ICSI MODULE 4:

IDS Concepts

4.1 IDS Concepts

There are six basic approaches to intrusion-detection and prevention. Some of these methods are implemented in various software packages, and others are simply strategies that an organisation can employ to decrease the likelihood of a successful intrusion.

Historically, when IDSs were first developed, hubs were used very frequently. Today, switches are used rather than hubs. With a hub, after a packet has travelled from its source network to the destination network (being routed by its destination IP address), it finally arrives at the network segment on which the target is located. After it gets to that final segment, the MAC address is used to find the target. All the computers on that segment can see the packet, but because the destination MAC address does not match the MAC address of their network card, they ignore the packet.

At some point, enterprise individuals realized that if they simply chose not to ignore packets not destined for their network card, they could see all the traffic on the network segment. In other words, one could look at all the packets on that network segment. Thus the packet sniffer was born. After that it was just a matter of time before the idea came about of analysing those packets for indications of an attack, thereby giving rise to intrusion-detection systems.

4.1.1 Pre-emptive Blocking

Pre-emptive blocking seeks to prevent intrusions before they occur. This is done by observing any danger signs of imminent threats and then blocking the user or IP address from which these signs originate. Examples of this technique include attempts to detect the early Footprinting stages of an imminent intrusion, then blocking the IP or user that is the source of the Footprinting activity. If you find that a particular IP address is the source of frequent port scans and other scans of your system, then you would block that IP address at the firewall.

This sort of intrusion detection and avoidance can be quite complicated, and there is the potential of blocking a legitimate user by mistake. The complexity arises from distinguishing legitimate traffic from that indicative of an impending attack. This can lead to the problem of false positives, in which the system mistakenly identifies legitimate traffic as some form of attack.

Usually, a software system will simply alert the administrator that suspicious activity has taken place. A human administrator will then make the decision whether or not to block the traffic. If the software automatically blocks any addresses it deems suspicious, you run the risk of blocking out legitimate users. It should also be noted that nothing prevents the offending user from moving to a different machine to continue the attack. This sort of approach should only be one part of an overall intrusion-detection strategy and not the entire strategy.

4.1.2 Anomaly Detection

Anomaly detection involves actual software that works to detect intrusion attempts and notify the administrator. This is what many people think of when they talk about intrusion-detection systems. The general process is simple: The system looks for any abnormal behaviour. Any activity that does not match the pattern of normal user access is noted and logged. The software compares observed activity against expected normal usage profiles. Profiles are usually developed for specific users, groups of users, or applications. Any activity that does not match the definition of normal behaviour is considered an anomaly and is logged. Sometimes we refer to this as “trace back” detection or process. We are able to establish from where this packet was delivered. The specific ways in which an anomaly is detected include:

- Threshold monitoring
- Resource profiling
- User/group work profiling
- Executable profiling

4.1.2.1 Threshold Monitoring

Threshold monitoring pre-sets acceptable behaviour levels and observes whether these levels are exceeded. This could include something as simple as a finite number of failed login attempts or something as complex as monitoring the time a user is connected and the amount of data that user downloads. Thresholds provide a definition of acceptable behaviour. Unfortunately, characterizing intrusive behaviour only by the threshold limits can be somewhat challenging. It is often quite difficult to establish proper threshold values or the proper time frames at which to check those threshold values. This can result in a high rate of false positives in which the system misidentifies normal usage as a probable attack.

4.1.2.2 Resource Profiling

Resource profiling measures system-wide use of resources and develops a historic usage profile. Looking at how a user normally utilizes system resources enables the system to identify usage levels that are outside normal parameters. Such abnormal readings can be indicative of illicit activity underway. However, it may be difficult to interpret the meaning of changes in overall system usage. An increase in usage might simply indicate something benign like increased workflow rather than an attempt to breach security.

