GIAC
CERTIFICATIONS

Global Information Assurance Certification Paper

Copyright SANS Institute
Author Retains Full Rights

Interested in learning more?

Check out the list of upcoming events offering
"Network Monitoring and Threat Detection In-Depth (Security 503)"
at http://www.giac.org/registration/gcia

Challenges of Managing an Intrusion Detection System (IDS) in the Enterprise

GCIA Gold Certification

Author: Russell Meyer

Adviser: Dominicus Adriyanto Hindarto

Accepted: March 24th 2008

Outline

1.   Introduction

While every enterprise is unique, there are common challenges in managing, monitoring and reacting to network IDS alerts. These include: managing the flood of alerts, creating actionable reports, and following-up on the reported alerts. This paper will explore the IDS challenges of a large organization with examples of specific lessons learned in monitoring the internal network.

2.   Background

The company referenced in this paper was a large healthcare organization with over 10,000 employees in 40 states and 200+ locations. Most of the servers were centralized but some locations had their own servers. As a healthcare company, the organization had some HIPAA Security Rule projects, including monitoring the IDS for HIPAA compliance (Beaver 2003) (Wilson 2007). The stated task was to implement IDS monitoring and reporting.

The Wide Area Network (WAN) was designed based on the hub and spoke model. Each site (spoke) was connected via a WAN link to the main site (hub), usually the local business office. Each business office was connected to the main data center via another WAN link, usually a T1 or better connection. The main data center was connected to the Internet via two T3 connections serviced by different carriers. The internal network was

Russell Meyer   3

separated from the Internet by firewalls but the internal network was not segregated. All

traffic going to or coming from the Internet traveled through the main data center.

There were about 8000 computers on the network, mostly workstation computers. The

organization used mostly Windows 2000 Professional on the desktops and had about 400 NT

4.0 & Windows 2000 servers. There were still some Windows 95 and 98 desktops and a few

UNIX and Linux servers. At the time, the organization was running an NT 4.0 domain. Most

of the hardware was managed centrally but there were some user-owned laptops and vendor-

provided and controlled systems that were used for patient care. These patient care systems

were maintained by the vendor as well as the local IT group.

The company had installed Cisco's Intrusion Detection System, one 4210 and one

4250 sensor. These all-in-one appliances were running hardened copies of Red Hat 7.2.

Cisco purchased their network based IDS solution, formally known as "NetRanger", from the

Wheel group in 1997 (Innella 2001). The 4210 sensor was monitoring traffic in the DMZ while

the 4250 was monitoring the internal network traffic passing through the main data center.

The organization used Cisco's Java based IDS Event Viewer (IEV) software to monitor their

two sensors and record those alerts.

Russell Meyer   4

3.   Managing the Flood of Alerts

A typical network IDS sensor in the enterprise generates a tremendous number of alerts. There are two major methods for detecting issues: signature detection and anomaly detection. Signature detection will generate an alert whenever traffic matches a known attack pattern while anomaly detection compares current traffic against "normal" traffic and generates an alert on the network. (Newman 2002) An alert does not necessarily mean the sensor found a problem, just that the sensor detected traffic that matched a signature or pattern. These alerts are called false positives and can overwhelm a sensor as well as the staff trying to monitor the sensor.

Russell Meyer   5
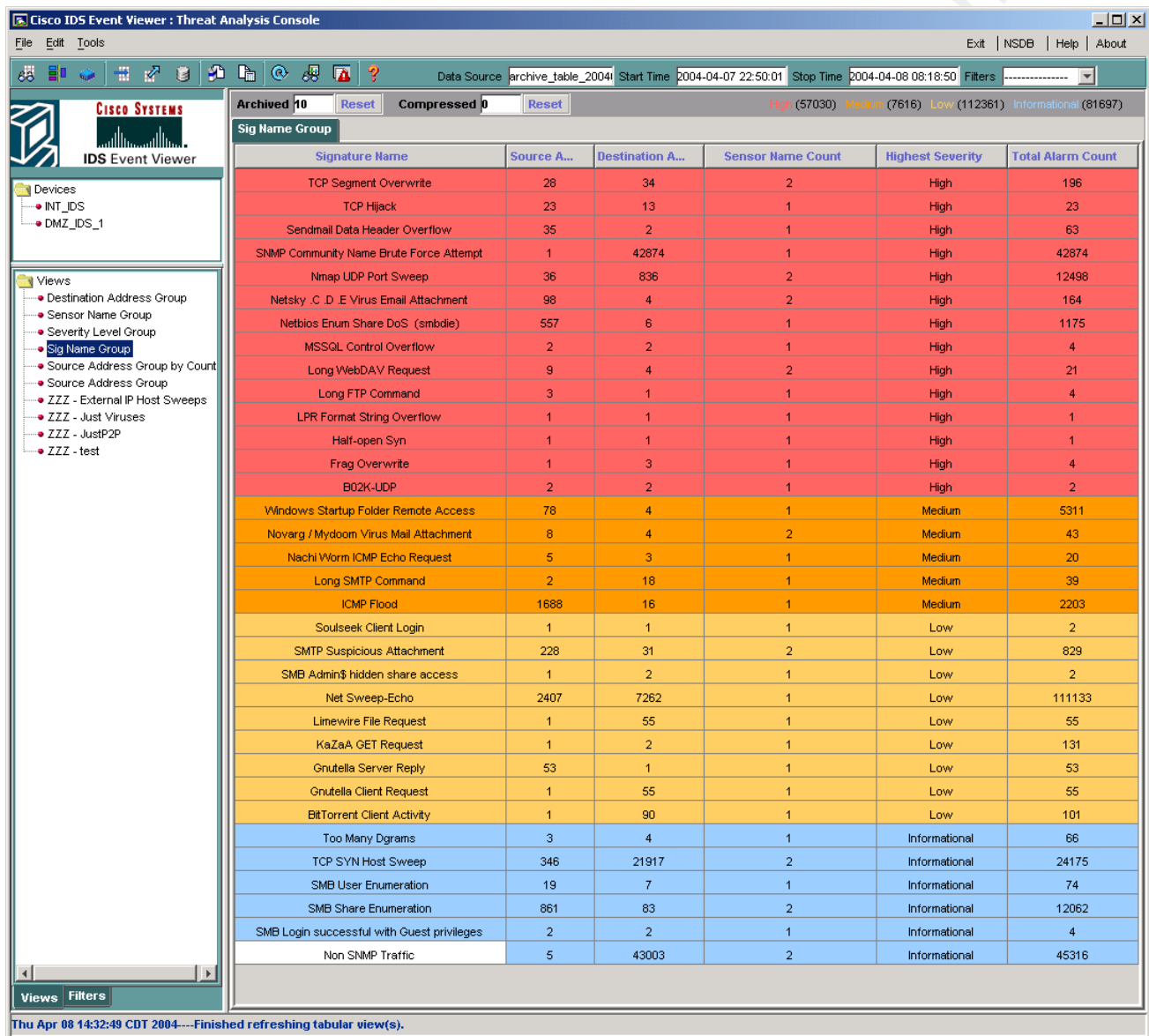
Alerts per Minute
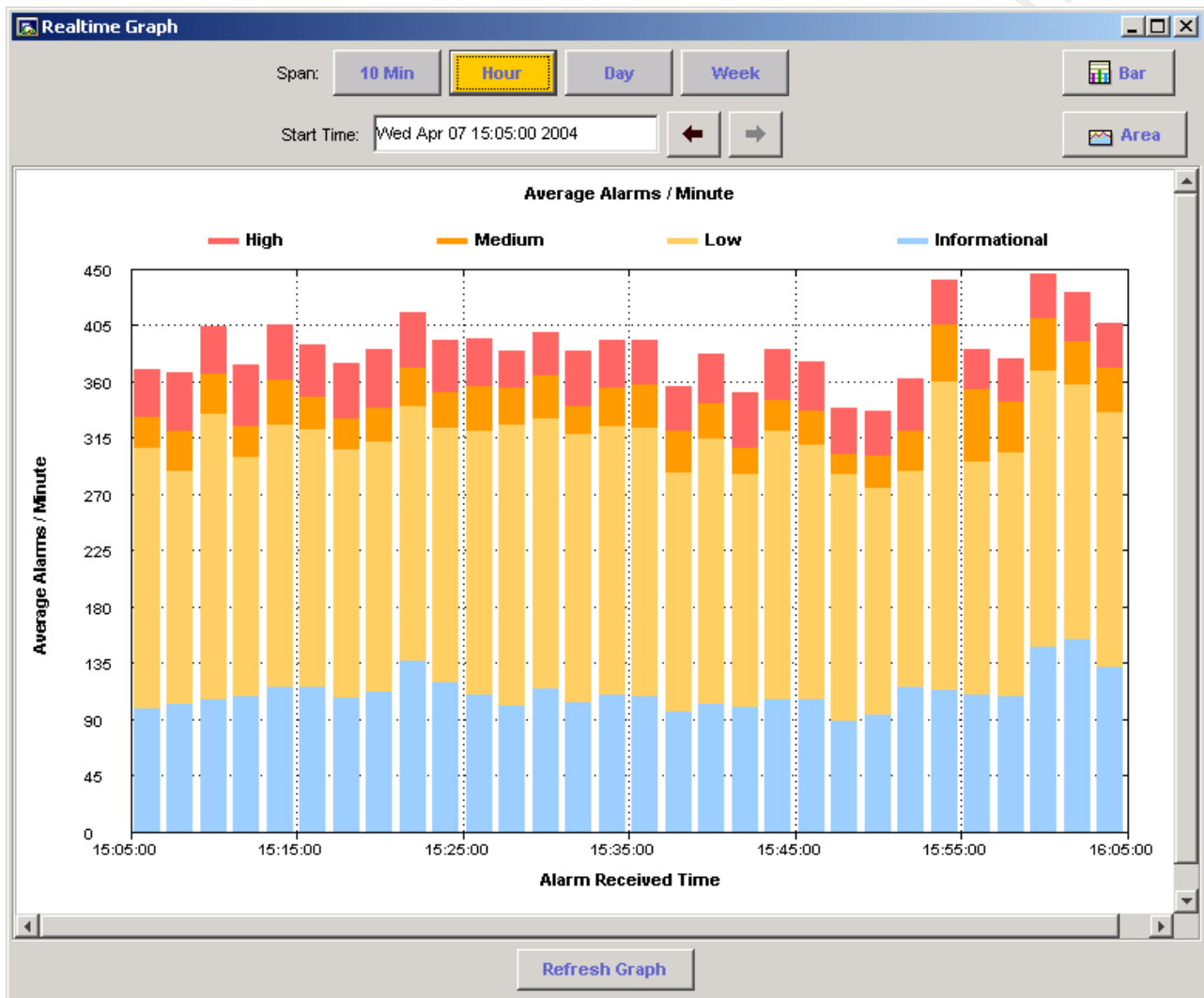
The above screen shows over 400 alerts a minute on the internal sensor and the

