GIAC
CERTIFICATIONS

Global Information Assurance Certification Paper

Copyright SANS Institute
Author Retains Full Rights

This paper is taken from the GIAC directory of certified professionals. Reposting is not permited without express written permission.

Interested in learning more?

Check out the list of upcoming events offering
"Security Essentials Bootcamp Style (Security 401)"
at http://www.giac.org/registration/gsec

Option 2 Case Study:

Enhancing ABC Inc's Security Strategy with IDS and Centralized Syslog

SANS GSEC Practical 1.4B

George Plytas
Date Submitted: July 05, 2004

© SANS Institute 2004, Author retains full rights.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Abstract

I am a Security Analyst/Administrator for a medium sized company, ABC Inc I, along with a team of System Administrators, am tasked with the responsibility of protecting our customer's confidential information, maintaining the integrity of our applications and keeping our systems available. We have been employing many industry best practices that were subsequently discussed throughout my SANS GSEC course. These include OS patching, application hardening, network tuning, Anti-Virus and Vulnerability Assessment. ABC Inc has gapped in some areas of defense in depth, a concept that is summarized in the following quote:

> "The concept of defense in-depth is simple…we need to be certain that if one countermeasure fails, there are more behind it. If they all fail, we need to be ready to detect that something has occurred and clean up the mess expeditiously and completely, and then tune our defenses to keep it from happening to us again." (Cole, p.15)

To date, our efforts have been focused on preventing unauthorized events from occurring with minimal attention to detection. From the above quote, the concept of defense in-depth states that one has not only to thwart intruders, be it a hacker, worm, virus, etc., but also to sense their presence and collect the appropriate data to mitigate the risk of reoccurrence. The goal of this paper is to implement an optimal, cost-effective security solution to progress our defense strategy. I will discuss the implementation of an Intrusion Detection System (IDS) using Snort and centralized logging with syslog. By focusing on these two key areas, ABC Inc will significantly improve our base line posture and decrease the risk of future exposure to related vulnerabilities.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Table of Contents

1. Introduction 4

2. Before 5
2.1 Description of the Problem 5
2.2 Architecture 5
2.3 Prevention Techniques 6
2.3.1 Physical Security 6
2.3.2 Security Newsletters/Mailing-Lists 6
2.3.3 Policies and CSIRT Team 7
2.3.4 CSIRT 7
2.3.5 Operating System Hardening and Vulnerability Assessment 7
2.3.6 Network Considerations 8
2.3.7 Anti-Virus (AV) 8
2.4 Risks/Vulnerabilities: Why Invest in IDS and Centralized Logging 8

3. During 9
3.1 How Did I Approach the Problem 9
3.2 Requirements for Proposed Solution 9
3.2.1 Intrusion Detection 9
3.2.2 Centralized Syslog 9
3.3 Research to Meet the Requirements 10
3.4 Cost Analysis for Required Enhancements 11
3.5 Gathering Required Software Packages 11
3.5.1 Snort and Contributing Software 11
3.6 Building the System 13
3.6.1 Hardening the server 13
3.6.2 Installing/Configuring Snort 14
3.6.3 Installing/Configuring ACID 18
3.6.4 Installing/Configuring Pigsentry 20
3.6.5 Configuring Syslog 21
3.6.6 FWAnalog 21
3.6.7 Installing/Configuring FISQ 23

4. After 24
4.1 Outline of Steps Taken 24
4.2 Enhanced State of Security 24

5. List of References 26

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

1. Introduction

> "In the Looking Glass Land, the Queen tells Alice, "It takes all the running you can do, to keep in the same place." (http://encyclopedia.thefreedictionary.com)

The first time I heard the Red Queen Hypothesis was in a 2nd year lecture on Evolutionary Biology. I was intrigued by the idea that all living creatures have to continually change only to maintain their place in their ecosystem. What this concept means to me is that an organism needs to continue evolving in order to meet the changing needs of their environment. Much like other good quotes, the Red Queen Hypothesis has been inherited by various lines of study. Her words ring true in many branches of technology specifically, and for the purposes of this paper, in Information Security.

The paper that follows addresses ABC Inc's required improvements to our security posture in order to maintain our place in the market. Like living organisms, a Corporation is at constant risk of being exposed to a malicious virus, attacks, foreign hosts or worms. If for instance, an animal had a very strong immune system that could deflect waves of known attacks; one might be inclined to think that it is well suited to its surroundings. However, if this animal was to come in contact with something that its immune system did not detect, despite its ability to defend against it, it may become sick or even perish. A Corporation is much like the animal in this analogy. In the Corporate world of system security this translates to a breach of security due to myriad of attack vectors and its immune system is synonymous with the concept of defense in-depth.

Using a preventative approach towards our infrastructure is our first line of protection for our small but critical infrastructure i.e. Public Web Sites and Ecommerce applications. GSEC training provided me with the knowledge that detection is a critical piece of a sound infrastructure. Intrusion Detection of unauthorized activity will support ABC Inc in our overall security strategy, aiding in alert of events and the clean up and tweak of architecture to avoid future incident. My paper will begin with a brief overview of preventative defense in-depth measures taken by ABC Inc, I will then discuss the vulnerabilities and risks that we face by not investing in Intrusion Detection and/or having a central log repository for trend analysis. In the During section, I will walk through the deployment of both of these techniques and additional tools used to compliment them. In the After section I will discuss our new security posture and how these tools assisted in better positioning ABC Inc's defense in-depth strategy and have aligned us with industry base-line security practices.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

2. Before

ABC Inc has been employing many industry best practices that were discussed throughout my SANS GSEC course. These include OS patching, network tuning, Anti-Virus and vulnerability assessments, among others. While each of these solutions are critical, they do not address all areas of a sound security defense strategy.