4.1.2.3 User/Group Work Profiling

In user/group work profiling, the IDS maintains individual work profiles about users and groups. These users and groups are expected to obey to these profiles. As the user changes his activities, his expected work profile is updated to reflect those changes. Some systems attempt to monitor the interaction of short-term versus long-term profiles. The short-term profiles capture recent

changing work patterns, whereas the long-term profiles provide a view of usage over an extended period of time. However, it can be difficult to profile an irregular or dynamic user base. Profiles that are defined too broadly enable any activity to pass review, whereas profiles that are defined too narrowly may inhibit user work.

4.1.2.4 Executable Profiling

Executable profiling seeks to measure and monitor how programs use system resources with particular attention to those whose activity cannot always be traced to a specific originating user. For example, system services usually cannot be traced to a specific user launching them. Viruses, Trojan horses, worms, trapdoors, and other software attacks are addressed by profiling how system objects such as files and printers are normally used not only by users, but also by other system subjects on the part of users. In most conventional systems, for example, any program, including a virus, inherits all of the privileges of the user executing the software. The software is not limited by the principle of least privilege to only those privileges needed to properly execute. This openness in the architecture permits viruses to covertly change and infect totally unrelated parts of the system.

Executable profiling enables the IDS to identify activity that might indicate an attack. Once a potential danger is identified, the method of notifying the administrator, such as by network message or e-mail, is specific to the individual IDS.

Components and Processes of IDS

4.2 Components and Processes of IDS

Regardless of what IDS you select, they all have certain components in common. It is important to have a general understanding of these components.

The following terms will familiarize you with basic components and functions in all IDSs:

- An activity is an element of a data source that is of interest to the operator.
- The administrator is the person responsible for organisational security.
- A sensor is the IDS component that collects data and passes it to the analyser for analysis.
- The analyser is the component or process that analyses the data collected by the sensor.
- An alert is a message from the analyser indicating that an event of interest has occurred.
- The manager is the part of the IDS used to manage, for example a console.
- Notification is the process or method by which the IDS manager makes the operator aware of an alert.
- The operator is the person primarily responsible for the IDS. This is often the administrator.
- An event is an occurrence that indicates a suspicious activity may have occurred.
- The data source is the raw information that the IDS uses to detect suspicious activity.

Beyond these basic components, IDSs can be classified either based on how they respond to detected anomalies or based on how they are deployed. An active IDS, now called an IPS (Intrusion Prevention System), will stop any traffic deemed to be malicious. A passive IDS simply logs the activity and perhaps alerts an administrator. The problem with IPS/active IDS is the possibility of false positives. It is possible to have activity that appears to be an attack, but really is not. You can also define IDS/IPS based on whether a single machine is monitored or an entire network segment is monitored. If it is a single machine, then it is called a HIDS (host-based intrusion-detection system) or HIPS (host-based intrusion prevention system). If it is a network segment then it is called a NIDS (network-based intrusion-detection system) or NIPS (network-based intrusion prevention system).

Implementing IDS

4.3 Implementing IDS

Many vendors supply IDSs, and each of these systems has its own strengths and weaknesses. Deciding which system is best for a particular environment depends on many factors, including the network environment, security level required, budget constraints, and the skill level of the person who will be working directly with the IDS.

4.3.1 Snort

Snort is perhaps the most well-known open source IDS available. It is a software implementation installed on a server to monitor incoming traffic. It typically works with a host-based firewall in a system in which both the firewall software and Snort run on the same machine. Snort is available for UNIX, Linux, Free BSD, and Windows. The software is free to download, and documentation is available at the website: www.snort.org. Snort works in one of three modes: **sniffer, packet logger, and network intrusion-detection**.

4.3.1.1 Sniffer

In packet sniffer mode, the console (shell or command prompt) displays a continuous stream of the contents of all packets coming across that machine. This can be a very useful tool for a network administrator. Finding out what traffic is traversing a network can be the best way to determine where potential problems lie. It is also a good way to check whether transmissions are encrypted.

