Global Information Assurance Certification Paper

The Second Step to IDS: Interpretation

GIAC Security Essentials Certification

Practical Assignment

Version 1.4c

Option 1

Cory Schooley

January 3, 2005
Class Washington D.C., July 25, 2004-August 1, 2004

Table of Contents

INTRODUCTION

As an employee for a computer consulting company, we specialize in all areas of computer and network assistance. However, network security is our forté. For our clients, the biggest concern is keeping intruders out of their networks. One way to prevent intruders out is by keeping their network secure. We build Intrusion Detection Systems (IDS) and place them on client networks to aid in securing their network. With the IDS machine, we are able to monitor any intrusions or compromises made on a network. We use software installed on a Linux operating system to detect the intrusions on our clients' networks upon entering the system. The results from the software are logged in a report. This report gives us the information we need to assess the damage on the network. Using this tool, we can determine what action to take to secure the network. In the following pages, I will explain what an IDS is and the different types of IDS. I will also explain the software used on our IDS machines, how we use the information the software gives us and why there is such an important need for IDS machines to be implemented on a company's network.

WHAT IS AN IDS?

An Intrusion Detection System (IDS) is a machine which sits behind a firewall on the network to sniff out network packets and detect any unusual activity coming in or going out of a company's network. It also monitors what the firewall does not block out. An IDS can identify internal and external attacks to a network, as well as any authorized or unauthorized access attempts being made from the firewall.[1] Attacks can leave pieces of evidence behind like a "fingerprint" the IDS system would be able to detect in order to identify a network or system attack.[2] IDS can also stand for Intrusion Detection Software to identify the software loaded on the machine doing the detecting of the network intrusions.

There are two fundamental techniques of an IDS: misuse detection and anomaly detection. Misuse detection relies on a predefined set of attack signatures set by the network administrator or the software manufacturer. The IDS tries to match every incoming network packet received to the signatures of a known attack. Anomaly detection has a set of instructions periodically monitoring segments of the network which compare the segments' state to the normal state of the network. The signatures look at raffic load, protocol breakdown, and typical packet size.[3]

An IPS machine is different from an IDS machine. IPS stands for Intrusion Prevention System. These machines have software installed on them and are proactive on preventing an intruder from entering the network by completely

---

[1] "Threat Management: The Future of Network Security" white paper by Demarc Security, Inc., 2004

[2] Hontanon, Ramon J., "Emerging Technology: Deploying an Effective Intrusion Detection System", http://www.networkmagazine.com/article/NMG20000830S0003, September 5, 2000.

[3] Hontanon, Ramon J., "Emerging Technology: Deploying an Effective Intrusion Detection System", http://www.networkmagazine.com/article/NMG20000830S0003, September 5, 2000.

blocking their access.[4] Some firewalls act as an IPS machine with the software installed on the hardware by the manufacturer. Companies who make IPS machines include Vigilant Network Security and the well-known anti-virus manufacturer McAfee.

TYPES OF IDS AND THEIR PRIMARY FUNCTIONS

Network-Based Intrusion Detection Systems (NIDS)

A Network-based Intrusion Detection System, or NIDS, is an IDS placed beside the network in front of a firewall off-line, watches the live network traffic as it goes by, and collects all that data. It also monitors packets coming in to the network. It operates on a stand-alone computer. The NIDS checks for anomalies that may indicate an attack.[5]

There are many pros and cons to having a NIDS on a network. The pros to having a NIDS are its ability to see network traffic across an entire segment of a network, it can monitor many nodes and report on possible threats with a single sensor, and it's easy to implement. Since it is an off-line machine, it cannot cause any network interruptions. It can also be configured in stealth mode so it doesn't place any load on the network.[6]

Unfortunately, there are many cons to having a NIDS on a network as well. The NIDS are not always placed on a network where it should be to detect all the intrusion attempts. Attacks occurring within the host that are invisible to the network can be missed. NIDS are more expensive and more difficult to deploy than the host-based Intrusion Detection System. The speed and amount of traffic a NIDS has to monitor can exceed the bandwidth limit of its sensor and completely exhaust it. Because of this, an attack could be made by flooding the network. Since there is so much data reported, having an employee or manager take the time to go through all of it is extremely time consuming and can slow down the response time if an attack occurs.[7]

Host-Based Intrusion Detection Systems (HIDS)

A Host-based Intrusion Detection System, or HIDS, checks the system logs on servers for evidence of any compromises on a network in real time. The software on a HIDS usually resides on one machine monitoring traffic on the host machine to see if a security breach has been made.

---

[4] "Threat Management: The Future of Network Security" white paper by Demarc Security, Inc., 2004

[5] "Beyond IDS: Essentials of Network Intrusion Prevention" white paper by Top Layer Networks, November 2002. http://www.intrusion-detection-system-group.co.uk/

[6] SANS Institute Track 1-SANS Security Essentials, Volume 1.3, "Internet Security Technologies", page 332. "Beyond IDS: Essentials of Network Intrusion Prevention" white paper by Top Layer Networks, November 2002.

[7] SANS Institute Track 1-SANS Security Essentials, Volume 1.3, "Internet Security Technologies", page 333. "Beyond IDS: Essentials of Network Intrusion Prevention" white paper by Top Layer Networks, November 2002.