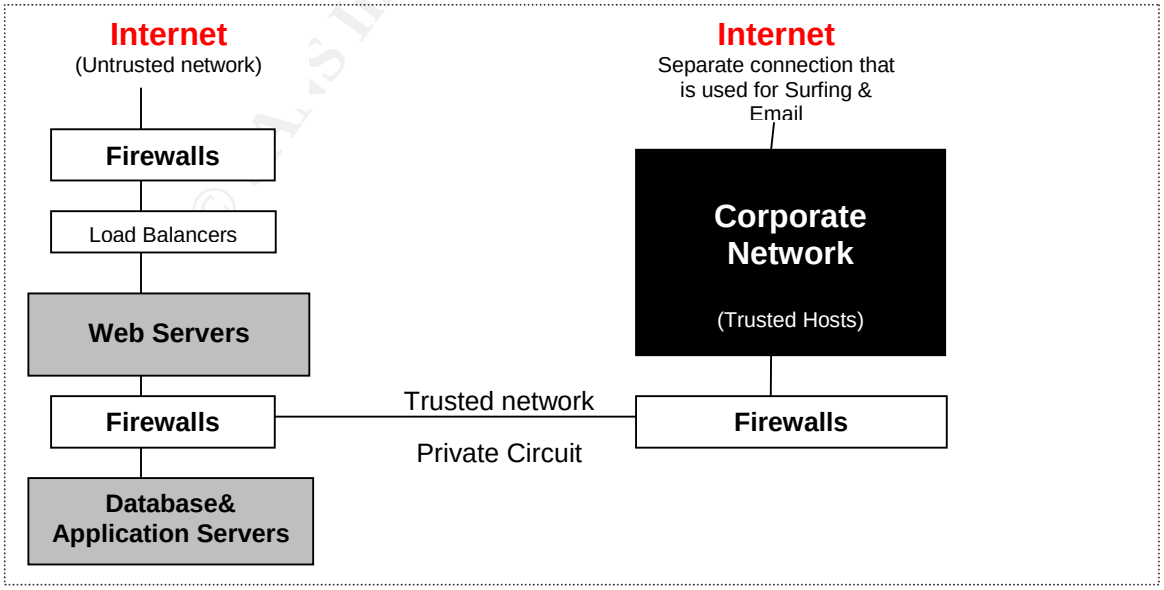
2.1 Description of the Problem

Our efforts were focused on preventing unauthorized events from occurring with minimal attention to detection and centralized logging. The SANS GSEC course identified how important it is to invest in security measures focusing on these two key areas.

Throughout this paper, I will set the stage by providing an overview of the architecture and preventative techniques that were in place before I attended the course. Upon completion of my GSEC course, I felt ABC Inc was at risk by not deploying an Intrusion Detection System and a centralized logging server where analysis could be performed.

2.2 Architecture

ABC Inc used a logical architecture layout for our servers separating machines into areas based on the duties they perform. Web servers are in a DMZ while an additional layer of firewalls further protects Application/Databases servers. The Internet connection for our server farm is used only for customer's access to our applications. We maintain a separate connection to the Internet used for Web Surfing and Email. While the environment is a relatively small component of our overall architecture, it is very important to ABC Inc as it houses our Public Internet site and other high profile applications. The diagram below illustrates the infrastructure prior to addressing intrusion detection.



© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

2.3 Prevention Techniques

Not investing in prevention is similar to leaving the front door of your house wide open. You would not do this even if you had an alarm system installed or a surveillance camera because your home would still be at risk. As a result, the team at ABC Inc invested efforts in preventing an attack from happening. Although prevention alone does not cover every aspect of a sound defense in-depth strategy, it is the most logical starting point.

Listed below are the preventative techniques used by ABC Inc to maintain system availability, protect client information and preserve the integrity of our systems.

2.3.1 Physical Security

ABC Inc houses its Internet-facing servers at a third party data centre that provides a variety of physical security controls. These include:

- Internet feeds supplied by multiple Carriers.
- Redundant power supply with UPS and Diesel Generator back-up.
- Surveillance camera monitored by 24/7/365 on site Security Guards.
- Advanced Fire Suppression.
- Combination Card-Reader and Biometrics, fingerprint scans, to gain access to secured areas. In addition, all entries to secured areas are logged.

These security measures are valuable as they safeguard ABC Inc from physical threat. No further investment in this area was required as current practices meet industry standards.

2.3.2 Security Newsletters/Mailing-Lists

As the Security Administrator of ABC Inc, I subscribe to various mailing lists to stay informed of new vulnerabilities. Our systems are primarily Red Hat Linux, Microsoft Windows 2000 and Cisco. I have included the mailing lists to which I am a member and links to their respective subscription sites:

- Security Focus [http://www.securityfocus.com/archive]
- RedHat [https://www.redhat.com/security/team/advisories.html]
- Microsoft [http://www.microsoft.com/security/bulletins/alerts.mspx]
- NAI [http://vil.nai.com/vil/join-DAT-list.asp]

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

2.3.3 Policies and CSIRT Team

It is important that Information Security Policies are clearly documented, signed off by upper management and easily accessible to the employees within the organization. At ABC Inc, our Information Security policy is hosted on our Intranet and covers many important topics, some of which include:

- Vulnerability Assessment of new servers,
- Firewall Rule change procedures,
- Password Minimal-Length and Strength,
- Regularity of Log file analysis,
- Analogue line usage, etc.

2.3.4 CSIRT

Our CSIRT, otherwise known as Computer Security Incident Response Team, has representatives from various lines of business, including Internal/External Communications, Information Technology, Information Security and Legal, when required. The incident response strategy is clearly documented and all members have access to these procedures. Each team member has a well-defined role and knows his or her function in the event of an incident. If ABC Inc did not have these procedures clearly defined; a minor situation could quickly develop into a major incident, resulting in an unnecessary investment of time, money and resources to resolve the situation. At ABC Inc, it is also a regular practice to hold post incident meetings to discuss lessons learned and how to prevent reoccurrence.

2.3.5 Operating System Hardening and Vulnerability Assessment

Operating Systems (OS) are a great example of the Red Queen Hypothesis, as mentioned in the Introduction to this paper. OS's continually need updating to maintain a security base line. ABC Inc has acquired tools to aid in Patch Management and Operating System Hardening. Nessus, an open-source Vulnerability Assessment tool, is used and complimented with Pedestal Software's Security Expressions package. An admirable feature of this software is its agentless auditing of desktops/servers spanning multiple Operating Systems that including Unix/Linux and Microsoft Windows NT/2000/XP. It can be used with predefined, customizable policy standards from SANS, NSA, NIST, Microsoft and Department of the Navy. While these are the primary tools used for OS Hardening and Vulnerability Assessment, I also use Nmap, GFI Languard's NetworkScanner and HPing2 for audits and assessments**.**

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

2.3.6 Network Considerations

All traffic from the Internet passes through border routers that are maintained by our hosting company and are configured to filter unwanted activity. Each firewall in ABC Inc's network design is configured with the concept of least privilege; deny all access except what is explicitly permitted to perform a pre-defined, approved function. Switches in our environment with resident firewall connections are configured with only one VLAN to prevent circumvention of access controls and/or configuration errors leading to firewall bypass.

2.3.7 Anti-Virus (AV)

An Anti-Virus program is installed on all Windows based systems at ABC Inc. Each system checks in with a central server on a daily basis for new virus definition updates. The activity from the AV program generates log messages, which are sent to the Windows Event Log.

ABC Inc invested significant effort in each of the areas above to prevent unauthorized access, however, prevention alone does not cover every aspect of a sound defense in-depth strategy.

2.4 Risks/Vulnerabilities: Why Invest in IDS and Centralized Logging

After completing my SANS course, I recognized that there were two key areas of vulnerability with our security defense. We did not have the following:

1. A Network Intrusion Detection System: This type of application is designed to sense malicious activity occurring on the network and provides real-time alerting to Administrators to investigate. The lack of such a system leaves ABC Inc at risk by not having the visibility of inappropriate network traffic and relying solely on a system events i.e. system crashes, to be alerted of malicious activity.

2. Centralized Logging with Syslog: This tool helps maintain the integrity of security and system logs by directing events to a centralized logging repository where they can be analyzed for trends. In the event of an attack, these logs are used as evidence to explain how an intruder bypassed various security controls.