4.3.1.2 Packet Logger

Packet logger mode is similar to sniffer mode. The difference is that the packet contents are written to a text file log rather than displayed in the console. This can be more useful for administrators who are scanning a large number of packets for specific items. Once the data is in a text file, users can scan for specific information using a word processor's search capability.

4.3.1.3 Network Intrusion-Detection

In network intrusion-detection mode, Snort uses a heuristic approach to detecting anomalous traffic. This means it is rule-based and it learns from experience. A set of rules initially governs a process. Over time, Snort combines what it finds with the settings to optimize performance. It then logs that traffic and can alert the network administrator. This mode requires the most configuration because the user can determine the rules that wishes to implement for the scanning of packets. Snort works primarily from the command line (Shell in Unix/Linux, command prompt in Windows).

Configuring Snort is mostly a matter of knowing the correct commands to enter and understanding their output. Anyone with even moderate experience with either Linux shell commands or DOS commands can quickly master the Snort configuration commands. Snort is a good tool when used in conjunction with host-based firewalls or as an IDS on each server to provide additional security.

4.3.2 Cisco Intrusion Detection and Prevention

The Cisco brand is widely recognised and well respected in the networking profession. Along with their firewalls and routers, Cisco has several models of intrusion detection, each with a different focus/purpose. In the past, Cisco had two specific, widely used IDS products, the Cisco IDS 4200 Series Sensors and Cisco Catalyst 6500 Series Intrusion-Detection System (IDSM-2) Services Module. There are a number of products in this group, notably the Firepower 4100 series, the Firepower 8000 series, and the Firepower 9000 series. All the products include malware protection as well as sandboxing. These Cisco products also integrate cyber threat intelligence features.

The 4100 series is meant for small networks and the 9000 series is designed for large scale networks. One of the chief benefits of using Cisco security products is their widespread use across the industry and the availability of good training. The fact that so many organisations use Cisco indicates a high level of successful field testing, which generally indicates a reliable product. Cisco also sponsors a range of certifications on its products, making it easier to determine whether someone is qualified on a particular Cisco product.

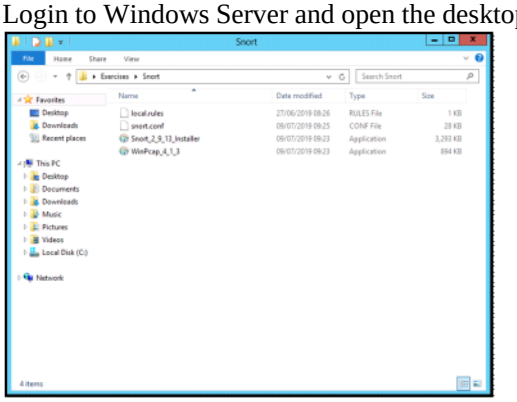
Guided Exercise: Implementing an IDS

4.4 Guided Exercise: Implementing an IDS

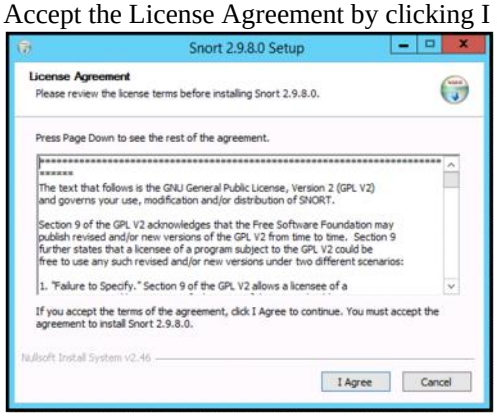
Resources	
Files	None
Machines	Windows Server, Ubuntu Server

In this exercise you are required to install Snort on Windows Server and capture data for analysis.

