are deployed inside the production network of the organization along with other production servers. Although such honeypots improve the overall state of security of the organization, they effectively capture only a limited amount of information related to the adversaries. Such honeypots fall under the low-interaction honeypot category and are extensively employed by large organizations and corporations.
* **Research Honeypots:** Research honeypots are high-interaction honeypots primarily deployed by research institutes, governments, or military organizations to gain detailed knowledge about the actions of intruders. By using such honeypots, security analysts can obtain in-depth information about how an attack is performed, vulnerabilities are exploited, and attack techniques and methods are used by the attackers. The main drawback of research honeypots is that they do not contribute to the direct security of the company.

based on their deception technology

* **Malware Honeypots:** Malware honeypots are used to trap malware campaigns or malware attempts over the network infrastructure. These honeypots are simulated with known vulnerabilities such as outdated APIs, vulnerable SMBv1 protocols, etc., and they also emulate different Trojans, viruses, and backdoors that encourage adversaries to perform exploitation activities.
* **Database Honeypots:** Database honeypots employ fake databases that are vulnerable to perform database-related attacks such as SQL injection and database enumeration. These fake databases trick the attackers by making them think that these databases contain crucial sensitive information such as credit card details of all the customers and employee databases. However, all the information present in the database are fake and simulated.
* **Spam Honeypots:** Spam honeypots specifically target spammers who abuse vulnerable resources such as open mail relays and open proxies. Basically, spam honeypots consist of mail servers that deliberately accept emails from any random source from the Internet.
* **Email Honeypots:** Email honeypots are also called email traps. They are nothing but fake email addresses that are specifically used to attract fake and malicious emails from adversaries. These fake email IDs will be distributed across the open Internet and dark web to lure threat actors into performing various malicious activities to exploit the organization.
* **Spider Honeypots:** Spider honeypots are also called spider traps. These honeypots are specifically designed to trap web crawlers and spiders. Many threat actors perform web crawling and spidering to extract important information from web applications.
* **Honeynets:** Honeynets are networks of honeypots. They are very effective in determining the entire capabilities of the adversaries. Honeynets are mostly deployed in an isolated virtual environment along with a combination of vulnerable servers.

**Detecting and Defeating Honeypots**

A honeypot is a security mechanism that is deployed to counterattack and trap attackers. Honeypots lure attackers into performing malicious activities, and this attack information provides insights into the level and type of threats a network infrastructure can face.

* **Detecting the presence of Layer 7 Tar Pits:** Tar pits are security entities that are similar to honeypots, which are designed to respond slowly to incoming requests. They slow down unauthorized attempts of hackers. Layer 7 tar pits react slowly to incoming SMTP commands by attackers/spammers.
* **Detecting the presence of Layer 4 Tar Pits:** Layer 4 tar pits manipulate the TCP/IP stack and are effectively employed to slow down the spreading of worms, backdoors, etc. In these tar pits, the iptables accept the incoming TCP/IP connection and spontaneously switch to a zero-window size, blocking the attacker from sending further data. This connection cannot be terminated by the attacker, as no data is transferred to the target machine. Layer 4 tar pits such as Labrea can be identified by the attacker by analyzing the TCP window size.
* **Detecting the presence of Layer 2 Tar Pits:** If an attacker launches an attack from the same network, the issue of Layer 2 arises. Layer 2 tar pits are used to block the network penetration of the attacker who gains access to the network as well as to prevent internal threats. The attacker can detect the presence of this daemon by looking at the responses with the unique MAC address 0:0:f:ff:ff:ff, which acts as a kind of black hole.
* **Detecting Honeypots running on VMware:** Owing to its numerous advantages, VMWare is widely used to launch honeypots. Attackers can identify instances that are running on the VMWare virtual machine by analyzing the MAC address. By looking at the IEEE standards for the current range of MAC addresses assigned to VMWare Inc., an attacker can identify the presence of VMWare-based honeypots.
* **Detecting the presence of Honeyd Honeypot:** Honeyd is a widely used honeypot daemon. It is used to create thousands of honeypots easily. It is a network-simulated and service-simulated honeypot deployment engine. This honeyd honeypot can respond to a remote attacker who tries to contact the SMTP service with fake responses.
* **Detecting the presence of User-Mode Linux (UML) Honeypot:** User-Mode Linux is an open-source software under GNU, which is used to create virtual machines and is efficient in deploying honeypots. Attackers can identify the presence of UML honeypots by analyzing files such as /proc/mounts, /proc/interrupts, and /proc/cmdline, which contain UML-specific information.
* **Detecting the presence of Sebek-based Honeypots:** Sebek is a server/client-based honeypot application that captures the rootkits and other malicious malware that hijacks the read() system call. Such honeypots record all the data accessed via reading () call. Attackers can detect the existence of Sebek-based honeypots by analyzing the congestion in the network layer, as Sebek data communication is usually unencrypted.
* **Detecting the presence of Snort\_inline Honeypot:** Snort\_inline is a modified version of Snort IDS that is capable of packet manipulation. It can rewrite rules in the iptables and is mainly used in GenII (2nd generation) honeynets to block known attacks and avoid attacker bouncing. Attackers can identify these honeypots by analyzing the outgoing packets.
* **Detecting the presence of Fake AP:** Fake access points are those that create fake 802.11b beacon frames with randomly generated ESSID and BSSID (MAC address) assignments. Fake access points only send beacon frames but do not produce any fake traffic on the access points, and an attacker can monitor network traffic and quickly note the presence of fake AP.
* **Detecting the presence of Bait and Switch Honeypots:** Bait and switch honeypots actively participate in security mechanisms that are employed to respond quickly to incoming threats and malicious attempts. They redirect all malicious network traffic to a honeypot after any intrusion attempt is detected.