sensor averaged 400,000 alerts a day. This number of alerts overwhelmed the response

capabilities of the security group. The organization was not able to justify the staff to follow-

up on this number of alerts, especially since the probability was that most of the alerts were

false positives. (Axelsson 1999)

Russell Meyer   6

Alerts by Name

The network IDS sensors not only generated thousands of alerts an hour but hundreds

of different alerts with different severity levels. In the example above, red was the most

severe (highest risk) and blue was informational (lowest risk), according to the Cisco IEV

Russell Meyer   7

software. Only a few signatures were generating the majority of the alerts. These alerts were not only in the high risk category but also in the low risk category. All categories needed followed up.

Few organizations have the resources or inclination to correlate and follow-up on all of the alerts generated by the IDS network sensors. The typical commercial IDS sensor comes with a default set of enabled alerts and these enabled alerts can be tuned to filter out obvious false positives. For example, it is usually necessary to tune out the IT management systems that trigger alerts. The IT management system generated alerts would be considered false positives since the traffic generated by these systems is not malicious. One of the issues the organization faces with tuning is "How much to tune out?" If the sensor tunes out too many IP addresses or alerts, the organization risks missing real problems but if the sensor is not sufficiently tuned, the staff will spend needless time chasing false positives. This can lead to a loss of confidence in the capability of IDS to protect the network (Carter, 2002).

In the referenced company, the solution was to record all but the obvious false positives and any additional alerts that the organization had determined worthwhile. A filter was used to weed out all of the alerts except those that needed to be addressed immediately. This way, all of the alert data was captured and recorded for trending and archiving but only the most important alert data was acted upon.

Russell Meyer   8

A note about false negatives; false negatives are network traffic that the IDS sensors

do not catch, usually because there is no network signature or rule for the event. This

organization's IDS approach did not and could not address false negatives since it relied on

the IDS sensor to trigger the process to gather more information. If the sensor does not see

the event, an alert will never be generated and no additional information will be collected.

Just as an alert or two from the sensor does not mean there is an issue, the absence of alerts

does not mean there are no issues. An organization should never rely on just one method for

detecting problems. In order to make the most out of any organization's IDS, it is important to

use up-to-date IDS rules or signatures.

The IEV software included an export function that allowed the export of alert data to a

comma delimited (CSV) text file. Using the IEV the IDS sensor data was exported to a text

file for additional manipulation.

IEV Configuration

Russell Meyer   9
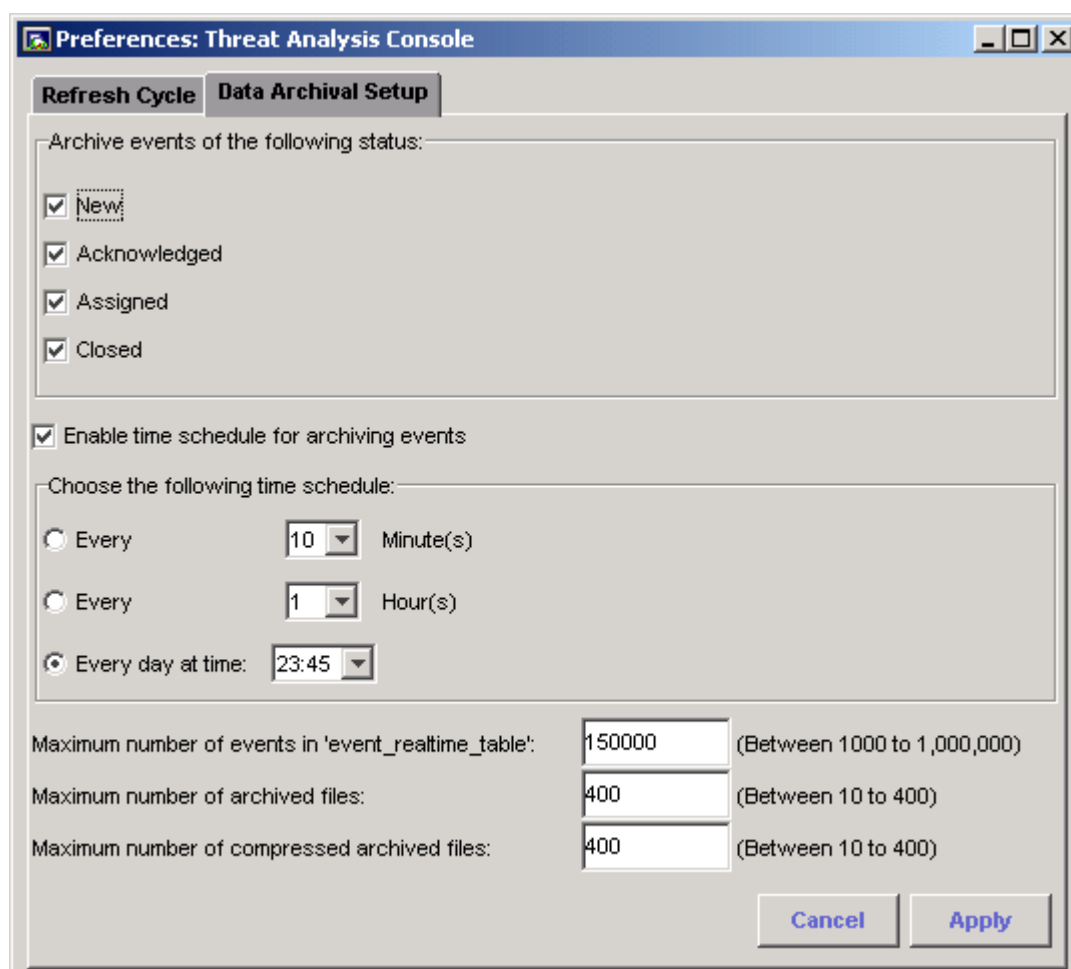
The IEV allowed the user to define how the alert data was archived. The above