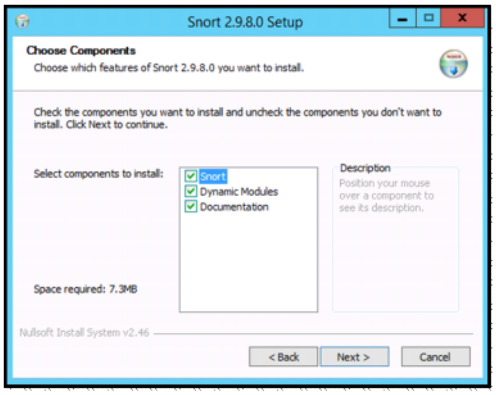
Login to Windows Server and open the desktop folder Exercises -> Snort. Double click the Snort Installer file to install it.



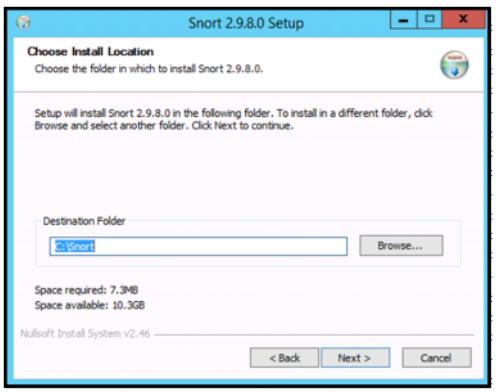
Accept the License Agreement by clicking I Agree.



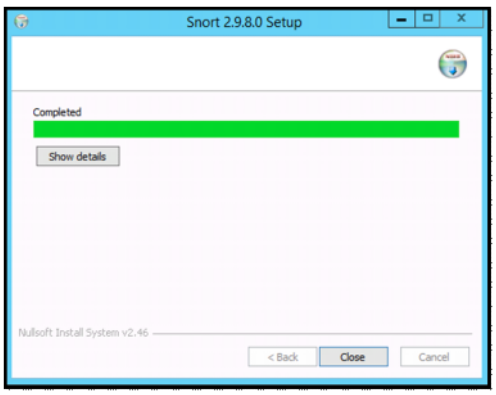
Click Next on the Choose Components window.



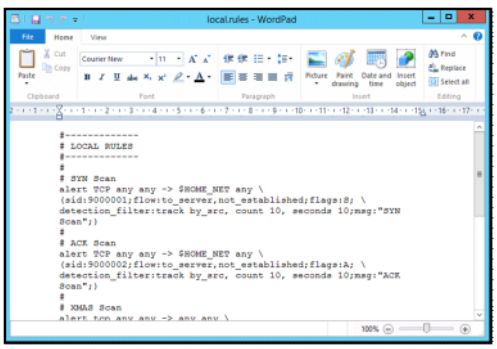
Click Next on the Choose Install Location.



Click Close once the installation finishes and then OK on the Snort Setup.



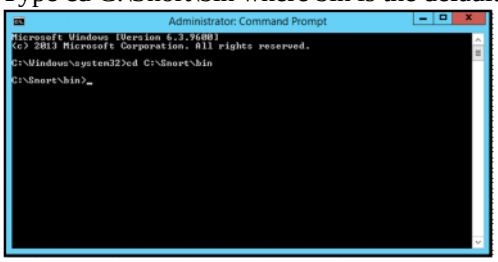
Copy the file snort.conf from the Desktop folder Exercises -> Snort to C:\Snort\etc and overwrite the file that is already there. Copy the file local.rules from the Desktop folder Exercises -> Snort to C:\Snort\rules.
Open the file local.rules using WordPad. Under the LOCAL RULES section there are different rules having a header and a body. The first rule detects a SYN scan and the second rule detects an ACK scan.



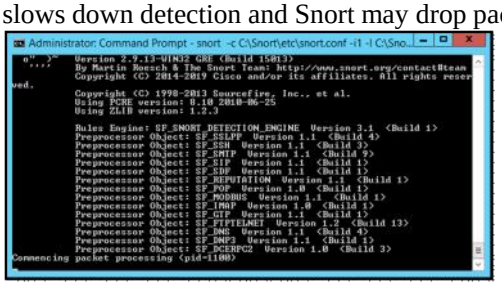
On the folder Exercises -> Snort double click the file WinPcap to install it. Click Next on the WinPcap Setup window and then click I Agree. Click Install on the next window and leave the check mark on Automatically start the WinPcap driver at boot time.