One of the pros to having an HIDS on a network is its ability to detect backdoor attempts coming into a network from a modem or a private link to the network. Also, a dedicated machine or other hardware is not required to run, and they cost much less than a NIDS to deploy.

One of the cons to having an HIDS on a network is it takes much more effort to deploy. Each machine needs to have the software installed on it and when an update needs to be made, each machine must be touched. Also, the more the host machine is reconfigured, the better the chance for false positives to be generated. However, both the NIDS and the HIDS machines can produce false positives. This can lead to important information and alerts to be ignored by whoever is reviewing the logs.

A company's network should have both a NIDS machine and a HIDS machine implemented. It will ensure the network has proper coverage if an attack should take place. For example, if an attacker did a port scan on a network, a NIDS machine will not be able to detect it, but an HIDS will detect the port scan..[8]

In-Line Intrusion Detection System

An In-Line Intrusion Detection System sits in front of a firewall, usually placed behind a router on a company's network. It has the same rule set a normal IDS machine plus an additional set of rules written to drop network packets automatically in real-time coming into the network. An aggressive or proactive IDS watches the network traffic coming into or leaving a network, where as a standard IDS machine is passive on how it watches IP packets pass through from within the network. If any of the packets are actually modified on an In-Line IDS machine, the machine then is classified as an IPS machine.[9]

Companies that Produce IDS Software and Assessment Tools

There are many different types of software produced for IDS machines. A network's size and needs will determine which software will be the best fit for a company. Even a company's budget can be the determining factor on which software to deploy. There are many great freeware software packages. One of the freeware software packages is Bro by Vern Paxson. Bro is built around an event engine which pieces network packets into events that report different types of activity. Events coming through the event engine are run through a policy script either written by an administrator or by the Bro software. [10]

Shoki by Sourceforge is a software package intended for a NIDS machine. Its functionalities include signature matching to filter expressions, multi-filter rule sets matching individual packets or ordered series of packets, threshold based

[8] SANS Institute Track 1-SANS Security Essentials, Volume 1.3, "Internet Security Technologies", page 334.
[9] Graeme Connell, "IDS: Re: definition for Inline IDS/IPS", September 27, 2004, Insecure.org forum.
[10] http.//www.securitywizardry.com/N_ids.htm

logging, fragment reassembly and remote OS identification via passive fingerprinting.[11]

There are also some very high end commercial software packages available for purchase. One product is by Cisco Sytems, Inc. called Cisco Secure IDS, formerly NetRanger, which is a NIDS misuse detection software. It is intended for large, enterprise-scale networks and performs in real-time. It is designed to detect, report and terminate unauthorized activity throughout a network, as well as look for patterns of misuse. [12]

Cyclops by e-Cop.net Pte Ltd is a Snort-based IDS software providing flexible intrusion detection at Gigabit speeds. It is also a high-speed packet analysis that detects suspicious activity in real-time and takes action before a network can be attacked.[13]

Sourcefire Intrusion Management System (IMS) by Sourcefire, Inc. was founded by the creators of Snort. It uses Snort technology with added interfaces, optimized hardware, data analysis, and policy management. Sourcefire Network Sensors come with IMS can monitor all networks speeds including beyond Gigabit speeds.[14]

WHY IT IS IMPORTANT FOR COMPANIES TO HAVE AN IDS ON THEIR NETWORK

Firewall Vulnerabilities

In the world we live in today, everything is highly dependent on computers. Whether it is a cell phone, an Internet connection at home, a debit card from a bank or everyday business practices, they all are dependent on computer networks. Hopefully, if a network administrator has done the proper setup of their company's network, a firewall has been implemented onto the company's network right behind their Internet connection. The firewall is an essential part of the network as it is the first line of blocking unauthorized access into the network. Certain ports can be opened or closed with restrictions depending on the action that needs to take place on the firewall to allow or deny access into the network. When ports are opened, an internal IP address can be specified so only the computer with the internal IP address can be authorized to gain access through a specific port. Services can also be specified on a firewall to gain authorized access to the network such as SSH (Secure Shell), SMTP, HTTP and HTTPS.

With all the different ways to allow or deny access onto a network from the firewall level, this should be enough to protect a network, right? Wrong. Having the firewall as a means of keeping intruders out of a network is simply not enough. For example, firewalls cannot prevent intruders from attacking a

[11] http.//www.securitywizardry.com/N_ids.htm
[12] http.//www.securitywizardry.com/N_ids.htm
[13] http.//www.securitywizardry.com/N_ids.htm
[14] http.//www.securitywizardry.com/N_ids.htm

network if their weapons of choice are DoS (Denial of Service) or hybrid attacks.[15] Not only can these types of attacks get through a firewall, the sheer volume of attacks can overwhelm the firewall as well. Having an IDS system beside a firewall can take up some of the volume of traffic.[16] Since some ports need to be open on a firewall such as port 80 (HTTP) for a web server, for example, an attacker can take advantage of an open port. Once the server has been compromised, an intruder can gain access to various areas on the network, which is known as an application-layer attack.[17] Other ports open on a firewall for SMTP, HTTP and FTP can allow attackers in to a network if they are able to gain an access account or crack weak passwords. Unfortunately, not all companies have a policy on password management. End users can leave their passwords the same as the day they started at a company, which is usually just the word "password" or something very simple such as their own user name or a pet's name. Outside attackers can get through these types of passwords very easily with minimal effort. If a firewall is not configured properly, a door to the network is wide open for intrusion.