Investment in these tools would provide ABC Inc with real-time alerting and the information required to perform forensic investigation.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

3. During

3.1 How Did I Approach the Problem

While the preventative techniques previously discussed provide an excellent starting point for a sound defense in-depth strategy, the two key areas of improvement needed to be tackled in a logical fashion. The first step was to gather requirements to address the vulnerability. Investigating the various solutions available to meet the requirements along with a cost analysis of these solutions followed.

3.2 Requirements for Proposed Solution

After identifying these gaps, I consulted with the Administrator team and together we determined requirements for the deployment of an Intrusion Detection System and Centralized Syslog server.

3.2.1 Intrusion Detection

This system would need to provide:

- Sensors positioned in ideal locations to capture major system events
- Real-time alerting for new events or reoccurring events
- A user friendly GUI to view alerts
- The ability to correlate events/alerts
- Easy management of alerts – preferably in a database for easy querying
- A non-disruptive deployment to existing network flow i.e. transparent deployment

3.2.2 Centralized Syslog

This system would need to provide:

- A consolidated store of important syslog messages of *nix, Windows 2000, Cisco equipment
- The ability to send firewall generated messages to a database in order to assist in the correlation of IDS events
- Process firewall logs for trend analysis and insight
- Non-disruptive changes to existing systems

The combination of these two solutions also needed to provide a means to identify and correct false positives from occurring to the Intrusion Detection System.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

3.3 Research to Meet the Requirements

The IT Security market is saturated with numerous products to meet each of the above requirements several of which were discussed during my GSEC training. Of particular mention during the course was the open source Network Intrusion Detection System Snort. My initial impression, before investigating Snort's capabilities, was that it could not meet the requirements of a robust Intrusion Detection System so I set out to compare its features to proprietary solutions.

First, I compared its ability to use customizable policies for each interface. Snort met this requirement while some of the available products could not. I set out to find a solution with an intuitive Graphical User Interface (GUI). Each of the products available for purchase had this feature while Snort's default installation package did not. Further investigation revealed that Analysis Console for Intrusion Detection (ACID), contributing software for Snort, was freely available and met this requirement along with some additional requirements. ACID had the ability to correlate events detected by Snort into logical groupings that could be sorted by source/destination IP, source/destination port, alert classification, unique alerts listings, searchable database and graphing of alert trends.

This peaked my interest in Snort and its contributing software so I decided to take an in depth look at its additional abilities. It quickly became apparent that the open-source community had tremendous following for the product with user-group postings and various articles that could answer most questions related to installation and ongoing operation of a Snort deployment. I also discovered that real-time alerting could be accomplished by using Pigsentry. Pigsentry is another contributing package that works in conjunction with Snort and triggers an email to administrators when a new alert is detected. Additional emails are sent to administrators when an increased pattern is detected for existing alerts.

The only shortcoming of using Snort was the time contribution required to install the system. Following the installation, the maintenance of the system could be automated and the time used to keep the system up to date is comparable with other solutions.

Snort, along with contributing software and zero cost, was chosen as the IDS solution to be used by ABC Inc.

The de-facto standard for logging on Unix is syslogd. According to the man pages, syslogd(8), it supports Internet and Unix Domain sockets and has the ability to capture remote host logs making it a suitable solution for ABC Inc's Central Log repository. Many tools exist to parse the logs and provide useful output, such as fwanalog. It could be configured to parse log files for critical events. In addition, the FISQ program could be used to load firewall logs into a database for easy querying and to assist in analysis of Snort events.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

3.4 Cost Analysis for Required Enhancements

SNORT: Cost $0
When investigating potential options for IDS, I realized that there were various solutions available. They ranged in price from zero (open-source) to approximately forty thousand dollars. I aspired to find a solution that was flexible, cost effective and reliable. SNORT, an open source NIDS met all these requirements. When combined with other contributing software, it became a very powerful tool.

Unix Syslog Daemon: Cost $0
The de-facto standard used for logging system-generated messages can be run on all flavors of Unix/Linux. This program is included with the base operating system; therefore, no additional costs were incurred for deployment. Much like IDS, supplementary, zero cost tools were required to conduct log analysis i.e. Fwanalog.

The only expenses for ABC Inc, to set-up the above systems, was the licensing cost of RedHat Enterprise Server ($1500/per year) and the cost of hardware, which was approximately a $2200 one-time fee with a three year maintenance cost of ~$400. This provided a server with the following hardware:

1 x 1U IBM X335 XEON / 2.4 GHZ with 512KB/512MB 24X CD
1 x PCI Ultra SCSI Controller
2 x 73.4 GB SCSI Hot Swap HDD
2 x 10/100/1000 Integrated Ethernet Card
1 x Intel Quad card

This analysis validated that the costs were minimal compared to the significant benefits these tools would provide to further enhance our security position. My manager approved funding for the hardware and Operating System costs.

3.5 Gathering Required Software Packages

3.5.1 Snort and Contributing Software

The Snort.org download site houses some of the packages that were required to set up Snort, ACID and Pigsentry. ACID had additional dependencies that are listed in the table below with their respective sites.

The Snort download site also had links to Roman Danyliw's installation instructions for ACID (http://www.andrew.cmu.edu/user/rdanyliw/snort/acid_config.html). From these instructions, I learned that ACID required the following packages to function correctly:

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

This data is available at: http://www.andrew.cmu.edu/user/rdanyliw/snort/acid_config.html

Package Name	Required Version	Description	Source of Package
MySQL	3.23.x+	Open-source relational database	**www.mysql.com**
Snort	1.7+	Network Intrusion Detection System	**www.snort.org**
PHP	4.0.4+	Web scripting language	**www.php.net**
Apache Server	1.3.x*+	HTTP server	**www.apache.org**
ADODB	1.2+	PHP Database abstraction library	**www.weblogs.com/adodb**
JPGraph	1.8+	PHP chart library/graphing library	**www.aditus.nu/jpgraph**
GD	1.8.*	Image manipulation library	**www.boutell.com/gd**
Libpng		PNG libraries	**www.libpng.org/pub/png**
Libpjpeg-6b		JPEG libraries	**www.jig.org**
Zlib		Compression library	**www.gzip.org/zlib**
ACID	0.9.6+	Analysis Console for Intrusion Detection	

I am using a RedHat Enterprise Server with support from the RedHat Network (RHN) so I first checked there to see if any of the programs were available. The advantage of using RPM packages from RHN is the ability to apply security fixes to previously installed packages. The only stipulation is that your server must be registered with RedHat.

Typing 'up2date –u' at the prompt of a server with Internet access automatically updates your packages if you desire. A great deal of flexibility exists with the up2date program providing you with dozens of options to suit your needs.