configuration allowed for 150,000 alerts per file and archived the day at 11:45 each night.

150,000 alerts were usually enough for one day's worth of IDS sensor data.

IEV Alert Export
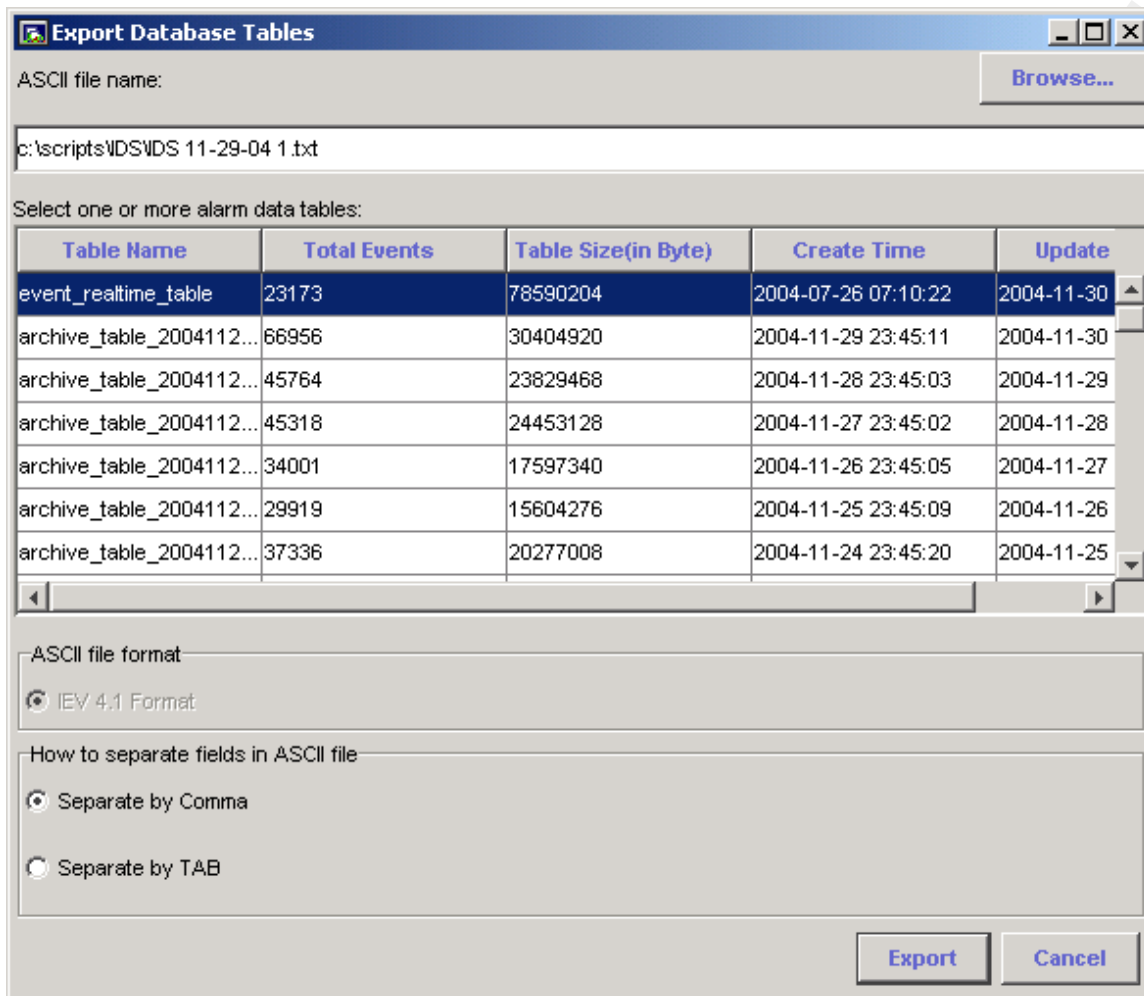
The export function of the IEV software allowed the alert data to be exported in either

common or tab separated format. In the example above, the IDS sensor data for 11-29-04

was exported to a file called "IDS 11-29-04 1.txt" in comma separated format.

Sample Exported Alert.

