11.1.6 Evade IDS Facts

There are various mechanisms that attackers use to evade an intrusion detection system (IDS). Each of these evasion mechanisms is designed to stop, slow, or otherwise compromise a host or network. As an ethical hacker, you are responsible for both testing systems to discover vulnerabilities and preventing IDS evasion.

This lesson covers the following topics:

- DoS and DDoS attacks
- Insertion attacks
- Obfuscation attacks
- IDS evasion countermeasures

DoS and DDoS Attacks

One of the most familiar attacks is a denial-of-service (DoS) or a distributed denial-of-service (DDoS) attack. In this type of attack, a host or network is compromised or completely brought down due to a flood of malicious traffic bombarding the system. The target can be an IDS or resources required by the IDS.

An IDS sniffs traffic and compares that traffic to rules or acceptable system baselines. This takes a considerable amount of resources to perform. Using scans and other tools, an attacker can identify the resources vital to the proper functioning of the IDS. The attacker can target those resources in a DoS or DDoS attack to make the IDS not function properly or be rendered useless. The following table lists the types of DoS attacks:

Attack	Description
Ping of death	The maximum size of a ping packet is 65,535 bytes. TCP/IP rules do not allow for a ping over this maximum. A classic attack, known as the ping of death, circumvents this rule by fragmenting the packets. When they are reassembled, the packet size is too large, causing a buffer overflow and a system crash. Although not much of a significant threat today due to ping blocking, OS patching, and general awareness, the ping of death was previously formidable and extremely easy to use as a DoS exploit.
TCP fragmentation	TCP fragmentation attacks, also known as teardrop attacks, prevent TCP/IP packets from being reassembled. This is done by setting the flags on all frames to indicate that that they are fragments and providing instructions to connect to a frame that doesn't exist. The target system does not know how to process these packet fragments and crashes or locks up.
SYN flood	A SYN Flood exploits the TCP 3-way handshake. An attacker creates SYN packets with a non-existent source address. When the target machine responds with a SYN-ACK, it goes to the non-existent address, causing the target machine to wait for a response that it will never get. Systems that provide services such as HTTP or Simple Mail Transfer Protocol (SMTP) are particularly vulnerable. Because the source IP address is spoofed in a SYN attack, it is hard for the attacker to be identified.
Smurf attack	The Smurf attack is a DoS attack that targets ICMP protocol weaknesses. The goal of this attack is to flood the target computer with ICMP traffic, making it difficult, if not impossible, to use. To initiate a Smurf attack, the attacker broadcasts ICMP echo request packets that have the spoofed IP address of the target computer. Computers receiving the broadcast request respond, flooding the target system.
Fraggle attack	A Fraggle attack is a DoS attack that targets UDP protocol weaknesses. A large number of UDP packets from a spoofed IP address are broadcast to a network in an attempt to flood the target computer. UDP port 7 is a popular port for this attack because it is the echo port and will generate additional traffic. Even if port 7 is closed, the victim will still be flooded with a large number of ICMP messages. If enough traffic is generated, the network bandwidth will be used up, and communication can come to a halt.
False positives	Another flooding attack is done by generating traffic that triggers a false positive responses from the IDS. The IDS becomes completely overwhelmed and the attacker is able to bypass the IDS.

Insertion Attacks

Another commonly used way that an attacker can evade an IDS is using an insertion attack. This type of attack can be conducted in one of the following ways:

- Insert malicious code into a packet payload. This method is used in:
  - HTTP tunneling because HTTP packets are allowed through the firewall without payload inspection.
  - ICMP tunneling because there is no specification for ICMP payload content.
  - TCP ACK tunneling because some firewalls do not check the contents of ACK packets.
- Modify the attack signature of a packet so that the signature is not recognized by the IDS.
- Change the packet header so that the packet cannot be processed by the IDS.

Obfuscation Attacks

Another method an attacker can use is obfuscation, which is changing the malicious code with the intent to disguise it as legitimate. Because an IDS relies on the ability to identify an attack signature, the process of obfuscating malicious code can be an effective evasion technique. This can be accomplished via manual manipulation of code or through the use of an obfuscator tool.

A common IDS obfuscation evasion method includes encoding attack code using Unicode. Unicode is a coding system used to support interchange, processing, and display of written texts through a network medium. Converting character strings to Unicode can avoid pattern and signature protocol IDS detection. Changing standard code such as HTTP requests and responses into Unicode equivalents can produce code that the target understands and can execute but the IDS does not recognize as a signature.

Encryption is one of the most successful and effective techniques used to bypass an IDS. A common means of obfuscation is done by encrypting the attack on such protocols as HTTPS. Another common method is using nmap to obscure the origin of scanning activities. Nmap has the ability to generate decoys that make the detection of the actual scanning system become much more difficult. The nmap command to generate decoys is **nmap -D RND:10** ***target_IP_address***.

IDS Evasion Countermeasures

There is no way to completely prevent an attacker from evading an IDS, but steps can be taken to reduce the threat to network applications and devices. Using a combination of evasion countermeasures and building defense in depth, an organization can make its network more secure. Identification and detection techniques can help play a part in defending a network against IDS evasions. Although countermeasures may not always prevent the attack, they can help proactively detect attacks early on.

Methods for prevent IDS evasion include:

- Closing ports associated with known attacks and allowing only necessary traffic. Blocking invalid addresses should also be considered as a defense to IDS evasion.
- Practicing effective patch management. Many types of attacks, not just DoS, can be mitigated by effective patch management. Although patch management might not prevent a zero-day attack, it can help in the overall security of the network.
- Implementing acceptable use policies and promoting user network security awareness training.
- Using many types of intrusion detection systems to block incoming packets from untrusted sources.
- Blocking ICMP inbound and outbound traffic at a critical gateway.
- Eliminating single points of failure and adding redundancy or extra bandwidth.
- Analyzing outbound traffic and blocking suspicious traffic to prevent hosts from being compromised.

The following table describes countermeasures.

Countermeasure	Description
Establish traffic baseline	An in-depth network analysis is necessary to establish a baseline for network traffic. In-depth analysis includes recording average packet rates and then flagging any flow deviations. A good practice is to use statistics and the calculations of a cumulative sum to estimate network flow verses actual traffic flow. Using statistics and cumulative sum calculations will reduce the number of false positives.
Maximize bandwidth	Maximizing bandwidth and load balancing are two important countermeasure steps. A system should always have more bandwidth than is expected to be required. In addition to deterring DoS attacks, additional bandwidth accommodates legitimate events that might cause a surge in traffic. Additional bandwidth can also help in absorbing an attack and can buy more time for response.
Use iptables	Programs such as iptables can be used to limit the rate of traffic and can filter on TCP flags and TCP protocols. These tools can control the flow of traffic and block malformed packets or terminate an TCP connection all together.