Package Name	Version Listed on RedHat Networks
MySQL	mysql-3[1].23.58-1.i386.rpm mysql-devel-3[1].23.58-1.i386.rpm mysql-server-3.23.58-1.i386.rpm
Snort	Not available on RHN
PHP	php-4[1].3.2-11.ent.i386.rpm php-mysql-4[1].3.2-11.ent.i386.rpm
Apache Server	httpd-2[1].0.46-32.ent.i386.rpm mod_ssl-2[1].0.46-32.ent.i386.rpm
ADODB	Not available on RHN
JPGraph	Not available on RHN
GD	Gd-1[1].8.4-12.i386.rpm
Libpng	libpng-1[1].2.2-21.i386.rpm
Libpjpeg-6b	libjpeg-6b-30[1].i386.rpm
Zlib	zlib-1[1].1.4-8.1.i386.rpm
ACID	Not available on RHN

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Out of the eleven required packages seven are available from RHN in RPM format. I chose to add the above packages through the RHN website. I also chose to install the mod_php package so that PHP would function with Apache and the mod_ssl package so that communication with ACID would be through an encrypted tunnel. Entering 'rhn_check' on the command line of my system established a connection with RHN and installed the selected packages.

The remainder of the packages (ADODB, jpgraph, Snort and ACID) were downloaded in *.tar.gz format and transferred securely to the server using my SCP/SFTP client. I then validated the integrity of the packages wherever checksums values were available from the various download sites:

```
# md5sum acid-0.9.6b23.tar.gz
d8c49614393fa05ac140de349f57e438
```

3.6 Building the System

3.6.1 Hardening the server

I scanned the server with Pedestal Software's Security Expressions program and corrected the issues it found. I then ran two nmap scans, TCP and UDP respectively, to verify that unneeded services were disabled:

```
# nmap –sS localhost

Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
Interesting ports on localhost.localdomain (127.0.0.1):
(The 1598 ports scanned but not shown below are in state: closed)
Port        State       Service
22/tcp      open        ssh
5555/tcp    open        freeciv    → used for Nagios/Cacti monitoring

# nmap -sU localhost

Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
Interesting ports on localhost (127.0.0.1):
(The 1460 ports scanned but not shown below are in state: closed)
Port        State       Service
161/udp     open        snmp
514/udp     open        syslog
```


© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

3.6.2 Installing/Configuring Snort

MySQL was already installed using RHN so I moved to unpacking Snort into '/opt' and compiled it with support for MySQL (--with-mysql):

```
# tar zxvf /usr/local/downloads/snort-2.1.2.tar.gz -C /opt
# cd /opt
# ln -s snort-2.1.2/ snort
# cd /snort
# ./configure --with-mysql && make && make install
# mkdir /var/log/snort
```

I ran the last command in the above sequence to create a snort directory in '/var/log/'. Snort would have failed to run and resulting in an error. In addition, a database and several tables were created in MySQL for Snort to deposit alerts into the database. Snort included a `'create_mysql'` file which layed out the database schema. This file is located in the install directory of Snort within the '/contrib' directory

```
# mysql –u root
# mysql> create database snort;
# mysql> exit

# cd /opt/snort/contrib

# mysql –u root snort < create_mysql
```

I verified that the tables were created successfully. Below is an abbreviated view of the output

```
# mysql –u root snort

mysql> show tables;
+------------------+
| Tables_in_snort  |
+------------------+
| data             |
| …
| …
| udphdr           |
+------------------+
16 rows in set (0.00 sec)
```

I choose to create the user 'ids' that Snort would use to connect to MySQL and granted the appropriate permissions for this user on the snort database. Using some guidance from Guy Bruneau's "Installation Instructions for Acid", I did the following: (http://www.whitehats.ca/main/members/Seeker/seeker_acid_mysql/seeker_acid_mysql.pdf)

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

```
# mysql –u root snort
mysql> delete from mysql.user where user='';
mysql> delete from mysql.user where password='';
mysql> grant all privileges on *.* to root@localhost \
       identified by 'password_of_choice' with grant option;
mysql> grant insert,update,select,delete on snort.* to \
       ids@localhost identified by 'different_password_of_choice';
mysql> exit
```

Some default values needed to be changed for Snort to be effective and to communicate with MySQL. In addition, one of the requirements of the IDS solution was real-time alerting. The Pigsentry program fulfilled this requirement. Pigsentry, as mentioned above, works by inspecting the alert logging file of Snort and uses predefined threshold values to decide when to email administrators. On the other hand, ACID requires that Snort events be deposited into a database. For this unique set-up, I maintained an alert file and an alert database. This required me to keep a watchful eye on the disk utilization so I requested for our System Administrators to setup this server to be monitored with our management software, Cacti & Nagios.

Using the vi text editor, I modified the 'snort.conf' file that is located in the 'etc/' folder of the Snort installation directory.

```
# vi /opt/snort/etc/snort.conf
```

For my configuration, the following values needed to be changed:

Networks x…, y…, & z… are examples of class C network blocks.

```
var HOME_NET [x.x.x.0/24,y.y.y.0/24,z.z.z.0/24]
```

I did not modify the following value so that I would not miss any external addresses.

```
var EXTERNAL_NET any
```

I entered Nagios/Cacti ranges which are ABC Inc's SNMP Servers. Failing to configure this value would generate a 'SNMP request udp' every time Nagios or Cacti SNMP polled a server.

```
var SNMP_SERVERS a.a.a.0/24
```

I then configured Snort to send alerts to MySQL by locating the Output section and updating accordingly. The alert file in /var/log/snort/ also needed to be maintained for Pigsentry so two output directives were required:

```
output database: alert, mysql, user=ids password=xxxx dbname=snort
output alert_full: /var/log/snort/alert
```


© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Note: I tried to use the output module alert_fast instead of alert_full in order to generate less logs, however, I discovered that Pigsentry does not function properly unless using alert_full.

For Snort to inspect all network packets, the interfaces needed to be configured in promiscuous mode. To do this quickly from the command line using the ifconfig command:

`# ifconfig ethX promisc` → Where 'ethX' is the interface of choice

The other preferred method and maintained over system reboots is in the configuration file for the interface. The following instructions are for a RedHat ES server:

```
# vi /etc/sysconfig/network-scripts/ifcfg-ethX

DEVICE=ethX
ONBOOT=yes
BOOTPROTO=PROMISC
```