Let us not also forget the other common attack a firewall cannot catch or block: an insider attack. This particular attack comes from inside a company by the employees, or even former employees, who have network information to get in to the network. They know how the network is setup and, in some cases, know how the network is configured, complete with logins and passwords. If someone is already in the network, authenticating with a login and password, then they have simply bypassed the firewall's protection. There is no way for a firewall to detect any foul play on a network if someone has properly authenticated themselves. To the firewall, it is normal activity. All of the company's assets, financial information and trade secrets are fair game to an insider. With wonderful advancements in technology, companies can be located all over the world and still communicate as if they are around the corner from each other. Insiders can use this global network of communication maliciously if they want to take advantage of a company. While having a firewall on the network is vital, it cannot protect the network alone. An IDS machine needs to be implemented in order for the network to have proper protection.

Malware

Malware is any kind of software written with the intent of doing damage internally to a network or computer system without the company or end user knowing about it.[18] The software is installed and ran usually without the end

[15] "Beyond IDS: Essentials of Network Intrusion Prevention" white paper by Top Layer Networks, November 2002.
[16] SANS Institute Track 1-SANS Security Essentials, Volume 1.3, "Internet Security Technologies", page 300.
[17] "Economic Impact of Network Security Threats" white paper by Cisco Systems, Inc., 1992-2001.
[18] SANS Institute Track 1-SANS Security Essentials, Volume 1.4, "Secure Communications", page 221.

user even knowing it has been done. Malicious software is written to destroy a company's data information, shut down a computer or network system, or develop code to gain access to a company's information and send it outside of the company. The software can simply be placed on media and ran on a network; however, the most common way malicious software finds its way into a company's network is through the Internet. Malware is installed on a computer when an end user browses to a web site that contains the information. It can also be installed when an end user receives an email that may have embedded software in the message. Usually, the message comes from an unknown source. Malware can also be found in an attachment to an email message and is activated when the end user opens the attachment.

The most commonly recognized forms of malware are viruses, worms and Trojans. A virus' intention is to replicate itself by attaching to another piece of software or executable code. It cannot exist by itself alone. It is activated when the software it has attached itself to is executed. The damage done by viruses can be destroyed data, errors in the programs it embedded itself in and poor system performance. A popular host for viruses is email programs. Many viruses have been written to execute when an email program, such as Microsoft Outlook, are launched and replicate itself by attacking the address book sending out the virus to everyone within the address book. A worm does not need to have a host to run, which means it can spread faster than viruses and can reach more systems. Worms infect host code and do not require a user to activate them. A Trojan disguises itself as a harmless program, fooling the system and end user into believing it is not malicious code. The full name is Trojan Horse, which references the clever attack the Greeks made on the city of Troy during the Trojan War. The Greeks presented a huge, wooden horse to the city disguised as a gift. However, Greek warriors were hiding inside the horse as a means to get inside and take over the city.

Common Attacks

Malware is one of the many forms of attacks networks experience. Backdoor attacks are created by malware opening up a port or access point into the network for intruders to access and attack a network whenever they want. Denial of Service (DoS) and Distributed Denial of Service (DDoS) are not used for getting into a network, but are used to make a service on a network unavailable by exhausting resources. DDoS use the same strategy as a DoS attack, but come from many, many machines instead of one. These types of attacks are one of the most difficult to defeat. SYN floods are a type of DoS attack that uses the three-way handshake to manipulate the attack. A client machine sends a server a SYN packet and the server acknowledges the request by sending a SYN-ACK back to the client machine. The client machine is supposed to send the server another SYN message to let it know it received its message to complete the handshake, but it does not respond. The SYN requests create many incomplete connections to the server, flooding it with traffic until it eventually shuts down. "Ping-of-Death" attacks send oversized

packets across a network causing systems to crash or freeze up. IP Spoofing occurs when an intruder learns of a specific IP range having access to resources on a network and uses the IP addresses of those trusted machines to perform attacks.[19] FTP (File Transfer Protocol) attacks use the FTP server to establish a connection with a computer inside a network by IP address and sends malicious code to it.

Financial Impact

Compromised systems and networks can have a huge financial impact on an organization. When one computer goes down, time and wages for the employee who works on the computer are just a fraction of what gets lost. Not only is the employee not able to work for the duration of the time the computer is down, resulting in a backup of work or another employee taking on double the work load, the technical staff has to spend time fixing the computer by cleaning off the malware, reinstalling software, or replacing hardware. All the effort is spent just for one computer. Now imagine what it could mean if an entire network was compromised by an intruder or had malicious code installed. If an intruder were to compromise a web server hosting an e-commerce web site, the web site could be shut down and the company could lose business. Depending on the type of attack, personal information about the customers who use the web site, such as credit card information, could be stolen. This could result in an even bigger loss for the company due to customers not trusting the security of their financial information while using the web site.

Demarc Security, Inc.'s white paper, "Threat Management: The Future of Network Security", reported an average company who experiences just one attack on their network can result in $2 million lost in revenue per incident. For a company to recover could cost up to $74,000 after hardware replacement and paying employees the time to clean up the network from an attack. The average amount of time the disruption lasts is twenty-two business hours.[20] This is all from just one attack. If a network gets attacked by a virus or worm, the costs are much, much greater.

