Dos & DDoS Attack Signatures

(note supplied by Steve Tonkovich of CAPTUS NETWORKS)

Signature based IDS systems use these fingerprints to verify that an attack is taking place. The problem with this method of discovery is that DoS tools are getting smarter, and have the ability to make an attack look like real traffic. Signature based IDS systems need to be updated immediately after a new DoS attack is discovered. This may not happen for 30-60 days, leaving the network vulnerable to these attacks during this time.

This is where the anomaly based detection system used in the CaptIO has a major advantage. All DoS attacks, single source or distributed, have one common goal, to flood the target network and attached resources so that they cannot respond to valid user requests. Most attacks are targeted at backend host systems. Flooding the WAN link, outbound border router, core switching gear, web servers, DB servers, etc. Putting the CaptIO behind the border router can prevent the core switching gear and backend systems from being flooded. In the event that this possible DoS attack is just an increase in real user traffic, our traffic shaping capabilities would reduce the load on the WAN link and border router as well.

More recent attacks, like the one used against Microsoft, targeted the outbound routers. These attacks flood the WAN link and force the router to filter traffic at very high packet rates. Older routers, like the Cisco 7200, cannot handle large DoS attacks forcing them to process this much data. Companies like Exodus, Yahoo, E-Bay, etc. have built up their systems with new Cisco GSR and Juniper border routers that can handle OC192 (9.6 Gbps) speeds. These systems have the resources to process this information and will forward it into the core of the network. Now flooding their core switching gear, servers, etc. If they do not have the CaptIO installed behind their routers, they have no protection to stop DoS attacks from penetrating their network at this level.

Below are some examples of common DoS attacks. These examples are intended to show that no matter what the attacker sends, the CaptIO has the ability to take action against the anomalous traffic. If the traffic does not respond to our traffic shaping requests it is probably a spoofed attack. At that time a packet filter rule will be written and the NOC alerted that this traffic did not respond to our requests to slow down. If the traffic does recede, then the CaptIO would remove the filters and allow traffic to resume. The NOC would receive a log information containing timestamp of when the attack started and then stopped. Allowing them to then analyze web server log files, host based IDS logs, etc. to determine if this traffic was an indeed an attack, network configuration error, or simply a traffic spike.

For examples of the differences between Signature based IDS and the CaptIO product line please see the IDS Categories document.

ICMP Echo Request Flood:
See Smurf. This attack would be handled in the same way as a Smurf attack.

Land Attack:
The Land attack tool is another widely available tool that is used to exploit TCP/IP implementations that are vulnerable to packets crafted in a particular way (a SYN packet in which the source address and port are the same as the destination-that is, spoofed). Any remote user that can send spoofed packets to a host can crash or "hang" that host.

Standards: The best method to reduce the number of IP-spoofed packets exiting onto the Internet would be to install filtering on all routers sending data out to the Internet. These filters would require that all packets leaving a network should have a source address matching that of the internal network. This is performed by the CaptIO which applies network ingress and egress filtering by default. Routers however would perform this filtering of all RFC1918 address space (private space) using performance reducing access control lists.

What we do: The Captus IDS looks at the header of the IP packet to verify that the source and destination addresses are not the same. If it is a spoofed packet we can then take action against that flow of traffic from that spoofed source. The source of the Land Attack would also generate a large amount of traffic triggering our IDS. Alerting the NOC every time the IDS takes an action.

Competition: ISS, NFR, Snort and NetRanger currently do not automatically sense an attack or take any actions against these attacks. They could only alert network management or their own console that an attack is taking place. As they are only passive probes on the network they cannot take action to stop this malicious traffic.

Smurf:
The Smurf attack entails sending ICMP (Internet Control Message Protocol) ping requests to a directed broadcast address. The forged source address of the request is the target of the attack. The recipients of the directed broadcast ping request respond to the request and flood the target's network. This network congestion, however, may not be the only consequence. The Smurf ping packets can contain spoofed IP source addresses such that the responses to the broadcast ping will be directed to the spoofed address. If the broadcast network is large enough it is possible to bring the target system down. This attack can also be used in a distributed manner - sending the spoofed ping to multiple broadcasts, thus causing many networks to attack a single target.

Standards: Currently, the best method to reduce the number of IP-spoofed packets exiting on the Internet would be for all companies to install filtering on their routers that requires packets leaving that network to have a source address from that internal network. The CaptIO applies network ingress and egress filtering by default. Filter all RFC1918 address space (private space) using access control lists.

What we do: The Captus IDS has the ability to take action against protocols or by host systems (source) that create an anomaly or exceed network thresholds. In the event that this is a DoS Smurf attack, the TLIDS TRAP technology would see a flood of ICMP traffic from a single source and take action against that source flow once it exceeded the threshold on that interface, inbound or outbound. If this were a DDoS Smurf attack, the TLIDS TRAP Technology would see an increase in the amount of ICMP traffic coming from many different sources. The CaptOS would then have the ability to take an action against all inbound or outbound ICMP traffic. Applying traffic shaping or a packet filtering rule. Alerting the NOC every time an action is taken.

Competition: : ISS, NFR, Snort and NetRanger currently do not automatically sense an attack or take any actions against these attacks. They could only alert network management or their own console that an attack is taking place. As they are only passive probes on the network they cannot take action to stop this malicious traffic.