'5','1078879337720906009','3','INT_IDS','sensorApp','2004-08-31','11:00:29',' 1093946247725656000',
'1093949487725656000','3327','Windows RPC DCOM Overflow','0','\\\\\\\x26lt;400 chars>\\','S49','1',
'10.128.7.6','3965','OUT','10.86.26.240','135','OUT','(<10.128.7.6><3965><T1VU>?<10.86.26.240><135><T1VU
>)','2','0','Interval','true','0','0',\N,\N,\N,\N,\N,'Traffic Source: int2 ; Interval Summary: 2 alarms this interval
;','New',\N'5','1078879337720905988','3','INT_IDS','sensorApp','2004-08-31','11:00:14','1093946232394410000'
,'1093949472394410000','3327','Windows RPC DCOM Overflow','0','\\\\\\\x26lt;400 chars>\\','S49','1','10.128.7.6'
,'3965','OUT','10.86.26.240','135','OUT','(<10.128.7.6><3965><T1VU>?<10.86.26.240><135><T1VU>)','0',\N,\N,
\N,'0','0','AAAAAMAAAAAAAABGb3GrNJ430hGsDQAQWg0aQgUABgAAsAAAxAIAALaSv8+Rn3zQAAAAANAV
CAABEAgAzMzMzIAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAFihFAAAAAAAYAAAAGAAAABNRU9XBAAA
AMABAAAAAAAwAAAAAAAAEY7AwAAAAAAMAAAAAAAABGAAAAADAAAAABAAEAmlvwgMaEQ0uNwY/
GB/9RiQIAAAAAAAAAAAAAAAAAAAAAAAAAAAAEAAAABEAgAzMzMzEgAAAAAAAAAAAAAAHC/CwAA
AAAAAAAAADjvCwAAAAAAAAAAA8AAAAAAAADwAAAA==$TwBMAE8ARwBZAFwAcwBlAGEZwBhAHQ
AZQAAABAA//9VAFMATwBOAEMATwBMAE8ARwBZAFwAcwBlAGEZwBhAHQAZQAAAAkA//9VAFMATwBO
AEMATwBBMAE8ARwBZAFwAcwBlAGEZwBhAHQAZQAAABEA//9VAFMATwBOAEMATwBBMAE8ARwBZAFwAcwB
lAGEZwBhAHQAZQAAABIA//9VAFMATwBOAEMATwBBMAE8ARwBZAFwAcwBlAGEZwBhAHQAZQAAABIA//9VAFMATwBOAEMATwBBMAE8ARwBZAFwAcwBlAGEZwBhAHQAZQAAAAkA//9VAFMATwBOAEMATwBBMAE8ARwBZAFwAcwBlAGEZwBhAHQAZQAAABA//9VAFMATwBOAEMATwBBMAE8ARwBZAFwAcwBlAGEZwBhAHQAZQAAABA//9VAFMATwBOAEMATwBBMAE8ARwBZAFwAcwBlAGEZwBhAHQAZQAAABIA//9VAFMATwBOAEMATwBBMAE8ARwBZAFwAcwBlAGEZwBhAHQAZQAAABIA//9VAFMATwBOAEMATwBBMAE8ARwBZAFwAcwBlAGEZwBhAHQAZQAAABIA//9VAFMATwBOAEMATwBBMAE8ARwBZAFwAcwBlAGEZwBhAHQAZQAAAAkA//9VAFMATwBOAEMATwBBMAE8ARwBZAFwAcwBlAGEZwBhAHQAZQAAABA//9VAFMATwBOAEMATwBBMAE8ARwBZAFwAcwBlAGEZwBhAHQAZQAAABAAAAAAAAAAAAAAAAAAAAAAAAABMTUVNODELADhVCwAAAAAAAAAAoCAABQYgcAAQAAAAAAAAAA
AAAAAAAA==',\N,\N,\N,\N,'Traffic Source: int2 ;','New',\N

The above represents one exported alert. Cisco documentation provided descriptions

of the fields. While there was a lot of information about the alert itself, the alert did not

indicate much about the system or system user.

Sample Alert Fields

AlertLevel	Alert Level: 0= informational, 1=low, 2=med, 3=high Alert
SensorName	Name of the sensor, 'INT_IDS' or 'DMZ_IDS_1"
Date	Date of the alert
Time	Time of the alert
AlertID	The Cisco alert ID, i.e. "3030"
AlertName	The Cisco alert name, i.e. "TCP SYN Host Sweep"
SourceIP	The IP address of the system that generated the IDS alert
SourcePort	The port of the system that generated IDS alert
DestIP	The IP address of the destination traffic
DestPort	The port(s) of the destination traffic

After tuning out the biggest generators of false positives, there were still too many

alerts and not enough information on the remaining alerts. A decision was made to write a

program (Meyer 2008) which would parse the exported IDS alerts, capture the above fields,

filter out all but the alerts the organization wanted to pursue and finally capture additional data

on the remaining alerts. Cisco Systems also used scripts to help alert their IDS operators to

unusually large numbers of IDS alerts (Cisco 2003).

After parsing the data, the Perl program filtered out all of the alerts except those

designated for follow-up. This approach to IDS alerts addressed a problem encountered after

tuning the IDS sensor. Once the sensor was tuned to eliminate a false positive by IP address

or alert, that IP address or alert would never be seen again. That had the potential become a

Russell Meyer   13

problem. If a system is generating false positives today and the sensor is configured to

ignore that IP address or the alert from that IP, the sensor will continue to ignore that IP or IP

& alert combination. Even if the IP address is reallocated, the system is taken off line or the

alert is updated to eliminate the issue causing the false positive in the first place. After

several months, years or IDS engineers, that sensor would become blinded to real threats.

By moving most of the filtering process from the sensor to a post-alert process, the sensor

can concentrate on capturing and processing as many alerts as possible.

This was a nontraditional approach to dealing with false positives. By using the filter to

eliminate most of the alerts and thus most of the false positives the organization could focus

on just the alerts it deemed worthy of follow-up. The IDS still generated tens of thousands of

alerts each day but the staff concentrated on the clear-cut alerts that could be addressed

immediately. These included alerts for known worm traffic, Peer-to-Peer file sharing traffic

and spyware & Adware; all of which any security group could make a business case for

addressing.

This approach to IDS is not for every enterprise. If the enterprise network is already

well run and monitored, the systems on the network are kept patched and there are effective

Network Access Control (NAC) measures in place, this approach will not be of great benefit.

However, if the internal network is not monitored, the systems on the network are not as

secure or the organization simply does not have the resources to implement a traditional IDS

system, this approach could be a starting point. It is also important to remember that the Perl

program does not replace the IEV, it supplements it by providing actionable reporting on the

alerts the organization has deemed worthy of follow-up.

Flowchart - Filtering

Russell Meyer   15
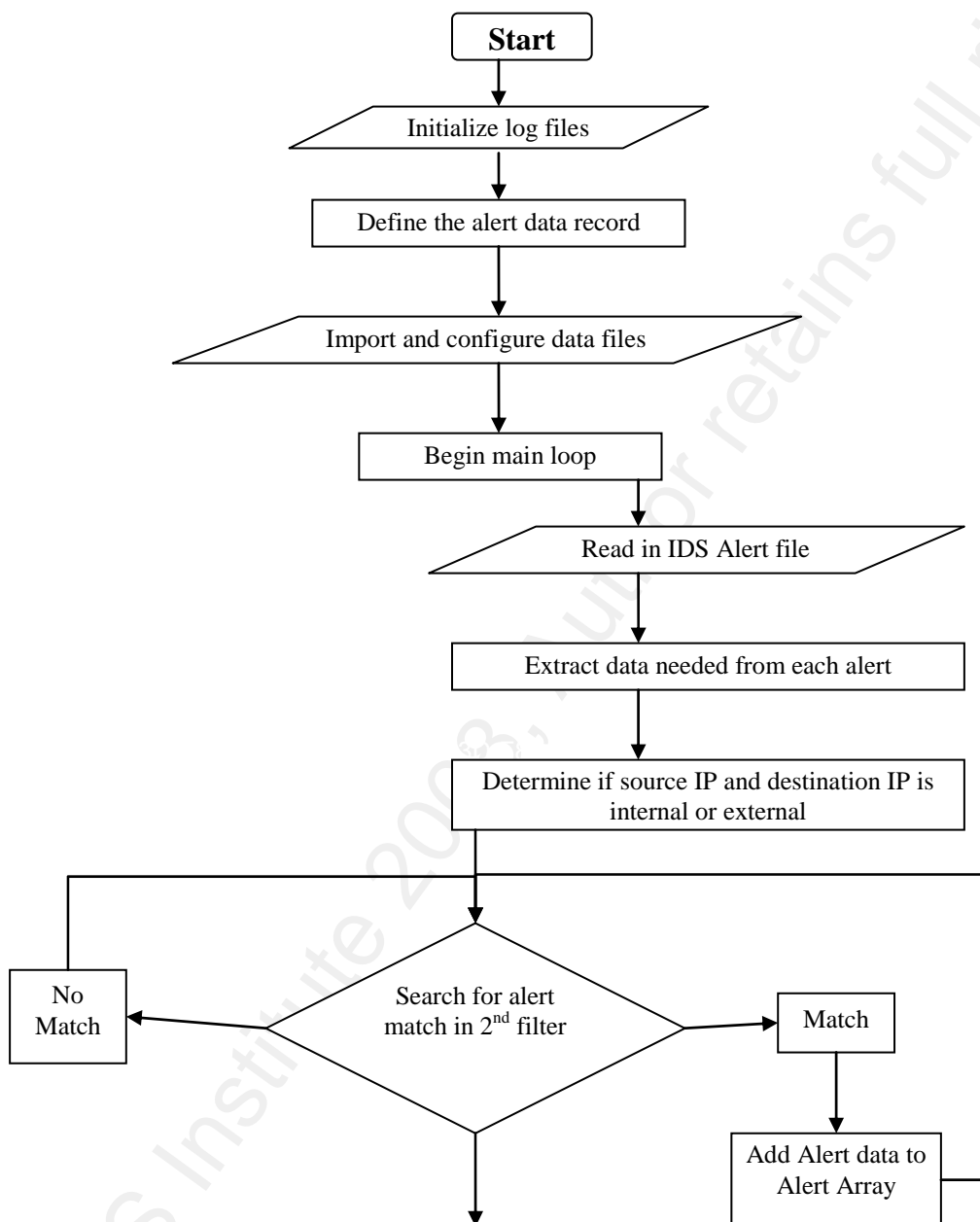
The flowchart illustrates how the program's filter works. From a high level perspective,