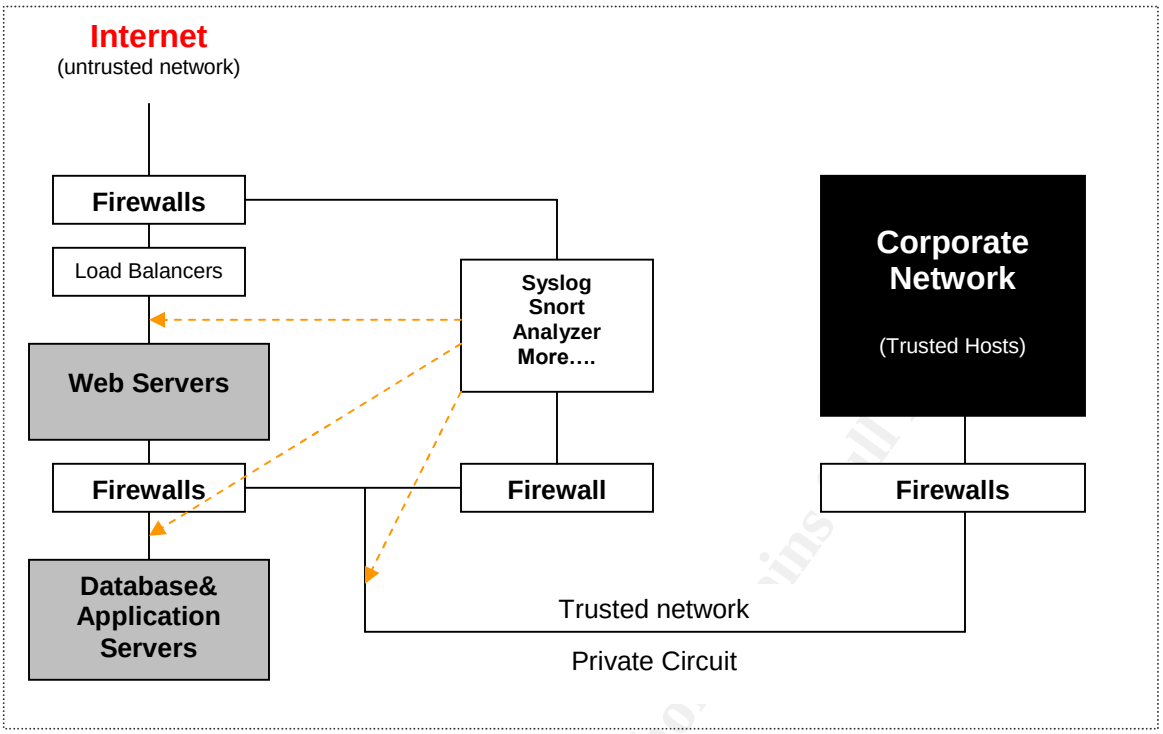
Where ethX was the interface that was to be used as a Snort sensor. I then ran the command '`ifdown ethX`' followed by '`ifup ethX`' that brought the interface up with no bound IP address and it began running in promiscuous mode. In order to test proper configuration of the interface I typed '`ifconfig ethX`' and examined the output (abbreviated view of output below):

```
# ifconfig ethX
ethX      Link encap:Ethernet  HWaddr 00:02:55:..:..:..
          UP BROADCAST RUNNING PROMISC MULTICAST  MTU:1500
          RX packets:71487709 errors:0 dropped:0 overruns:0
          TX packets:569156 errors:0 dropped:0 overruns:0
          collisions:0 txqueuelen:1000
          RX bytes:1625902015 (1550.5 Mb)TX bytes:32863647
          Interrupt:25 Base address:0x2240 Memory:feb7e000-
```

I have multiple interfaces on my system so I had to perform the above action to three of them (eth1, eth2, and eth3), as they were to act as Snort Sensors. On eth0 I left an IP address so that I could continue to connect and administer the server.

There are several methods for Snort to examine network traffic. These include the use of a Hub, a Network Tap or port monitoring on a Switch, among others. One of the requirements for the IDS deployment was to set-up the system to be un-disruptive to existing network traffic. For this reason I chose to use the Switch port monitoring because it is effective and would cause no downtime. I chose to span only the load balancer or firewall ports on the switch. Monitoring these ports is ideal because they act as gateways for network traffic in and out of their respective network segments. In the case of the trusted network segment, as seen in the diagram below, I spanned two firewalls ports that had a combined peak traffic rate of 15Mbs. This an important point because if the combined traffic had been higher then 100 Mbs, the port configured to do port monitoring would be overwhelmed and would drop packets.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.



I started Snort for the first time on each of the interfaces designated as sensors

```
# snort –c /opt/snort/etc/snort.conf –i ethX &
```

It is possible to maintain separate snort.conf files for each of the interfaces however I chose to use the same file for the time being. When events occurred, there were entries in the database and in the Snort alert file. To check the database if events occurred I ran:

```
# mysql -u ids -p snort

mysql> select count(*) from event;
+----------+
| count(*) |
+----------+
|        1 |
+----------+
1 row in set (0.00 sec)
```

I also ensured that entries were being sent to the alert file by checking that the file had a file size greater then 0 bytes

```
# ll /var/log/snort/alert
-rw-------  root root   1677 Jun 01 18:08 /var/log/snort/alert
```


© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

3.6.3 Installing/Configuring ACID

Installing and configuring ACID takes only a moment.

```
# cp acid-0.9.6b23.tar.gz /var/www/html/
# cd /var/www/html
# tar zxvf acid-0.9.6b23.tar.gz
# ls     → make sure that the acid directory is created
```

ACID had dependencies on additional programs in order to properly function. In the following steps I installed ADODB, jpgraph and made configuration changes to ACID:

ADODB

```
# cp adodb422.gz /var/www/html
# cd /var/www/html
# tar zxvf adodb422.gz
```

JPGraph

```
# cp jpgraph-1.13.tar.gz /var/www/html
# cd /var/www/html
# tar zxvf jpgraph-1.13.tar.gz
# ln –s jpgraph-1.16 jpgraph
```

ACID

```
# vi /var/www/html/acid/acid_conf.php
```

I set the following options with the relevant information for ACID to function properly.

```
$Dblib_path="/jpgraph/src";
$chart_file_format = "png";

$alert_dbname   = "snort";
$alet_host     = "localhost";
$alert_port     = "";
$alert_user     = "ids";
$alert_password = "xxxx";
```

I checked that Apache was running by typing '`ps –ef`' on the command line looking for the several `/usr/sbin/httpd -k start` processes. I then created a one-line file in the root HTML directory of Apache to validate that PHP pages displayed properly:

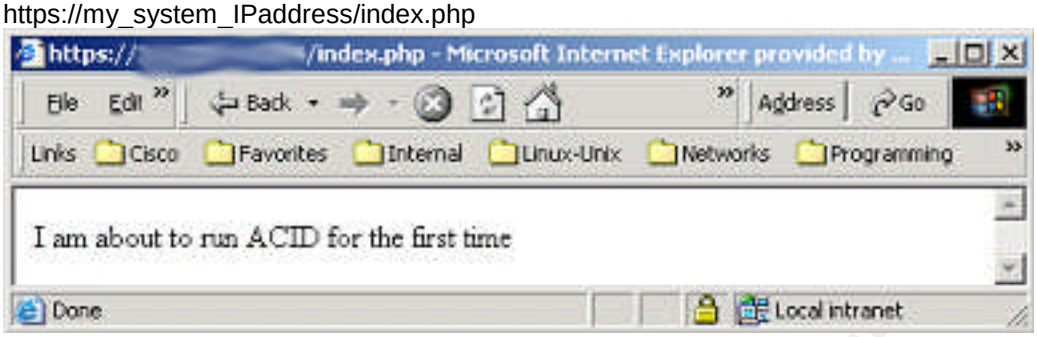
```
# vi /var/www/html/index.php

<?php echo "I am about to run ACID for the first time";?>
```