Risk Management Solutions published a table of statistics for the major viruses and worms companies worldwide have by attacked by from 1999 to 2004. In 1999, the "Melissa" virus infected eight million computers causing $400 million in damages. In 2000, the "I Love You" virus spread through email clients infecting twenty million computers causing $1 billion in damages. The "Code Red" worm attacked ten-thousand servers costing companies $2 billion in damages. Another worm detrimental to companies was the "SQL Slammer" worm, which affected over one-hundred thousand servers resulting in $2 billion lost.[21] Forms of these worms still pop up and are detected by IDS systems

[19] "Economic Impact of Network Security Threats" white paper by Cisco Systems, Inc., 1992-2001.

[20] "Threat Management: The Future of Network Security" white paper by Demarc Security, Inc., 2004

today. Each year, the virus or worm became more sophisticated and was able to spread to more machines doing greater financial damage. The Computer Security Institute (CSI/FBI) Computer Crime and Security Survey stated this year alone, $822,000 was lost due to some form of an attack on a company's network. The financial losses included the system down time of desktops and servers, a loss of employee productivity due to the network being down, and of course the revenue lost due to the down time.[22]

IDS ESSENTIALS: SNORT, LABREA, ARPWATCH

What is Snort and What Does it Do?

Snort is an open-source NIDS software written by Marty Roesch in 1998 and distributed under the GNU Public License. It performs traffic analysis on networks in real-time, actively auditing networks. It is also signature-based, which means rules are written to determine what an attack looks like and trains the software to watch for those types of attacks. Snort started out a primarily a packet sniffer, but now it logs packet information, performs protocol analysis, and does content matching of the packets to detect an attack on the network. It can be used as a packet sniffer, a packet logger to save the packet information into a file, or a NIDS by configuring what is monitored with rule sets. A machine running Snort should have ample hard drive space due to all the logging of the network traffic, a remote capabilities to monitor the machine without needing to be physically on it, and a second NIC to do the sniffing set to promiscuous mode to monitor all the network traffic.

Snort has four basic components. The first is the packet sniffer, which listens stealthy on a network to monitor the data and IP traffic passing through. This is a great tool for network and performance analysis. The second component is the preprocessor that processes the packets. Plug-ins are added to Snort so raw packets can be checked against them to look for a certain type of behavior being made. Once the behavior has been identified and the packets have been reviewed for alerting, dropping or modification, they are passed on to the detection engine. The detection engine checks the packets against a set of predetermined rules. If the rules match the information in the packets, an alert is triggered and the information is logged to a file, which is the alerting and logging component of Snort. The preprocessor, detection engine and alert components are all plug-ins to Snort so the source code may be edited.

The rules have two parts to them: the header and the option. The rule header tells the detection engine which action to take whether it is to log the information or send an alert. The rule header also lists the type of network packet, the source and destination IP addresses, and the port affected. The rule option lists the packet information the rule should match. Rules can be customized for each network. The most efficient way to make a rule set is to download load

[21] Risk Management Solutions

[22] "Vulnerability Assessment and Remediation Management in a Distributed Enterprise Environment" white paper by eEye Digital Security, 2004

them from the Snort website at www.snort.org/dl/rules. The rules are continuously updated and modified as new attacks are discovered. When a rule is written, there are five different action options it is classified in by default. The pass option passes the packet, which just ignores it. The log option logs the packet according to how the configuration was setup. The alert option sends an alert which also logs the packet, but also alerts the administrator by whatever method was defined during the configuration. The dynamic option tells the rule to stay dormant until an activate rule option, the last of the rule options, triggers it on. Then, the dynamic rule acts like a log rule option. The activate rule option alerts and starts a dynamic rule when it is triggered, which is great for complex attacks on a system or for just categorizing the data a little differently. All of these options are installed by default, but customized rule options may also be written.[23]

Snort-Inline is a different software package used on an Inline IDS machine. However, with the latest version of Snort, 2.3, the Snort-Inline features have been integrated into the standard Snort program. Snort-Inline has the same basic function as the IDS machine running the standard Snort program, but the machine is placed inline with a network connection in front of the firewall. The IDS is configured with rule sets to drop, reject or modify network traffic that passes through it. An IDS running Snort-Inline is generally classified as an IPS that uses IDS signatures.

What is LaBrea and What Does it Do?

LaBrea is a clever piece of open-source software designed to fool intrusions or a worm attack to a network by making connection attempts think they are connected to an actual, live computer on the network when in reality, it is a virtual machine the software has created. LaBrea uses the unused IP addresses on the network to create the virtual machines. LaBrea utilizes the SYN/ACK three-way handshake to deter the attacker from the live machine. When a SYN request is made to a machine, the ACK returned is the IP information of a non-existent machine and a fake MAC address of 0:0:f:ff:ff:ff. The three-way handshake appears on the attacker's side to be complete, but the attacker is kept from performing their task. LaBrea was written in response to the CodeRed worm a few years back that was continuously attacking a network with scans. The unused IP addresses of virtual machines were utilized to slow down the worm from spreading through the network and infecting other machines.