Russell Meyer   16

the first half of the program does the following:

- Initializes the program
- Processes the configuration and data files
- Begins the main loop and reads the first alert data file
- Extracts and records details of each alert
- Filters out all but the alerts the organization wants to deal with

Once the IDS has been running awhile and the organization has filtered the systems that generated the greatest number of false positives at the sensors, it is time to begin to build the list of alerts and alert criteria for the alerts the organization wants to address. The filter is built using an Excel spreadsheet to document the filter and then exported to a CSV file as input for the program. There are several fields that can be populated:

Russell Meyer   17

Filter Details

Field	Description	Example
Action-Filter ID	An internal ID number, starts at 1000	1016
Alert Name	The Cisco NSDB Alert Name	ICMP Network Sweep w/Echo
Alert ID	The Cisco NSDB Alert ID	2100
Alert level	The Cisco IDS Alert level	3 - Low
Category	Broad and simply defined alert categories (spyware, P2P, etc..)	Worm\Virus\Trojan
Action threshold	How many Alerts have to be triggered by IDS before the program reports it. '0' to filter event	10
Action to take	Action to take. Filter, investigate, etc…	Investigate
Source Network	Internal, External or DMZ Internal would be any IP that starts with 10, DMZ = 192, external IP anything else	Internal
Source IP Address	The IP address of the source of the alert, 'any' for any	Any
Source Port	The source TCP or UDP Port number of the alert	Any
Destination Network	Internal, External or DMZ Internal would be any IP that starts with 10, DMZ = 192, external IP anything else	External
Destination IP Address	The IP address of the destination of the alert, 'any' for any	Any
Destination Port	The destination TCP or UDP Port number of the alert	Any
Notes	Filter notes, free text, not used tin the reports. Can be used to document reason for filter	Notes here

In the example above, Cisco alert #2100 "ICMP Network Sweep w/Echo" could be

evidence of worm activity or network management activity. Cisco classified the alert as "low"

but if the alert is evidence of a worm, the organization would want know about it.   The "Action

threshold" is set to "10" and the "Destination network" is set to "External". This means the

alert will not be reported unless there are 10 alerts and the destination network is external (i.e.

Russell Meyer   18

the Internet). This should eliminate network management activity since the network

management tools should not be scanning external network addresses, i.e. the Internet.

In developing the filter list, take into consideration the resources the organization has

for IDS alert follow-up. While many of the alerts could justify follow-up from a pure security

standpoint, the organization may not be able to justify the resources, at least not in the

beginning. It is better to start with clear-cut alerts that are causing the most problems. Care

should be taken in determining the criteria to filter an alert. Threshold values need to be

determined and justified. It is too easy to just filter an IP address or a range of IP addresses

instead of just the false positive alerts they are generating. As time goes on, the filter list

should be refined based on alert follow-up feedback in order to produce the most accurate

reports.

Once the benefits of following up on the alerts were established, the organization

expanded the scope of the alerts to be investigated. Use of the historical data collected

pointed out that there were other issues that needed to be addressed. These new alerts were

addressed through informal channels at first in order to demonstrate the value of the alerts

and to unofficially verify their value for follow-up.

There were no easy solutions to managing the flood of network IDS alerts. The filter

approach allowed the sensors to generate the largest number of alerts while filtering out all

Russell Meyer   19

but the alerts the organization chose to address.

4.   Creating Actionable Reports for Follow-up

The second challenge that many organizations have is creating an actionable report on

the IDS alerts of interest for follow up. The IDS sensors generate little alert data to go on;

typically just the name of the alert, IP addresses, ports used and date and time information.

In a large organization, just having the IDS alert data is too little information to justify an

investigation or follow-up. Additional information is needed.

Once the enterprise has determined the alerts and alert criteria to add to the alert filter,

additional information about the systems should be collected to produce a report with enough

detail to be actionable. This report should correlate the IDS alert data with network, user and

alert source system data to produce a better picture of the event. The report should include

enough data that would allow a system owner or IT staff to find the system and help identify

the issue.

Flowchart - Reporting

```
┌──────────────────┐              ┌──────────────────┐
│  Return to start │              │  Continued...    │
│  of main loop    │              └──────────────────┘
└──────────────────┘                       │
         ▲                                  ▼
         │              Total alerts and generate Alert
         │                    Summary report
```
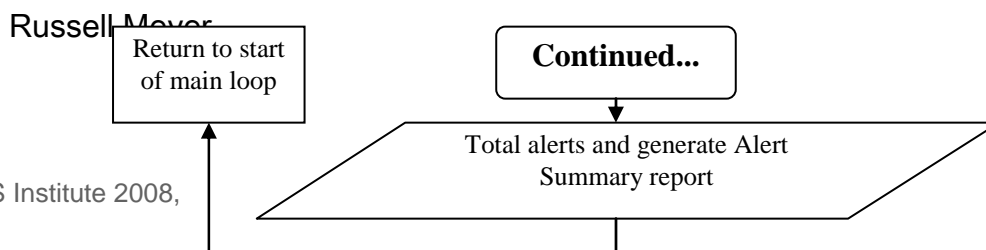
From a high level perspective, the report generating part of the Perl program does the

following:

- Summary report data is generated for all of the alerts
- Begin loop for alerts not filtered
- Gather more data about each alert
- Output actionable data details
- End loop for alerts not filtered
- End main loop
- Output summary matrix data
- End program

While follow-up details required will be unique to each organization, the referenced

organization gathered the following additional data for each alert that needed follow-up.

Russell Meyer   21

Additional Alert Details

Field	Description
Name of the IDS alert	The Cisco name for the IDS alert
IP Address	The host IP address; the host that generated the alert
Host Name	Name of the host, IF it can be resolved (may no longer be on the network)
Source Port	The source port used on the host workstation; where the traffic came from
Destination Port	The port of destination, where the traffic was directed to
Total Alerts	The total number of alerts for this host
Date of Last Alert	Date of the last alert activity
Time of Last Alert	Time of the last alert activity on the host (CST
Category	I defined several categories to help classify the alerts. These included: virus, Spyware, P2P, etc…
Internal or External	It the host targeting an internal (10. and 192.) or external (i.e.Internet) IP address.
User logged into host at time report was run	If available, what user was logged on WHEN the IDS report was run. If the alert is old or the report is rerun with an old exported IEV data file, this information may not be accurate.
NT User Comment Field	The User comment field, from the User's domain account
Date and Time of Last Login	Date and time the user last logged in
MAC Address of Host	The MAC address of the Host
Network Card Manufacture	Company that has registered the MAC address; helpful in tracking down the workstation (for example, if the MAC is registered to Dell, you would look for a
	Dell computer.)
Host OS (based on TTL)	You can usually identify the operating system via the TTL (packet Time-To-Live value)
Site Name	From a lookup table based on the first 3 octets of the IP address. Used to identify the location of the system
Site Contact	The person responsible for the site. This person may or may not be the system owner or custodian but they would know who to contact

Russell Meyer   22

Following is an example of an IDS report generated on 2/25/05, focusing on just one

host with five alerts.