I had previously configured SSH (TCP port 22) on the firewall to allow Secure Shell access to the system for configuration and then added an additional rule to allow

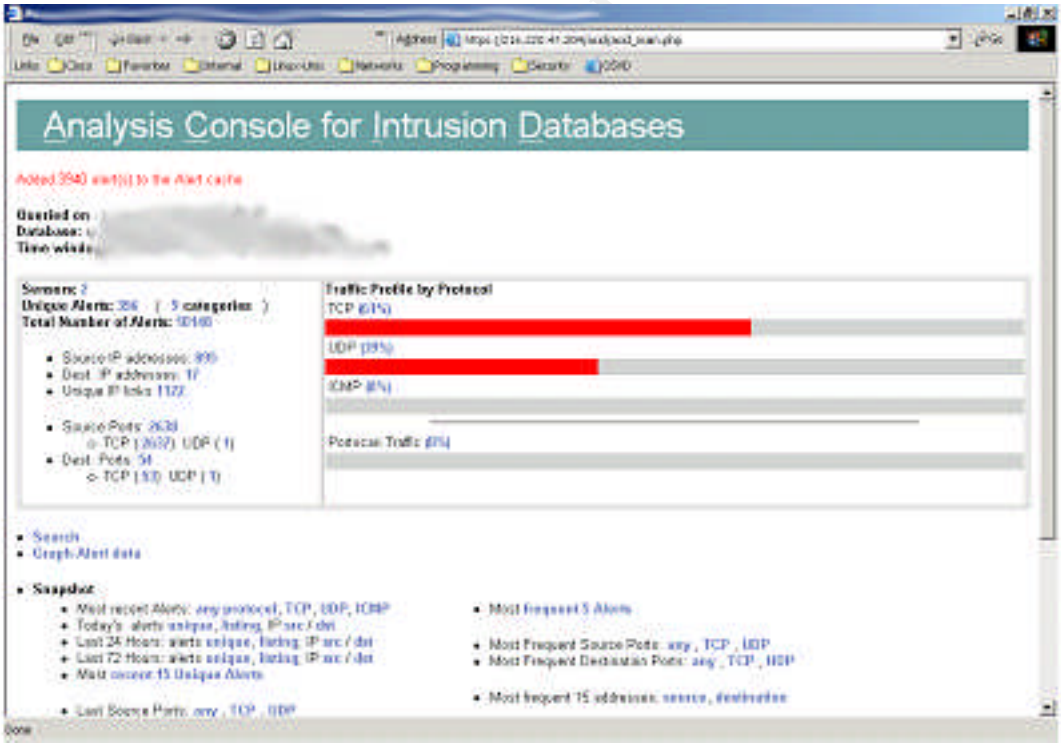
© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

HTTPS access (TCP port 443). I choose only these two ports to ensure that all communication with the server was using encrypted sessions. I pointed my Web Browser to the IP address of my system to view the test page:

https://my_system_IPaddress/index.php



I then pointed my Web Browser to the ACID directory at: https://my_system_IPaddress/acid/ and was prompted with a page indicating that ACID specific tables needed to be created. After clicking on the 'Setup page' link I was brought to a database configuration page that did the creation of the ACID tables. I clicked on the "Create ACID AG" button and a new page was presented, notifying that the tables were successfully created. I clicked the 'Home' link on the top right corner of the web page to have my first view of the ACID GUI:



I allowed several alerts to accumulate for a couple of days and then created Alert Groups (AG) to organize alerts into logical groupings. Roman Danyliw's description of AG is:

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Alerts groups (AG) are a method by which alerts can be logically grouped. For example, an AG can be used to associate multiple alerts that compose an incident or to assign priority. Likewise, AG is currently the only way annotation can be done on a single or multiple alerts. AGs are supported by extending the basic Snort database structure. (Danyliw)

3.6.4 Installing/Configuring Pigsentry

Pigsentry, a program available through the Snort download site was designed to alert by email without overwhelming the inbox of the receiver. It sends an email each time a new alert occurs but then maintains a state table thereafter sending further emails only when a trend increase occurs on existing alerts. For example, after having received the initial email of a new alert, no additional email would be sent regarding that particular event unless the occurrence of it has risen to a predefined threshold. Many values are configurable for Pigsentry including the threshold triggers values:

```
--warn-throttle    # How long since last trend alert, before sending a new one # default=12
--state-expire     # how long does and alert stays in the state table # default=24 hours
--poll          # How many seconds between trend intervals # default=300
--retention         # How many intervals to keep in trend stretching # default=12
--threshold-alert # Alert on a % spike over average # default=10
```

The only adjustment required was to the last value above because Pigsentry was designed to run on Snort sensors with 200,000 – 300,000 alerts per day. According to 'Detection Time' graph in ACID, I average ~350 alerts per day. This is due to the positioning of the Snort sensors to reside on the protected side of ABC Inc firewalls. The firewalls are configured to allow only a limited amount of ports from the Internet keeping the amount of alerts to a minimum. I found that the most of the configurable values suited my needs except for the threshold-alert value that I set to 5% instead of the default 10%.

Downloading Pigsentry version 1.2 consisted of copying the 608 lines of Perl code, including comments, from the Snort website and pasting it into a file. I reviewed the code for malicious commands that might cause information leaks such as FTP or email of our events to an external party but I found that the code appeared clean. To ensure that Pigsentry would run as an unprivileged user I used the sudo command.

```
# sudo –b –u pigsentry /opt/snort/Pigsentry –l /var/log/snort/alert \
  –m my_email@ABCinc.com --threshold-alert=5
```

Below is a sample of an alert email that I received after starting Pigsentry:

From: pigsentry@snort-sensor.com
To: administrator@ABCinc.com
Subject: PigSentry alert: New event: (http_inspect) BARE BYTE UNICODE ENCODING

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

3.6.5 Configuring Syslog

There are many reasons for using a centralized Syslog server to enhance ABC Inc's security position. A Hacker can easily modify logs on a server they have compromised to cover their tracks. Sending logs to another server allows for cross-reference of log entries and make it more difficult for an intruder to cleanse their actions. Having a centralized logging server allows for the consolidation of information to assist in troubleshooting issues, for analysis of system events and auditing.

Syslogd is installed on most Unix/Linux systems. The centralized syslog server, as mentioned above, is Linux RedHat ES and already had syslogd installed. The default installation for RedHat does not allow remote hosts to log to the syslog service. Simple configuration changes enabled remote logging and helped organize the logs into logical files.

I first defined the hosts that will be logging to the server in '`/etc/hosts/`':

```
192.168.x.1         PIX-01
192.168.y.1         LB-01
192.168.z.1         Iptbl-01
192.168.y.100       Web-01
192.168.z.100       DB-01
```

Then I modified the '`/etc/sysconfig/syslog`' file:

```
SYSLOGD_OPTIONS="-m 0 -r -l PIX-01:LB-01:Iptbl-01:Web-01:DB-01"
```

Where '`-r`' enables logging from remote hosts and '`-l`' enables logging with simple names instead of their FQDN.