Once the installation finishes click on Finish.

Open a command prompt by right clicking the Start button and select Command Prompt (Admin).
Type cd C:\Snort\bin where bin is the default directory where the snort executable resides.

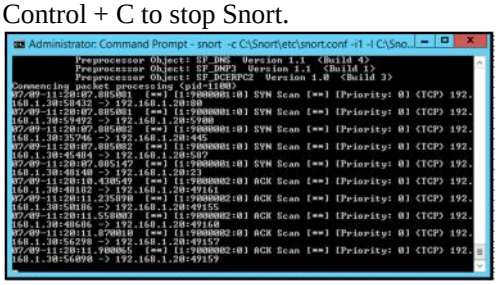


Type the following command “snort –c C:\Snort\etc\snort.conf –i1 –l C:\Snort\log –A console” and press enter. The option –c tells Snort to find the configuration file. The option –i1 tells Snort to capture on interface 1. The –l option tells Snort to log alerts and where to save them. The –A console option tells Snort to send alerts also to the console. This option is normally not used because it slows down detection and Snort may drop packets.

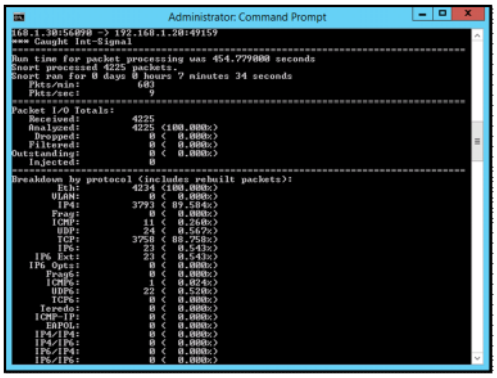


Login to Ubuntu Server and run the comannd nmap –A 192.168.1.20. Allow the scan to complete and then check the Snort command prompt on Windows Server.

Switch to the Windows Server and on the Snort command prompt you should see 5 SYN scan alerts and 5 ACK scan allerts. Press Control + C to stop Snort.



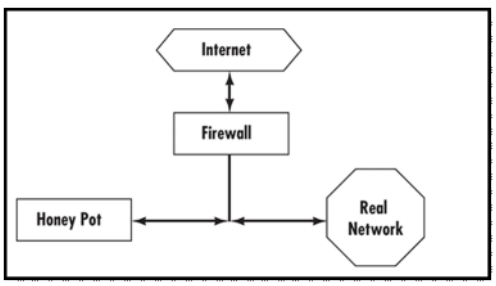
Once you stop Snort a list with different statistics will be revealed.



Honeypots
4.5 Honeypots

A honeypot is a single machine set up to simulate a valuable server or even an entire subnetwork. The idea is to make the honeypot so attractive that if a hacker breaches the network's security, to be attracted to the honeypot rather than to the real system. Software can closely monitor everything that happens on that system, enabling tracking and perhaps identification of the intruder.

The underlying premise of the honeypot is that any traffic to the honeypot machine to be considered suspicious. Because the honeypot is not a real machine, no legitimate users should have a reason to connect to it. Therefore, anyone attempting to connect to that machine can be considered a possible intruder. The honeypot system can entice him to stay connected long enough to trace where is connecting from. Figure 5-3 illustrates the honeypot concept.



4.5.1 Specter

Specter is a software honeypot solution. Complete product information is available at www.specter.com. The Specter honeypot is comprised of a dedicated PC with the Specter software running on it. The Specter software can emulate the major Internet protocols/services such as HTTP, FTP, POP3, SMTP, and others, thus appearing to be a fully functioning server. The software was designed to run on Windows 2000 or XP but will execute on later versions of Windows, but it can simulate AIX, Solaris, UNIX, Linux, Mac, and Mac OS X.