IDS report

Item	IP Address	Host Name	Source Port	Destination Port	Total Alerts
IRC Channel Join	10.128.9.57	CORP01LT802	1158	6667	1
TCP SYN Host Sweep	10.128.9.57	CORP01LT802	1197	445	130
Windows LSASS RPC Overflow	10.128.9.57	CORP01LT802	1473	445	2
Windows RPC DCOM Overflow	10.128.9.57	CORP01LT802	1181	135	6
Windows SMB/RPC NoOp Sled	10.128.9.57	CORP01LT802	1181	135	8

Date of last alert	Time of last Alert	Category	User logged into host at time report was run	NT User comment field
2/25/2005	13:10:07	Instant Messenger and more	John Smith	101102 Business Office Clerk, Corporate
2/25/2005	13:22:12	Worm\Virus\Trojan	John Smith	101102 Business Office Clerk, Corporate
2/25/2005	13:17:05	Worm\Virus\Trojan	John Smith	101102 Business Office Clerk, Corporate
2/25/2005	13:20:03	Worm\Virus\Trojan	John Smith	101102 Business Office Clerk, Corporate
2/25/2005	13:20:03	Worm\Virus\Trojan	John Smith	101102 Business Office Clerk, Corporate

Date and time of last login	MAC Address of Host	Network Card Manufacture	Host OS (based on TTL)
2/25/2005 12:51	00-D0-59-D9-2A-84	AMBIT MICROSYSTEMS CORP.	NT-2000-XP (126)
2/25/2005 12:51	00-D0-59-D9-2A-84	AMBIT MICROSYSTEMS CORP.	NT-2000-XP (126)
2/25/2005 12:51	00-D0-59-D9-2A-84	AMBIT MICROSYSTEMS CORP.	NT-2000-XP (126)
2/25/2005 12:51	00-D0-59-D9-2A-84	AMBIT MICROSYSTEMS CORP.	NT-2000-XP (126)
2/25/2005 12:51	00-D0-59-D9-2A-84	AMBIT MICROSYSTEMS CORP.	NT-2000-XP (126)

Russell Meyer 23

External or Internal	Site name (as identified by first 3 octets of the IP address)	Site Contact
External	Main data Center (VPN)	Robert Smith
External	Main data Center (VPN)	Robert Smith
External	Main data Center (VPN)	Robert Smith
External	Main data Center (VPN)	Robert Smith
External	Main data Center (VPN)	Robert Smith

The above Perl report was generated as a CSV file. The data was reviewed and

added to an Excel template formatted using legal size paper in order to include all the alert

details on one line. Details of the above example IDS report show how the additional

information was generated.

Items - The IDS sensors generated five alerts for this host. The last three alerts were

named after well-known Windows vulnerabilities. These alerts were an issue for non-

managed or recently re-imaged workstations. If the workstation was not updated before a

worm found it, the system had the potential to become infected.

IP Address - The IP address indicated the source of the traffic that generated the alert.

Most organizations use Dynamic Host Configuration Protocol (DHCP) to assign workstations'

IP addresses, so care must be taken when tracking down systems based on old IDS reports

since that IP address could now belong to another system.

Special care should to be taken with VPN assigned IP addresses since the VPN IP

Russell Meyer   24

addresses are often recycled and used again. The rapid reuse of IP addresses is found with

VPN users and other organizations with sites that have a limited number of IP addresses. In

this case, the DHCP lease was three days and the DHCP server usually renewed desktops

with the same IP. This meant that users could use the same IP address for months. The IDS

report does not prove a workstation is infected, just that a certain IP address triggered an IDS

alert at a certain date and time. Further investigation would be needed to determine the

problem.

Host Name - In this example, "LT" identified the system as a laptop. A laptop user

outside the company was using the VPN (Virtual Private Network) to access the internal

company network. CORP001 told us the laptop probably belonged to a user from Corporate,

or at the very least, was configured for a user at Corporate. If the IP address was no longer

on the network when the report was generated, a lookup table based on IP address and

system names was used to determine the hostname. These cases are noted were a "*" after

the name.

Source Port – The source ports were usually ports above 1023. This was the case in

this example.

Destination Port(s) - The destination port(s) may help identify the program or system.

For the IRC alert, the port was 6667, a common IRC port. Ports 445 and 135 were used by

Russell Meyer   25

Windows hosts for Windows-to-Windows communications, which included RPC (Remote

Procedure Call) and DCOM (Distributed Common Object Model) traffic.

Total Alerts - This number gave an idea of how much trouble this host was causing.

For example, the Cisco "TCP SYN Host Sweep" alert fired once for every fifteen host scans.

This meant that the above 130 alerts for the "TCP SYN Host Sweeps" alert actually

represented 1,950 scans; the infected host had scanned 1,950 other hosts looking for

systems to infect.

With the information gathered it was possible to make some assumptions regarding

these alerts. The scanning suggested the host had been infected with a worm and was

actively trying to infect other systems. The worm was probably a part of a bot network that

used IRC for command and control over port 6667. When the worm found a Windows host

that responded to the TCP SYN packet scan, sometimes referred to as "half-open" scanning,

a fast method to scan for hosts (Fyodor), the Windows host tried to infect it via a LSASS RPC

overflow, RPC DCOM overflow or a SMB/RPC NoOp sled.

Date of Last Alert - Date (from the sensor) of the last alert activity helped to determine

how old the activity was and therefore the chance the user was still on the workstation. The

alerts were less than a day old.

Russell Meyer   26

Time of Last Alert – The time recorded the last alert activity on that particular host. This was recorded in order to determine how old the activity was and the chance that the user was still on the workstation. In this case, the data was exported from IEV and the report was generated around 1:30 p.m. The last alerts were around 1:20 p.m.

Category - In the example, two categories are listed: Instant Messenger (IM) and Worm\Virus\Trojan. The "IRC Channel Join" alert can be indicative of bot activity. IRC (Internet Relay Chat) was the forerunner and protocol of modern Instant Messaging. The presence of several IRC Channel Join alerts combined with other alerts could mean the workstation had been infected with a worm and was part of a bot network. In this case, the worm turned out to be a Gaobot variant. The Gaobot worm with over 1000 variants and could have exploited several Microsoft vulnerabilities including MS04-011 (LSASS), MS03-001 (RPC Locator), MS03-026 (DCOM RPC) and MS03-007 (WebDAV) (W32/Gaobot.worm.gen 2004).

Internal or External - In this case, the source was scanning and attacking hosts on the Internet. IP address 10.128.9.57 was scanning hosts in the 24.9.x.x IP range. The 24.9.x.x range was registered to "Comcast Cable Communications" and was probably someone's cable modem. Based on a "whois" lookup, the IP range 24.8.0.0 to 24.9.255.255 appeared to be in Colorado.