How LaBrea works is it will listen on a network for ARP (Address Resolution Protocol) requests being made. An ARP request is an IP address translated into a physical MAC address. A request can be sent and the IP address and MAC address of the NIC on the destination machine should match.[24] If an ARP

[23] Snort 2.0 Intrusion Detection, Jay Beale, James C. Foster, Jeffrey Posluns, Brian Caswell, 2003 Syngress Publishing, Inc.
[24] http://www.inetdaemon.com/tutorials/lan/arp.html

request for an IP address is not answered within a specific amount of time, usually within three seconds, then LaBrea sends an ARP reply back with a fake MAC address so all the traffic for the IP address is lost. LaBrea tries to never capture an IP address of a live machine, which is why there are built in features to avoid capturing live IP addresses. When LaBrea is started, it does an entire sweep of the subnet and records any IP addresses responding back as excluded. If it sees an ARP signaling a new machine on a network, it will mark the IP address as excluded. Each ARP response is also marked as excluded.

One of LaBrea's great functionalities is the ability of tarpitting. Tarpitting traps an intruder's connection they make into the network and doesn't release it. It is a virtual tarpit and a means of keeping spammers out of the network. A worm, for example, can attempt an attack, but will get stuck in the tarpit. It will not be able to move through a network to infect and attack other machines. Once an intruder trying to get into a network gets their connection held, they usually are forced to give up and terminate their connection because they cannot successfully invade the network. Plus, they do not want to take the risk of getting caught.

The activity logged by LaBrea can be classified into three categories. Persist activity refers to a connection being forced into a persist state by making the TCP packet window size zero. By doing so, LaBrea is able to hold on to the connection. Capturing a local IP address indicates an ARP for an IP address has been detected and LaBrea responds back with a fake ARP. It then takes over the IP address. Additional activity is everything outside of persist activity.[25]

What is Arpwatch and its function?

Arpwatch is an open-source program matching MAC address and IP address pairings on a machine. If any changes are made to the IP address matched with the specific MAC address, an alert is made and logged. Arpwatch uses syslog to record activity and report changes.

Arpwatch reports on four major changes in activity. New activity on a network is determined by the MAC and IP address pairing. If it has not been on the network for six months or more, it will be logged as new activity. If a new MAC address arrives on a network for the first time, it is logged as a new station since the MAC has never been seen before on the network. A flip flop indicates the MAC address on the NIC has changed from the most recently seen address to the second most seen address. A changed Ethernet address means the computer has changed to a new Ethernet address.[26]

LOGCHECK

What Is Logcheck and how does it work?

[25] Gordon, Loren, www.sourceforge.net, [Labrea-users] log format question, April 20, 2004.
[26] Craig Leres of the Lawrence Berkeley National Laboratory Network Research Group, University of California, Berkeley, CA

The software is in place to audit our network, now we need a way to compile all the information the three software packages gathered for us into one report. Logcheck is an open-source software package covered by the GNU Public License designed to run and check system log files automatically for security breaches and unusual activity. Rule files are written to automatically spot any problems or security violations recorded in the log files. Included in the rules are words for Logcheck to look for or ignore in the log files. Logcheck aids in automating the tedious process of auditing and does not record any normal activity on the network in the logs. If it did, the logs would be overwhelming to review. This way, any violations on the network or intrusions can be spotted easier and acted on faster. Reviewing information logged from an attack by the IDS is extremely important. The logs can be reviewed and analyzed to prevent future attacks to a network. In some cases, if an attacker is caught, log files can be used for criminal prosecution.[27]

Logcheck uses a program called logtail to review the log files for changes. Logtail remembers the last place it read from in the file and looks for any changes added or removed. The Logcheck script should run every hour on the host machines. Any unusual activity reported will be recorded in the log. A cron job can be written in Linux to email a complete report Logcheck compiles from the IDS machines. The report from Logcheck is what we use to review the findings on a company's network. The files constantly monitored by Logcheck are *messages*, *authlog*, *kernlog* and *syslog* all within the */var/log/* directory.[28]

Logcheck will report on exactly what is specified in the rule files. Unfortunately, a lot of false negatives get reported, and can cause headaches for the network administrator who has to sort through and check all the logs. So, to cut down on the false positives reported, Logcheck has specific directories setup to enter information into files to ignore when it runs the report. Logcheck has three categories of events it reports. They are active system alerts, possible security violations, and unusual system events. At the top of the Logcheck report are the active system alerts and possible security violations. Since these are the most critical to a company's network, they are listed first. However, some activity on a network can be non-threatening activity such as a network administrator issuing a port scan for testing purposes. An alert will be triggered and reported by Logcheck a system attack may be taking place. This is what a false positive is because no threat is being made to the network. If the network administrator is going to be doing a lot of testing using a port scan, they may issue a line in one of the Logcheck ignore files to ignore their specific IP address if a port scan is issued from it. The entry can be made in the directory *logcheck.violations*.[29]

---

[27] Warren, Trevor, "Intrusion Detection Systems, Part IV: Logcheck", http://www.freeos.com/articles/3540/, February 12, 2001.
[28] Warren, Trevor, "Intrusion Detection Systems, Part IV: Logcheck", http://www.freeos.com/articles/3540/, February 12, 2001.
[29] "Intrusion Detection Systems, Part IV: Logcheck, Trevor Warren, http://www.freeos.com./articles/3540/, February 12, 2001.

PUTTING IT ALL TOGETHER: WHAT EACH PROGRAM TELLS US USING LOGCHECK

Now that we know what an IDS system is and what it does for a network, we can use the four pieces of software previously mentioned to start analyzing what Logcheck has compiled for us. Depending on the state of a network, some of the reports Logcheck generates can be rather lengthy if there is a lot to report. We have set the Logcheck configuration to generate a report every hour, two minutes past the hour, on all the IDS machines we are monitoring. Below are sections of a report generated by Logcheck after we did a portscan on our network. The full Logcheck report emailed was 154 pages. When the portscan was executed, the traffic flowing on the network was being attracted back to the IDS box by LaBrea and Arpwatch. Snort reported on the portscan itself and gave us details on its activity. Also, the network in this example was running DHCP.