The file '`/etc/syslog`' also needed to be modified to act as a traffic cop for the various log entries sent from remote hosts or Unix domain sockets. Below is a sample of some additional entries I added to this file

```
kern.*   /var/log/kern     # kernel and iptable msg's
local4.* /var/log/pix      # Cisco PIX entries
user.*   /var/log/ntsyslog # W2K SQL server
```

3.6.6 FWAnalog

Administrators of Internet facing firewalls can attest to the enormous amount of denied messages generated. Analyzing these logs can provide tremendous insight by displaying the blocked attempts to your infrastructure. Depending on the level of logging configured on a firewall, even when tracking only denies/drop, it can generate several thousand lines of entries per day. I took a 7 day average of our Internet facing Cisco PIX, configured with the logging level of warning, and found that it had approximately 50,000 entries per day while protecting a /27 network (30 usable Public IP addresses). For this reason, analyzing the logs manually would have been

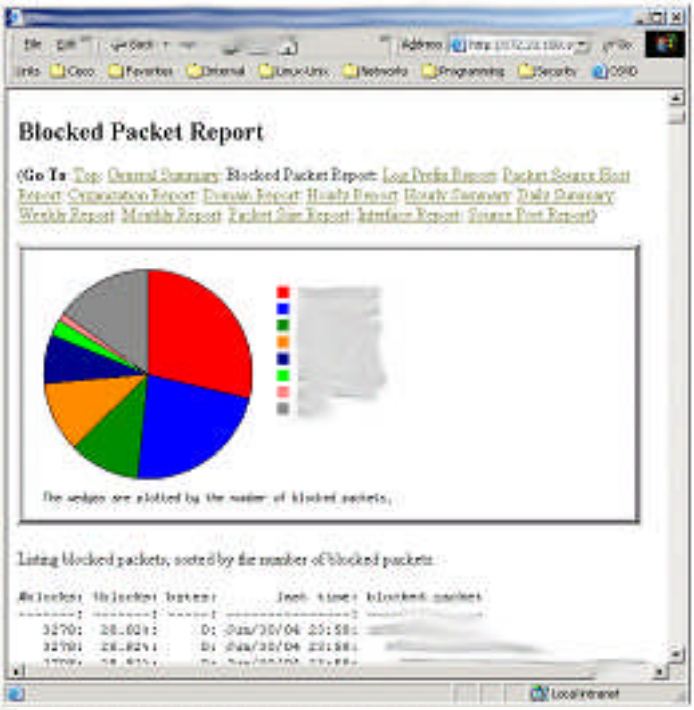
© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

tremendously difficult and a log analysis tool was required in order to maximize the return to my working hours.

I had previously installed the open-source tool Fwanalog on a different system, a shell script that parses and then summarizes the log files generated by Cisco PIX, IPF, Linux 2.2 ipchains, Linux 2.4 iptables, Watchguard Firebox or Firewall-One Checkpoint. After configuring each of the firewall to send their syslog messages to the central log server, I had the ability to process the logs for trends and to assist in investigating IDS logs.

Using a scp script and a daily cron job, I transferred the compressed log files to a dedicated log analysis server. The purpose of this server was to collect files from multiple locations and generate HTML reports for easy analysis. The sample report shown below was generated for an ABC Inc firewall's daily activity:

Sample fwanalog report



© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

3.6.7 Installing/Configuring FISQ

Firewall SQL Import Script from Activeworx.org was available from their download site (http://www.activeworx.org/downloads/index.htm). It did not include documentation but after reviewing the three files in the package I quickly learned it contained a database schema, a README file and the executable file used to load the database.

```
# tar zxvf  fisq.v0.9.tar.gz
# cd fisq
# echo "create database fwlogs;" | mysql –u root –p
  Enter password: xxxx
# echo "grant all privileges on fwlogs.* to ids" | /
  mysql –u root –p
  Enter password: xxxx
# mysql –u ids –p fwlogs < fw.sql
```

To start fisq and have our firewall logs inserted into the 'fwlogs' database, I used the sudo command:

```
# chmod 705 /opt/fisq/fisq.pl

# sudo -u fisq /opt/fisq/fisq.pl –D –u ids –p xxxx –h localhost /
  -i /var/log/pix -t
```

By using the –t flag, the command above performed a `'tail –f'` on the Cisco PIX logs and the –D flag forked a process. This process was repeated for our iptables firewall logs.

```
# sudo -u fisq /opt/fisq/fisq.pl –D –u ids –p xxxx –h localhost /
-i /var/log/kern -t
```

This completed the set-up of the IDS and Centralized Logging repository and their contributing applications. ABC Inc, has now strengthened their defence strategy by mitigating the risks identified in the Before section of this paper. Each of the requirements have been met. The installation process was smooth, especially with the help of the RedHat Network, and the contributing documentation posted on the Snort.org website.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

4. After

4.1 Outline of Steps Taken

The following table summarizes the steps taken to meet the requirements.

Intrusion Detection System

Requirement	Solution	Impact
- Ideal positioning of Sensors	Port Monitoring of Load Balancers & Firewalls Ports	Ability to monitor traffic in and out of network segments
- Real Time Alerting	Pigsentry	Ability to be alerted as events occur
- User-friendly GUI	ACID	Greater visibility of the alert database
- Event correlation	ACID	Feature of GUI
- Easy management of alerts	ACID-Alert Groups (AG)	Logical grouping of alerts
- Non-disruptive deployment	Switch Port Monitoring	Non disruptive set-up

Centralized Syslog

Requirement	Solution	Impact
- One consolidated repository	Syslogd configured to receive message from other hosts	Log file integrity and consolidation of all logs
- Ability to send firewall logs to a database	FISQ	Easy querying of database for offending host
- Process firewall logs for trend analysis	Fwanalog	Ability to process firewall logs to identify trends and assist in the investigation of IDS events
- Non disruptive changes to systems	Minor configuration to systems	Integrity of logs

4.2 Enhanced State of Security

Upon the deployment of Snort with ACID, I improved ABC Inc's security posture by providing a tool capable of inspecting each packet traveling in and out of the monitored network segments. We now have the visibility into our environment and can monitor for hacking attempts. In addition, we no longer have to rely on local system logs, after-the-fact logging entries, to get a picture of the activity occurring to a server as these logs may have been modified. Through the use of Pigsentry, our administrator team can be alerted real-time of an event occurrence.