Russell Meyer 27

User Logged into Host at Time Report was Run - This information was collected when

the IDS report was run. In many cases, the same person stayed logged on to the same host

all day but in this case, since the VPN IP address space was reused after a user

disconnected, it was impossible to tell if the computer logged was actually infected. Since the

last alert was at 1:20 PM and the report was run at 1:30 PM, it was probable that this was the

user and laptop that generated the alerts. Old IDS alert data run through the Perl program

may not generate accurate reports.

If a user is logged into an infected system, the user name can be captured via the

"nbtstat –A" command. For example:

```
c:\>nbtstat -A 10.28.40.250
Local Area Connection
Node IpAddress: [10.128.40.250] Scope Id: []
    NetBIOS Remote Machine Name Table
    Name                    Type         Status  --
HOU001WS1023    <00>  UNIQUE      Registered
COMPANY         <00>  GROUP       Registered
DSMITH          <03>  UNIQUE      Registered
HOU001WS1023    <20>  UNIQUE      Registered
COMPANY         <1E>  GROUP       Registered

    MAC Address = 00-0B-CD-43-90-AF
```

The above was from a Windows host on a NT domain. DSMITH was the user's login

name.

NT User Comment Field - Once the user name had been determined, the "net user"

Russell Meyer   28

command was used to help identify the individual via the NT "User Comment" field. The User

Comment field was from the user's NT domain account details. The company used this field

to store information about the user account which made it useful in tracking down the user.

While the information in the NT User Comment field was unique to this organization, many

organizations add additional information to the user account which can be leveraged to

augment the IDS reports.

Date and Time of Last Login - This indicated the date and time the user logged in.

This helped show that the user and his laptop were infected with a worm. John logged in at

12:51, the last alerts were around 1:20 and the IDS report was generated around 1:30.

MAC Address of Host - The MAC (Media Access Control) address of the Host that

generated the IDS alert. The MAC address of systems on the network was registered with

both WINS (Windows Internet Name Service) and DHCP. The IDS report determines the

MAC from the output of "nbtstat –A".

Network Card Manufacture - Knowing the network card manufacturer was helpful in

tracking down the workstation but not 100% accurate. In this example, the MAC address

suggested the system was using an "AMBIT MICROSYSTEMS CORP" network card. While

this did not identify the workstation brand, it did imply the workstation was not a HP, Compaq,

IBM or Dell since they usually have their own branded network card.

Each Network Interface Card (NIC) in a computer has a unique address called the

MAC address which is made up of a 12-digit hexadecimal numbers. The IEEE (Institute of

Electrical and Electronics Engineers) assigns the first half of the address to the card

manufacturer while the last half of the address is assigned by the manufacturer. ("MAC

Address" 2008) Since the MAC is unique and usually static, the MAC helped identify the

systems on the network.

Using the list of MACs and manufacturers downloaded from the IEEE web site

(http://standards.ieee.org/regauth/oui/index.shtml), it was possible to cross reference the NIC

MAC address with the manufacturer via a hash. For example, the following systems

generated alerts on 1/21/05.

00-90-27-74-85-66	INTEL CORPORATION
00-02-B3-1D-96-D0	Intel Corporation
00-A0-C9-67-04-84	INTEL CORPORATION - HF1-06
00-D0-59-D9-3F-8B	AMBIT MICROSYSTEMS CORP.
00-02-8A-2B-7B-2A	Ambit Microsystems Corporation
00-06-5B-22-00-E7	Dell Computer Corp.
00-50-04-26-85-4A	3COM CORPORATION
00-0D-87-37-9F-FA	Elitegroup Computer System Co. (ECS)
00-04-AC-D8-69-0D	IBM CORP.
00-09-6B-B0-0F-94	IBM Corporation
00-0B-CD-2B-31-0C	Compaq (HP)
00-50-8B-FB-3E-3C	COMPAQ COMPUTER CORPORATION

00-08-02-32-E0-FE   Compaq Computer Corporation

00-0D-9D-9D-96-5C   Hewlett Packard

Note the "Compaq (HP)" name, a product of the Compaq\HP merger.

Host OS (based on TTL) - The TTL or Time-To-Live value determined how many

routers a packet could pass through before the packet was discarded. The TTL was used to

identify the operating system. This in turn helped track down the system. It should be noted

that most of the workstations on networks today are running Windows 2000, XP or Vista, all of

which have a default TTL of 128.

TTL Value	Operating System
128	Windows NT, 2000, XP, 2003, Vista
64	Many Linux and Unix Systems
32	Windows 95, 98 and Millennium Systems

The above are some of the most common operating systems and TTL values. If the

organization has more then 32 routers between two systems, the Linux and Unix systems

may be interpreted to be Windows 9x systems. The "ping -a -n 1 -w 500" command was used

to determine the TTL of the system in question. The "-a" attempted to resolve the hostname

while "-n 1" sent just one packet. This organization pinged internal systems on a fast network

and used the "-w" switch to timeout the ping in 500 milliseconds. The TTL should not be

100% relied on since the value can be changed but it should work for most computers on the

Russell Meyer   31

network. The value reported also gave an idea of how far away the systems were. For

example:

    *NIX (59)
    9x-ME (27)
    9x-ME (29)
    NT-2000-XP (122)
    NT-2000-XP (126)

Site name - Since the Main Data Center managed the VPN IP range, the "Main

Datacenter (VPN)" was the site name. The internal network used the private class A address

space of 10.0.0.0. A lookup table was used to match the IP address to the physical location

based on the first 3 octets of the IP address. This list was read into a hash for easy lookup.

For example:

    10.106.1.,Northwest - South
    10.106.2.,Northwest - Valley
    10.128.1.,Datacenter
    10.128.9.,Main Datacenter (VPN)
    10.131.11.,Corporate – 11th floor
    10.131.12.,Corporate – 12th floor
    10.131.14.,Corporate – 14th floor
    10.131.15.,Corporate – 15th floor
    10.133.16., Doctors Hospital – NW Lab

Russell Meyer   32

10.133.17., Doctors Hospital – Eastside

Site Contact - Robert Smith was the IDS report contact for the "Main Datacenter (VPN)." A

contact that was responsible for IDS follow-up at the site was mapped to each IP subnet. A

flat text file and a hash were used for simple site contact lookup. For example:

10.151.6.,Doug Smith
10.151.9., Doug Smith
10.86.15.,Jason Rogers
10.152.9.,Becky Will
10.128.9., Robert Smith

Once the IDS alerts had been identified for follow-up, additional information on the

alerts was needed to produce an actionable report. The exact details depended on the needs

of the organization; the ones identified above met this organization's needs. Care had to be

taken with using DHCP to assign IP addresses since the additional details generated by the

Perl program were based on when the report was run, not when the IDS sensor generated

the alert. To avoid some of these issues, the report was run on a regular basis, at least once

a day during business hours.

Russell Meyer   33

5.   Following up on the Alerts

The third challenge of IDS in the enterprise was to follow-up on the alerts; this was the most difficult challenge. While the first two challenges were more technical in nature, the third was dependent on the organization, its resources and its politics.

After an actionable report had been generated, it was necessary to determine who would be responsible for following up on the alerts. In a large organization, the group that monitored the IDS is not necessarily the same group that managed the servers and workstations. Additional issues addressed how to handle alerts generated for systems outside the IT area of responsibility. For example some systems were supported by the vendor but on the organization's network. Procedures were developed to track the alerts to resolution. Reports showed alert trending. Guidelines were developed to determine when to try and clean an infected system vs. just re-imaging the system. As the procedures were developed, it allowed the various groups to assign resources to address the alerts more timely.