Logcheck Report

Subject: hostname 2004/11/12 13:02
From: root<root@hostname.companyname.com>
Date: Fri, 12 Nov 2004 13:02:04 -0500
To: root@hostname.companyname.com

```
This mail is sent by logcheck. If you do not want to receive it any more, please modify
the configuration files in /etc/logcheck or deinstall logcheck.

Unusual System Events
=-=-=-=-=-=-=-=-=-=-=
Nov 29 13:55:15 hostname snort: [1:469:3] ICMP PING NMAP [Classification: Attempted
Information Leak] [Priority: 2]: {ICMP} 192.168.160.128 -> 192.168.160.148
Nov 29 13:55:15 hostname snort: [1:384:5] ICMP PING [Classification: Misc activity]
[Priority: 3]: {ICMP} 192.168.160.128 -> 192.168.160.148
Nov 29 13:55:15 hostname snort: [1:408:5] ICMP Echo Reply [Classification: Misc activity]
[Priority: 3]: {ICMP} 192.168.160.148 -> 192.168.160.128
Nov 29 14:01:27 hostname arpwatch: new station 192.168.160.49 0:0:f:ff:ff:ff
Nov 29 14:01:27 hostname /usr/sbin/LaBrea: Capturing local IP: 192.168.160.192
Nov 29 14:01:27 hostname arpwatch: new station 192.168.160.50 0:0:f:ff:ff:ff
Nov 29 14:01:27 hostname arpwatch: new station 192.168.160.160 0:0:f:ff:ff:ff
Nov 29 14:01:27 hostname arpwatch: changed ethernet address 192.168.160.162 0:0:f:ff:ff:ff
(0:c0:9f:4b:9b:74)
Nov 29 14:01:27 hostname /usr/sbin/LaBrea: Additional Activity: 192.168.160.128 59628 ->
192.168.160.171 80
Nov 29 14:01:27 hostname /usr/sbin/LaBrea: Additional Activity: 192.168.160.128 59628 ->
192.168.160.172 80 *
Nov 29 14:01:27 hostname /usr/sbin/LaBrea: Additional Activity: 192.168.160.128 59628 ->
192.168.160.173 80
Nov 29 14:01:27 hostname /usr/sbin/LaBrea: Additional Activity: 192.168.160.128 59628 ->
192.168.160.174 80 *
Nov 29 14:01:27 hostname /usr/sbin/LaBrea: Additional Activity: 192.168.160.128 59628 ->
192.168.160.175 80
Nov 29 14:01:27 hostname arpwatch: report: pausing (cdepth 3)
Nov 29 14:01:27 hostname /usr/sbin/LaBrea: Additional Activity: 192.168.160.128 59628 ->
192.168.160.176 80 *
Nov 29 14:01:27 hostname /usr/sbin/LaBrea: Additional Activity: 192.168.160.128 59628 ->
192.168.160.177 80
Nov 29 14:01:27 hostname /usr/sbin/LaBrea: Additional Activity: 192.168.160.128 59628 ->
192.168.160.179 80 *
Nov 29 14:01:27 hostname /usr/sbin/LaBrea: Additional Activity: 192.168.160.128 59628 ->
192.168.160.180 80
```

```
Nov 29 14:01:27 hostname /usr/sbin/LaBrea: Additional Activity: 192.168.160.128 59628 ->
192.168.160.181 80 *
Nov 29 14:01:27 hostname /usr/sbin/LaBrea: Additional Activity: 192.168.160.128 59628 ->
192.168.160.182 80
Nov 29 14:01:27 hostname /usr/sbin/LaBrea: Additional Activity: 192.168.160.128 59628 ->
192.168.160.183 80 *
Nov 29 14:01:27 hostname arpwatch: new station 192.168.160.52 0:0:f:ff:ff:ff
Nov 29 14:01:27 hostname /usr/sbin/LaBrea: Additional Activity: 192.168.160.128 59628 ->
192.168.160.187 80
Nov 29 14:01:27 hostname /usr/sbin/LaBrea: Additional Activity: 192.168.160.128 59628 ->
192.168.160.190 80 *
Nov 29 14:01:27 hostname /usr/sbin/LaBrea: Additional Activity: 192.168.160.128 59628 ->
192.168.160.191 80
Nov 29 14:01:27 hostname /usr/sbin/LaBrea: Additional Activity: 192.168.160.128 59629 ->
192.168.160.192 80 *
Nov 29 14:01:27 hostname /usr/sbin/LaBrea: Additional Activity: 192.168.160.128 59629 ->
192.168.160.191 80
Nov 29 14:01:27 hostname arpwatch: new station 192.168.160.53 0:0:f:ff:ff:ff
Nov 29 14:01:27 sniffer_gold snort: [1:2000587:3] BLEEDING-EDGE Malware SpywareLabs
VirtualBouncer Seeking Instructions [Classification: A Network Trojan was detected]
[Priority: 1]: {TCP} 192.168.160.128:59628 -> 192.168.160.180:80
Nov 29 14:01:28 hostname arpwatch: flip flop 192.168.160.177 0:0:f:ff:ff:ff
(0:d:56:80:c9:63)
```