An effective Intrusion Detection system must be cared for in order for its ongoing success. Upon installation of Snort, I wrote policies that cover the management and the regularity of required updates for Snort signatures. Otherwise, the Snort system would be quickly using outdated rules and would be ineffective. Also included in the policy is the regular classification of alerts into AG groups for logical grouping and to better manage alerts in the database. In the policies written, I stated that Snort rules must be

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

checked once per week and I scheduled recurring time in my calendar to investigate the latest version of these rules. If a vulnerability to a system or program employed by ABC Inc is identified through my subscription to various mailing lists, I check if the Snort rules have been updated immediately. I then apply the new rule set during the next available change window. At the beginning and end of every day, I also check the ACID console for new alerts and investigate where needed. I also leave time in my calendar to modify rules that might be causing a false positive.

A centralized Syslog server allows for the consolidation of all system logs into one location where analysis can be easily performed. This has greatly assisted in troubleshooting issues that have arisen with new systems having misconfigured network settings, lack of appropriate firewall rules, etc. I can also review the firewall log database if I need to view the activity of a particular offending IP address that has been identified by Snort as sending irregular traffic to an ABC Inc server. If they have performed a port scan, the firewall logs will indicate this. The usual starting point for querying the firewall log database is ACID, however if the daily HTML output of fwanalog indicates an increase in traffic to a particular port I will perform further investigation in order to get a better understanding of the possible risk.

Conclusion

The training provided during my SANS GSEC course provided me with the knowledge necessary to identify and mitigate the risk in ABC Inc's security posture. I was introduced to the concept of defense in-depth and how to take a layered approach to security. The course strengthened my ability in the areas of prevention techniques but also helped me gauge our security posture against industry best practices.

The actions taken during this case study have increased the general state of security for ABC Inc. I successfully installed an Intrusion Detection system that can monitor for unwanted activity on the network and a repository for all system logs in one secure location. The successful deployment of both solutions was accomplished with little impact to the existing infrastructure and with minimal cost.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

5. List of References

Cole, Eric; Fossen, Jason; Northcutt, Stephen and Pomeranz, Hal. Track 1 – SANS Security Essnetials and the CISSP 10 Domains, 1.2 Defense In-Depth. Reading: www.sans.org, 2004. pg 12-16.

Unknown. "Red Queen Hypothesis." URL: http://encyclopedia.thefreedictionary.com/Red%20Queen%20Hypothesis

Pedestal Software. "Security Expressions." URL: http://www.pedestalsoftware.com/products/se/

Unknown. "Welcome to Snort Download Centre". URL: http://www.snort.org/dl/ (5 July, 2004).

Danyliw, Roman. "ACID: Installation and Configuration". Analysis Console for Intrusion Detection. October 9, 2002. URL: http://www.andrew.cmu.edu/user/rdanyliw/snort/acid_config.html

Burneau, Guy. "Installation Instruction for ACID." Version 1. 1 October 2003. URL:http://www.whitehats.ca/main/members/Seeker/seeker_acid_mysql/seeker_acid_mysql.html

Danyliw, Roman. "ACID: Alert Group." Analysis Console for Intrusion Databases. URL: http://www.andrew.cmu.edu/user/rdanyliw/snort/acid_ag_instruct.html

Bárány, Balázs. "fwanalog." 18 March 2004. URL: http://tud.at/programm/fwanalog/

Activeworxs, Inc. "FISQ - Firewall-SQL Import Script." 27 June 2004. URL: http://www.activeworx.org/programs/fisq/index.htm (July 5 2005).

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Last Updated: May 1st, 2017

Upcoming Training

Click Here to
{Get CERTIFIED!}

SANS Riyadh 2017	Riyadh, Saudi Arabia	May 06, 2017 - May 11, 2017	Live Event
SANS Security West 2017	San Diego, CA	May 09, 2017 - May 18, 2017	Live Event
SANS Northern Virginia - Reston 2017	Reston, VA	May 21, 2017 - May 26, 2017	Live Event
SANS Melbourne 2017	Melbourne, Australia	May 22, 2017 - May 27, 2017	Live Event
SANS London May 2017	London, United Kingdom	May 22, 2017 - May 27, 2017	Live Event
SANS Stockholm 2017	Stockholm, Sweden	May 29, 2017 - Jun 03, 2017	Live Event
SANS San Francisco Summer 2017	San Francisco, CA	Jun 05, 2017 - Jun 10, 2017	Live Event
Security Operations Center Summit & Training	Washington, DC	Jun 05, 2017 - Jun 12, 2017	Live Event
Community SANS Boise SEC401	Boise, ID	Jun 05, 2017 - Jun 10, 2017	Community SANS
SANS Houston 2017	Houston, TX	Jun 05, 2017 - Jun 10, 2017	Live Event
Community SANS Ottawa SEC401	Ottawa, ON	Jun 05, 2017 - Jun 10, 2017	Community SANS
SANS Rocky Mountain 2017	Denver, CO	Jun 12, 2017 - Jun 17, 2017	Live Event
Community SANS Portland SEC401	Portland, OR	Jun 12, 2017 - Jun 17, 2017	Community SANS
SANS Secure Europe 2017	Amsterdam, Netherlands	Jun 12, 2017 - Jun 20, 2017	Live Event
SANS Charlotte 2017	Charlotte, NC	Jun 12, 2017 - Jun 17, 2017	Live Event
SANS Rocky Mountain 2017 - SEC401: Security Essentials Bootcamp Style	Denver, CO	Jun 12, 2017 - Jun 17, 2017	vLive
Community SANS Philadelphia SEC401	Philadelphia, PA	Jun 19, 2017 - Jun 24, 2017	Community SANS
SANS Minneapolis 2017	Minneapolis, MN	Jun 19, 2017 - Jun 24, 2017	Live Event
SANS Columbia, MD 2017	Columbia, MD	Jun 26, 2017 - Jul 01, 2017	Live Event
SANS Cyber Defence Canberra 2017	Canberra, Australia	Jun 26, 2017 - Jul 08, 2017	Live Event
SANS Paris 2017	Paris, France	Jun 26, 2017 - Jul 01, 2017	Live Event
SANS London July 2017	London, United Kingdom	Jul 03, 2017 - Jul 08, 2017	Live Event
Cyber Defence Japan 2017	Tokyo, Japan	Jul 05, 2017 - Jul 15, 2017	Live Event
SANS Cyber Defence Singapore 2017	Singapore, Singapore	Jul 10, 2017 - Jul 15, 2017	Live Event
SANS Munich Summer 2017	Munich, Germany	Jul 10, 2017 - Jul 15, 2017	Live Event
Community SANS Minneapolis SEC401	Minneapolis, MN	Jul 10, 2017 - Jul 15, 2017	Community SANS
SANS Los Angeles - Long Beach 2017	Long Beach, CA	Jul 10, 2017 - Jul 15, 2017	Live Event
Community SANS Phoenix SEC401	Phoenix, AZ	Jul 10, 2017 - Jul 15, 2017	Community SANS
Mentor Session - SEC401	Ventura, CA	Jul 12, 2017 - Sep 13, 2017	Mentor
Mentor Session - SEC401	Macon, GA	Jul 12, 2017 - Aug 23, 2017	Mentor
Community SANS Colorado Springs SEC401	Colorado Springs, CO	Jul 17, 2017 - Jul 22, 2017	Community SANS