Specter works by appearing to run a number of services common to network servers. In fact, in addition to simulating multiple operating systems, it can also simulate the following services:

- SMTP
- FTP
- TELNET
- FINGER
- POP3
- IMAP4
- HTTP
- SSH
- DNS
- SUN-RPC

Even though Specter appears to be running these servers, it is actually monitoring all incoming traffic. Because it is not a real server for your network, no legitimate user should be connecting to it. Specter logs all traffic to the server for analysis. Users can set it up in one of five modes:

- Open: In this mode, the system behaves like a badly configured server in terms of security. The downside of this mode is that you are most likely to attract and catch the least skilful hackers.
- Secure: This mode has the system behaving like a secure server.
- Failing: This mode is interesting in that it causes the system to behave like a server with various hardware and software problems. This might attract some hackers because such a system is likely to be vulnerable.

- Strange: In this mode, the system behaves in unpredictable ways. This sort of behaviour is likely to attract the attention of a more talented hacker and perhaps cause him to stay online longer trying to figure out what is going on. The longer the hacker stays connected, the better the chance of tracing him.
- Aggressive: This mode causes the system to actively try to trace back the intruder and derive his identity. This mode is most useful for catching the intruder.

In all modes, Specter logs the activity, including all information it can derive from the incoming packets. It also attempts to leave traces on the attacker's machine, which can provide clear evidence for any criminal action. Users can also configure a fake password file in all modes. These are particularly useful because most hackers attempt to access a password file to crack the passwords. If they are successful, they can then log on as a legitimate user. The holy grail of hacking is getting the administrator's password. There are multiple ways to configure this fake password file:

- **Easy:** In this mode the passwords are easy to crack, leading an intruder to believe that she has actually found legitimate passwords and usernames. Often a hacker with a legitimate logon will be less careful covering her tracks. If you know that logon is fake and the system is set up to monitor it, you can track it back to the hacker.
- **Normal:** This mode has slightly more difficult passwords than the easy mode.
- **Hard:** This mode has even harder passwords to crack. There is even a tougher version of this mode called mean, in which the passwords are very difficult to break so that the hacker can be traced while he is taking time to crack the passwords.
- **Fun:** This mode uses famous names as usernames.
- **Warning:** In this mode the hacker gets a warning telling him he has been detected if he is able to crack the password file. The theory behind this mode is that most hackers are simply trying to see if they can crack a system and do not have a specific objective. Letting this sort of hacker know he has been detected is often enough to scare him off.

4.5.2 Symantec Decoy Server

Because Symantec is such a prominent vendor for both antivirus software and firewall solutions, it should come as no surprise that it also has a honeypot solution. The first Symantec honeypot product was Decoy Server. It simulated a real server by simulating many server functions, such as incoming and outgoing e-mail traffic.

As the Decoy Server works as a honeypot, it also works as an IDS monitoring the network for signs of intrusion. If an attack is detected, all traffic related to that attack is recorded for use later in whatever investigative, criminal, or civil procedures that may arise.

Decoy Server is designed to be part of a suite of enterprise security solutions that work together, including enterprise versions of Symantec's antivirus software, firewall software, and antispyware.

QUIZ:

- 1.A profiling technique that monitors how applications use resources is called?
- 2.Specter is an advanced IDS system
- 3.Attempting to attract intruders to a system setup for monitoring them is called?
- 4.A system that is setup for attracting and monitoring intruders is called?
- 5.IDS is an acronym for:
- 6.A series of ICMP packets sent to your ports in sequence might indicate what?
- 7.Specter aggressive mode tries to trace the attacker and gain its identity
- 8.Which of the following is NOT a profiling strategy used in anomaly detection?
- 9.What type of IDS is Snort?
- 10.What is another term for pre emptive blocking?