Snort

The IDS we used to monitor the portscan test has the default rule settings for Snort, version 2.3, downloaded from www.Snort.org and www.bleedingsnort.com. Snort can be customized to record the information you want to capture. Signatures can also be manually written for Snort at www.bleedingsnort.com and imported into the program. The options are almost endless; however, the focus of the paper is to analyze what Snort reports when an attack occurs. As an example, we are analyzing a portscan. In the Logcheck report above, IP address 192.168.160.128 represents the machine in which the portscan originated from. The IDS machine has an IP address of 192.168.160.148. At the top of the report, Snort records when the portscan started (Nov 29 13:55:15), what type of attack and the program used (ICMP PING NMAP), where it originated from (192.168.160.128), and what machine it was targeting (192.168.160.148). On the next two lines of the report, we can see where the request has been sent by the attacker, in our case, machine 192.168.160.128, and answered by the IDS machine.

The IDS machine used in the example has default and custom rule sets configured for Snort. One of the rule sets is to detect malware. During the portscan, Snort detected one of the machines on the network to have a Trojan virus on it, which is the last entry in our Logcheck report above. The reporting of malware is very useful information for determining what machines may have been compromised on the network. Along with the type of attack or compromise, source IP and destination IP in the Snort entries, a signature ID appears at the beginning of the Snort entry in brackets. The signature identification number can be used to look up exactly what the message Snort has reported by going to www.snort.org and entering the signature identification number. The signature identification number can then be used to further investigate what has been compromised and to what extent the damage could

be.

Now we will look at what LaBrea can tell us about the portscan attack.

LaBrea

Most of LaBrea's activity will appear under Unusual System Events in Logcheck. All the activity reported by LaBrea in the portscan attack is classified as additional activity. LaBrea detected the portscan and immediately went to work trying to grab the unused IP the scan was trying to capture. It is classified as additional activity because the IDS machine was receiving unusually more traffic than normal. Each entry recorded by LaBrea staes the source IP address of the IDS machine. The destination IP address is the unused IP address LaBrea claimed. The port in which the attack was made on is also reported after the IP address. We can use the information to detect which IP addresses are being captured and from what machine the attack is coming from. In our example, it is from an internal address so it is relatively easy to find what machine the attack came from. If an attack came from an external source, the public IP address would be listed and could be tracked down. From there, two actions can be made. The first could be to block the public IP from coming into the network again. This can be done on an In-line IDS machine configured to recognize the public IP address and then drop any packets coming into the network from it. The second option could be legal action against the attacker. In most cases, companies are too swamped with work to bother with the legalities of attacks if no damage was done. However, if there was damage done to the company's information from someone accessing their network utilizing what they found from the portscan, it may be worth it to prosecute.

Arpwatch

Arpwatch lets us know of any activity of machines on our networks where their IP address paired up with a specific MAC address has been changed. The reports are sent as emails with the type of report in the subject field of the email. The type of reports Arpwatch generates are new activity, new station, flip flop, and changed Ethernet address. The new activity report gives us the host name, the IP address of the machine, the Ethernet or MAC address, the Ethernet vendor, the time stamp with the exact date and time, the previous time stamp of the date and time of the last time it was recognized by Arpwatch on the network, and delta which tells us the amount of time passed between the previous time stamp and the current time stamp. No new activity was reported by Arpwatch when the portscan was done, but below is an example of what a new activity email will look like.

A new activity report email will look like this:

> **Subject:** new activity
> **From:** arpwatch@hostname.companyname.com (Arpwatch)

Date: Wed, 27 Oct 2004 15:32:11 -0500
To: arpwatch@hostname.companyname.com

hostname: <unknown>
ip address: 169.254.179.50
ethernet address: 0:8:2:92:a6:56
ethernet vendor: Compaq Computer Corporation
timestamp: Wednesday, October 27, 2004 15:23:11 -0500
previous timestamp: Tuesday, November 11, 2003 8:44:48 -0500
delta: 351 days

The next report type, new station, gives us the host name, the IP address of the machine, the Ethernet or MAC address, the Ethernet vendor, and the time stamp of the exact date and time the new machine arrived on the network. When the portscan was done, LaBrae took over the unused IPs on the network and reported them back to the IDS machine. In doing so, the unused IPs then "belonged" to a machine and made it appear as if they were new stations on the network.

A new station report will look like this:

Subject: new station
From: arpwatch@hostname.companyname.com (Arpwatch)
Date: Mon, 29 Nov 2004 14:01:27 -0500
To: arpwatch@hostname.companyname.com

hostname: <unknown>
ip address: 192.168.160.160
ethernet address: 0:d:60:78:d9:3d
ethernet vendor: Next, Inc. [Ne XT]
timestamp: Monday, November 29, 2004 14:01:26 -0500

A flip flop report gives us the same information a new activity report sends out with the addition of the old Ethernet IP address and vendor. The report tells us a machine has changed from the most recently seen IP address to the second most recently seen IP address for the machine. The delta gives us the amount of time lapsed between the old Ethernet IP address and the new Ethernet IP address. The report below shows the IP number 192.168.160.177 originally belonged to a machine with a MAC address of 0:d:56:80:c9:63. At the time of the portscan, the IP address no longer belonged to the machine and LaBrae assumed the unused IP address. Now, the new MAC address is 0:0:f:ff:ff:ff for the IP address. Arpwatch tells us the MAC address 0:0:f:ff:ff:ff has belonged to the IP address before, which results in a reporting of a flip flop.