The desktop support group or the helpdesk may be the best choice for follow-up on alerts produced by workstations. Before investigating alerts there should be guidelines and training so that issues are appropriately dealt with and escalated if needed. The person or

Russell Meyer   34

group that responds to the alerts will vary depending on the number and the types of alerts. A single workstation that triggers a spyware alert would command one response while a thousand workstations infected with a worm should trigger a different response.

The policies, procedures and guidelines for addressing the systems that generate alerts should be developed before there is a problem. There should be a policy to work with the vendors responsible for systems that IT is not responsible for. Every system should have an owner and maybe even an IT custodian who is responsible for responding to the IDS alerts and cleaning up the machines. Proactive security measures could include patching and anti-virus updates as well as host based protection like software firewalls.

The person or group responsible for following up and investigating the alert depended on the resources available. For servers in the main data center, the IT custodian of the system was involved as well as the data owner. The system administrator did the initial investigation based on the IDS alert data but the system custodian and system owner had to be informed if the issue was not routine.

Alerts that are generated by systems outside of the responsibility of the internal IT group made up the smallest number of systems but generated the greatest share of the alerts. These systems included portable workstations brought in by consultants, vendors and business partners. There could also be unknown systems that were managed by outside

vendors but connected to the network. The portable workstations were addressed via policies

(do not connect unauthorized workstations to the network) and procedures (procedure to

authorize the workstations – i.e. after they have been checked & certified clean). Technical

means, for example network access control (NAC), should be used to enforce the policies.

Once the report was generated and the party responsible for the system was notified

that their system generated a network IDS alert there was mechanism to track the alert for

follow-up and resolution. The responsible party documented the alert to resolution. In this

organization this was accomplished using the existing work ticketing system. The alert and

additional details were entered into the ticket tracking system and assigned to the person or

group responsible for the system. The responsible party worked the ticket and entered the

details regarding the ticket resolution. The IDS monitoring team then reviewed the completed

tickets and in the case of false positives tuned the IDS sensors or modified the filter.

The level of detail and effort expended in tracking the alerts will depend on the

resources available. The IDS report distribution and tracking process can begin as an

informal process but should mature as time goes by and the staff becomes comfortable

responding to the alerts. Closing the loop on alerts is critical for long-term network IDS

trending and analysis.

IDS reports should be generated on a regular basis. While 24/7 monitoring is usually

Russell Meyer   36

ideal, (no one can move faster then a packet) reports will always be generated after the fact.

If the organization is not continually monitoring and reporting on the network IDS, the report

should be generated on a regular basis. Generate and distribute the report too often and the

staff charged with follow-up will spend all their time investigating the same alerts; if the

organization does not generate and distribute the report on regular basis, there could be too

many alerts to follow-up on and too many systems compromised. This organization reviewed

the IEV in the morning and several times during the day but usually generated the report only

in the afternoon so there was still time in the day to address any systems that needed

immediate follow-up.

With multiple building and sites, reports were generated per location using the IP

address of the systems generating the alerts. These were then distributed to the local IT staff

charged with the responsibility for the local systems.

Weekly or monthly trending was done in order to ensure the alerts were being

addressed. Those trending reports were useful in several ways. They demonstrated the

effectiveness of network IDS monitoring to management and helped justify additional sensors

or sensor placement. Trending reports included all alerts captured, not just the alerts that

passed through the filter. These reports were used to improve the effectiveness of the filter

by adding and subtracting alerts.

Russell Meyer   37

Incident response and handling is outside the scope of this paper but there should be some guidelines if not policies and procedures for incident handling for the IDS alerts. Each organization must determine how they plan to deal with the most common alerts and to what extent. For example, should spyware just be removed from workstations or should it be investigated? Should a workstation be re-imaged if a worm is found or only if the worm can not be removed? When should a revenue generating production system be taken off line in order to remove the worm and patch the system? Who should make that call? If a breach may have resulted in exposing customer, employee or health records, when should Human Resources or the Compliance departments become involved? What is the protocol if pornography is discovered on a workstation that is also infected with multiple viruses? When is law enforcement called? What evidence is preserved? How is it preserved? What about the discovery of pirated movies, MP3s or software discovered on a workstation? Who should be notified, how and when should it be done?. These are important questions to consider before an event occurs.

6.   Program Improvements

The Perl program can be improved. The IEV alert data export process was a manual process. It would be useful if the program could be automated to query IEV's MySQL database and then configured to generate reports on a regular basis. It takes several minutes

Russell Meyer   38

or longer to process the IEV alert data. This is due to the use of the 'ping' and 'nbtstat'

commands to gather additional information. If the process could query the domain controllers

for the same information, the program's performance would improve. The program output is

in CSV files. While it takes only a few minutes to sort the data and then paste into a

spreadsheet template, the reports should be produced in Excel format. Report distribution

was a manual process but could have been automatically sent to the ticketing system or even

emailed automatically. The Perl program served the purpose for which it was designed. It

turned tens of thousands of alerts into an actionable report that could be followed-up on by

the appropriate people.

7.   Conclusion

A network IDS can play an important security role in an organization but only if the

number of alerts can be managed. The most important alerts must be identified, reported and

followed up in a timely and organized way. Monitoring the IDS gave insight to the network but

generated a great deal of alerts which required follow-up. Organizations that implement an

IDS should be prepared for the additional resources required to monitor and follow-up. By

creating an actionable IDS report and reacting to a predefined set of alerts, an organization

Russell Meyer   39

can better manage the resources required for the IDS response.

8.   References

Axelsson, S. (1999, May 20). *The Base-Rate Fallacy and its Implications for the Difficulty of*

Intrusion Detection Retrieved March 16, 2008, from http://www.raid-

symposium.org/raid99/PAPERS/Axelsson.pdf

Beaver, K. (2003, September 23). *Does spyware and adware qualify as 'malicious software'*

under the HIPAA rules? Retrieved January 28, 2008, from

http://searchsecurity.techtarget.com/expert/KnowledgebaseAnswer/0,289625,sid14_gci

930265,00.html

Carter, E. (2002, February 15). *Intrusion Detection Systems* Retrieved March 17, 2008, from

http://www.ciscopress.com/articles/article.asp?p=25334

Fyodor (n.d.). *Port Scanning Techniques* Retrieved March 16, 2008, from

http://nmap.org/man/man-port-scanning-techniques.html

Cisco (2003). *How Cisco IT Upgraded Intrusion Detection to Improve Scalability and*

Performance. Retrieved January 28, 2008 from

http://www.cisco.com/web/about/ciscoitatwork/downloads/ciscoitatwork/pdf/cisco_ids42

50_casestudy.pdf

Innella, P. (2001, November 16). *The Evolution of Intrusion Detection Systems*. Retrieved

January 28, 2008, from http://www.securityfocus.com/infocus/1514

Russell Meyer   41

MAC address (n.d) Retrieved March 16, 2008, from http://en.wikipedia.org/wiki/MAC_address

Meyer, R. (2008, March 12). *Actionable IDS Reports* Retrieved March 12, 2008, from

http://sourceforge.net/projects/actionableids/

Newman, D, Snyder, J, Thayer, R. (2002, February 24). *Crying wolf: False alarms hide*

attacks Retrieved March 15, 2008, from

http://www.networkworld.com/techinsider/2002/0624security1.html

W32/Gaobot.worm.gen (2004, May 16). Retrieved March 16, 2008, from

http://vil.nai.com/vil/Content/v_100785.htm

Wilson, T. (2007, June 12). *Pfizer Falls Victim to P2P Hack*. Retrieved January 28, 2008,

from http://www.darkreading.com/document.asp?doc_id=126297