SYN flood:
A SYN flood is a request that's sent to a server when establishing a network connection. TCP SYN floods are mostly sent with random source IP addresses. Therefore, when the server replies to a SYN with its SYN ACK, it sends it to a nonexistent system, or one that didn't make the initial request and isn't waiting for it. When the ACK from the client isn't returned, the incomplete connection sits in the queue until it times out. Because ACKs are normally returned in a matter of milliseconds, a connection that takes minutes to expire occupies space in the queue for a relatively long time. Given enough malformed SYNs, the kernel data structures on the destination servers or hosts are used up faster than they can be released, and no additional connections can be made.

Standards: The SYN Cookie method builds sequence numbers calculated from source address, source port, source sequence, destination address, destination port and a secret seed. If an ACK comes from the client, the server can recalculate it to determine if it's a response to the former SYN-ACK. In this way, the server avoids to keeping half-open connections and will not be affected by a SYN flood.

What we do: The CaptOS uses the SYN Cookies method but only if the CaptOS is the target of the incoming / outgoing packets. This would be the case when we were running Network Address Translation. If we are not running in this mode then we would not keep track of the session information (stateful).

The Captus IDS has the ability to take action against protocols or host systems (source) that create an anomaly or exceed network thresholds. In the event that this is a SYN attack, the TLIDS TRAP Technology would see a flood of ICMP traffic from multiple sources and take action against that protocol once it exceeded the threshold on that interface, inbound or outbound. Applying traffic shaping or a packet filtering rule. Alerting the NOC every time an action is taken.

Competition: Today's IDS systems will alert on a SYN attack. Stateful firewalls like Checkpoint and PIX can also protect against these types of or the attacks. However the commonest method that is employed is to Spoof the SYN request before the request reaches the intended host. The protecting device then determines if the SYN is in fact a legitimate function i.e. does the source host complete the full SYN handshake. If the handshake does not complete then the request is malicious, action can be taken and the destination host is protected. If however the SYN is legitimate (as the vast majority are) then the spoofing action must be continued for the full length of the session, it cannot be handed off to the destination host. This can impose a very high overhead on the device doing the spoofing and severely reduce its performance.

TCP Flood:
See Smurf. This attack would be handled in the same way as a Smurf attack.

Teardrop Attack:
The Teardrop attack tool is a widely available tool that exploits implementations of the TCP/IP IP fragmentation re-assembly code that do not properly handle overlapping IP fragments. This allows any remote user to crash a vulnerable machine.

Standards: An operating system patched to the current revision should have no problems handling this situation.

What we do: The CaptOS has the ability to stop frag attacks but only if the CaptOS is the target of the incoming / outgoing packets. This would be the case when we were running Network Address Translation. If we are not running in this mode then we would not stop overlapping fragmented packets but we would not be affected by the Checkpoint Firewall vulnerability of crashing during this situation.

Competition: Today's IDS systems will alert of this frag attack. Stateful firewalls like Checkpoint and PIX can also protect against these types of attacks.

UDP Flood:
See Smurf. This attack would be handled in the same way as a Smurf attack.

Attack Tools

A properly configured CaptIO device would stop most traffic from these attack tools from every leaving the host network. With the TLIDS TRAP Technology enabled all of this malicious traffic would be recognized, shaped and then filtered since they all spoof source IP addresses.

Trinoo: (AKA Trin00)
An early DDoS tool, is relatively unsophisticated by current standards. It initiates only a UDP flood attack. Communication between the master and agents uses unencrypted TCP and UDP. Root/administrator privileges are not needed to use Trinoo. This means that any regular user can deploy a Trinoo constellation without having to compromise a systems administration account. Given Trinoo's relative simplicity, it is easier to detect and combat than more recently developed tools.

Communication between clients, handlers and agents use the following ports (Default ports):
TCP 1524 & 27665
UDP 27444 & 31335
http://staff.washington.edu/dittrich/misc/trinoo.analysis

Tribe Flood Network:
TFN and TFN2K - use multiple attack types, including UDP, ICMP and TCP SYN floods. It can also emulate a Smurf attack. Communication

between the master and the agents uses ICMP_ECHOREPLY packets. Commands and arguments are sent as part of the ICMP ID field and in the data portion of the packets. The main difference between TFN2K and TFN is that the agent is silent in TFN2K, making it more difficult to detect. The master sends multiple commands to the agent and relies on the probability that at least one will get through. In addition, the command packets are mixed with a number of decoy packets sent to random destinations. As TFN evolves, it becomes easier to cause outages and more difficult to detect. TFN and TFN2K are more difficult to deploy than Trinoo, because they require root or administrator privileges on the system running the agent.

Communication between clients, handlers and agents does not use any specific port (it may be supplied on run time or it will be chosen randomly by a program) but is a combination of UDP, ICMP and TCP packets.
http://staff.washington.edu/dittrich/misc/tfn.analysis

Stacheldraht:
This tool has multiple attack options, including UDP, ICMP, TCP SYN and broadcast ping floods. Its use of ICMP_ECHORE PLY is similar to TFN's, but Stacheldraht can encrypt the console-to-master TCP session. Stacheldraht also has an auto-update feature. Like TFN and TFN2K, Stacheldraht requires root or admin privileges on the system running the agent as well as the master.

Communication between clients, handlers and agents use the following ports (Default ports):

TCP 16660 & 65000
ICMP ECHO & ICMP ECHO REPLY

http://staff.washington.edu/dittrich/misc/stacheldraht.analysis