A flip flop report will look like this:

Subject: flip flop

From: arpwatch@hostname.companyname.com (Arpwatch)
Date: Mon, 29 Nov 2004 14:01:28 -0500 -0500
To: arpwatch@hostname.companyname.com

hostname: <unknown>
ip address: 192.168.160.177
ethernet address: 0:0:f:ff:ff:ff
ethernet vendor: Next, Inc. [Ne XT]
old ethernet address: 0:d:56:80:c9:63
old ethernet vendor: <unknown>
timestamp: Monday, November 29, 2004 14:01:26 -0500
previous timestamp: Friday, November 5, 2004 15:08:15 -0500
delta: 23 days

The final report, changed Ethernet address, gives us the exact same information as a flip flop report. The only difference is, this report states there has been a change in the Ethernet MAC address on the host machine to a new address. The report below shows IP number 192.168.160.162 originally belonged to a machine with a MAC address of: c0:9f:4b:9b:74. At the time of the portscan, the IP address no longer belonged to the machine and LaBrae assumed the IP address, which assigned a MAC address of 0:0:f:ff:ff:ff to the IP address. Since the IP address and MAC address have never been paired up before, this is not a report of a flip flop, but of a changed Ethernet address.

A changed Ethernet address report will look like this:

Subject: changed ethernet address
From: arpwatch@hostname.companyname.com (Arpwatch)
Date: Mon, 29 Nov 2004 14:01:27 -0500
To: arpwatch@hostname.companyname.com

hostname: <unknown>
ip address: 192.168.160.162
ethernet address: 0:0:f:ff:ff:ff
ethernet vendor: Next, Inc. [Ne XT]
old ethernet address: 0:c0:9f:4b:9b:74
old ethernet vendor: Quanta Computer, Inc.
timestamp: Monday, November 29, 2004 14:01:26 -0500
previous timestamp: Thursday, September 30, 2004 11:05:18 -0500
delta: 60 days

SUMMARY

In conclusion, every company needs an additional means of protecting its network and valuable information from being compromised by an intruder. Without Intrusion Detection Systems on their network, there is no way for a company to detect or catch an intrusion. Using an IDS with Snort installed on it can report about malicious activity taking place on a network in real time including spyware and portscans. Adding LaBrea to the IDS system can create tarpits of virtual machines to trap intruders from scanning the company's network and report false MAC addresses to protect the live machines already on

the network. Finally, installing Arpwatch can determine if an unrecognized or unfamiliar machine is plugged into a company's network and if a machine has changed its identifying IP address/MAC address pairing. Reviewing the reports Logcheck prepares from the findings of these three software packages can help a company determine, in real time, unusual activity taking place on their network, and defend itself against an attack from an inside or outside intruder.

REFERENCES

Beale, Jay, Brian Caswell, James C. Foster and Jeffrey Posluns, "Snort 2.0 Intrusion Detection", Syngress Publishing, Inc., 2003.

Cisco Systems, Inc., "Economic Impact of Network Security Threats", 1992-2001.

Coburn, Andrew, PHD, 60 RISK & INSURANCE, Top 10 Risks: Cyber Attack, "Pentecost Worm Unleashed On Computer Networks", http://www.rms.com/Publications/10GrtCats_Computer_R&I_041504.pdf#search='pentecost%20worm%20unleashed%20on%20computer%20networks', April 15, 2004.

Connell, Graeme, "IDS: Re: definition for Inline IDS/IPS", Insecure.org forum, September 27, 2004.

Demarc Security, Inc., "Threat Management: The Future of Network Security", 2004.

eEye Digital Security, "Vulnerability Assessment and Remediation Management in a Distributed Enterprise Environment", 2004.

Gordon, Loren, www.sourceforge.net, [Labrea-users] log format question, April 20, 2004.

Hontanon, Ramon J., "Emerging Technology: Deploying an Effective Intrusion Detection System", http://www.networkmagazine.com/article/NMG20000830S0003, September 5, 2000.

InetDaemon Enterprises, "Address Resolution Protocol", http://www.inetdaemon.com/tutorials/lan/arp.html, 1996-2004.

The Intrusion Detection System Group, http://www.intrusion-detection-system-group.co.uk/, 1993-2001.

Leres, Craig, Lawrence Berkeley National Laboratory Network Research Group, University of California, Berkeley, CA., Arpwatch manual.

NSS Group, “Internet Prevention Systems (IPS)”, January 2004, http://www.nss.co.uk/WhitePapers/intrusion_prevention_systems.htm.

SANS Institute Track 1-SANS Security Essentials, Volume 1.3, “Internet Security Technologies”, 2004.

SANS Institute Track 1-SANS Security Essentials, Volume 1.4, “Secure Communications”, 2004.

Security Wizardry, “Network Intrusion Detection Systems”, http://www.securitywizardry.com/N_ids.htm, 2004.

Top Layer Networks, “Beyond IDS: Essentials of Network Intrusion Prevention”, November 2002.

Warren, Trevor, “Intrusion Detection Systems, Part IV: Logcheck, http://www.freeos.com./articles/3540/, February 12, 2001.