Complex Signature IDS

Correlating System and Application Logs with Traffic Traces and IDS Alerts

Mike Poor
mike@digitalguardian.net

The Dilemma

Network ‘Attack’ Detection Systems

PROS:

– very complex

– “good” at detecting attacks

– real time performance and speed

CONS

– Systems will not indicate whether attack was successful or not

– Overwhelming amounts of alerts

Objectives

iptables log analysis
tcpdump audit trail
Microsoft Event Viewer
Web server log data
Snort alert processing
Syslog signatures
Correlation of data

Iptables Firewall Detect

May 26 11:42:17 bello kernel: FW-DROP-DEFAULT IN=eth0 OUT=MAC=00:07:95:ad:53:2e:00:00:c5:79:60:5c:08:00 SRC=211.186.120.193 DST=192.168.1.7 LEN=48 TOS=0x00 PREC=0x00 TTL=111 ID=37130 DF PROTO=TCP SPT=3230 DPT=1433 WINDOW=16384 RES=0x00 SYN URGP=0

Iptables firewall detects a massive scan for TCP 1433, commonly associated with MS-SQL

May 25th Incidents.org notices massive spike in scans for TCP 1433

What the firewall detect sees is only the dropped packets…

What if a packet (or 100) got through? Would you know?

May 26 11:42:17 bello kernel: FW-DROP-DEFAULT IN=eth0 OUT= MAC=00:07:95:ad:53:2e:00:00:c5:79:60:5c:08:00 SRC=211.186.120.193 DST=192.168.1.7 LEN=48 TOS=0x00 PREC=0x00 TTL=111 ID=37130 DF PROTO=TCP SPT=3230 DPT=1433 WINDOW=16384 RES=0x00 SYN URGP=0

May 26 11:42:17 bello kernel: FW-DROP-DEFAULT IN=eth0 OUT= MAC=00:07:95:ad:53:2e:00:00:c5:79:60:5c:08:00 SRC=211.186.120.193 DST=192.168.1.4 LEN=48 TOS=0x00 PREC=0x00 TTL=111 ID=37131 DF PROTO=TCP SPT=3227 DPT=1433 WINDOW=16384 RES=0x00 SYN URGP=0

May 26 11:42:17 bello kernel: FW-DROP-DEFAULT IN=eth0 OUT= MAC=00:07:95:ad:53:2e:00:00:c5:79:60:5c:08:00 SRC=211.186.120.193 DST=192.168.1.10 LEN=48 TOS=0x00 PREC=0x00 TTL=112 ID=37134 DF PROTO=TCP SPT=3233 DPT=1433 WINDOW=16384 RES=0x00 SYN URGP=0

Check the initial Handlers Diary at Incidents.org (http://www.incidents.org/diary/diary.php?id=156) for more information on traffic spikes relating to the SQL-Snake worm. This particular worm exploited machines running MS-SQL Server which have ‘SA’ accounts with no password.

Response to Iptables Detect

Common Reactions:

- Add offending hosts to drop lists
- Check Cert and Bugtraq for vulnerabilities associated with TCP 1433
- Scan own Network to see if you are running TCP 1433

Am I running anything on TCP 1433?

- Examine tcpdump audit logs for connections to/from TCP 1433 and the outside world

The common reactions to scans run the gamut from "its just a scan… why should I worry about a scan" to "lets automatically block any ip that scans us". Both are naïve, and in my opinion wrong. Blocking offending hosts based on analysis of inappropriate network activity is perfectly acceptable, but auto-blocking sets one up for a serious DOS attack. Not paying attention to scans will cost you dearly in lost information gathering on your enemy at the gate.

We can learn many things from scanning activity. If the scan is for a specific service, we can make the assumption that the attacker has an exploit for that service, or that he/she is looking for a backdoor into your systems. If they are scanning for a specific service that is only found on one operating system, the attackers probably have a method of compromising that O.S.

Iptables detect response (cont.)

 # tcpdump –r logfile.dmp 'tcp and src port 1433 and (tcp[13] = 18)'

192.168.1.20.1433 > 211.186.120.193 .3256: S 1769109842:1769109842(0) ack 959773354 win 5840 <mss 1460> (DF)

192.168.1.10.1433 > 211.186.120.193.3273: S 3205250073:3205250073(0) ack 3833518038 win 16616 <mss 1260> (DF)

192.168.1.110.1433 > 211.186.120.193.3358: S 2619591529:2619591529(0) ack 3480979530 win 16616 <mss 1460> (DF)

What we have done here is read in a binary (pcap format) format log file named logfile.dmp looking for all machines that sent a SYN/ACK packet from a source port of 1433.

Now that we have a list of potentially compromised machines, the System Administrator can now don the hat of the Incident Handler and isolate those machines and assess the level of compromise. The next steps would certainly be to run forensic tools on the machines, gather recovery information from Cert, Security Focus, Incidents.org asd other security information sources.

tcpdump audit trail

tcpdump –r logfile.dmp 'tcp and src port 1433 and (tcp[13] = 18)'

This simple command would provide you with all SYN/ACK packets coming from a source port of 1433

Now a system administrator can concentrate on Incident Response if any machines have responded

tcpdump is freely available, and highly portable.

tcpdump was originally developed by Lawrence Berkeley Labs, funded by DARPA.

tcpdump is maintained by http://www.tcpdump.org

A simple cursory knowledge of BPF filter creation can give an IDS analyst awesome power to extract and analyze events of interest on the network. In the case of the above filter:

'tcp and src port 1433 and (tcp[13] = 17)'

we are looking for tcp packets, coming from a source port of 1433, that have a value of 3 in the 13th byte offset from zero of the tcp header. This field corresponds with the tcp flags field, and the two least significant bits are the SYN and the ACK, as shown below.

128	64	32	16	8	4	2	1
RES	RES	URG	ACK	PSH	RST	SYN	FIN

tcpdump audit trail

tcpdump's default capture length is 68

Frame header = 14 bytes

IP header = 20 bytes (min); 60 bytes (max)

TCP header =20 bytes (min); 60 bytes (max)

Set capturelength to 200 allowing for capture of tcp packet with options +- 80 bytes of embedded data

tcpdump audit trail will give you the fidelity logging that is required to do forensic traffic analysis

Skip ports of extremely high traffic if necessary

The main concerns when setting up a tcpdump audit trail is: how much data can/should we capture? We need to capture enough data to establish whether or not connections took place, and give us a general idea of what took place during that connection.

It is not necessary on the other hand to capture every byte of data in and out of your networks. Perhaps you have a web server farm that is well monitored by other technologies (NIDS, syslog logging facility, and webserver logs). You may wish to exclude TCP port 80 from your captures (this should eliminate your heaviest burden).

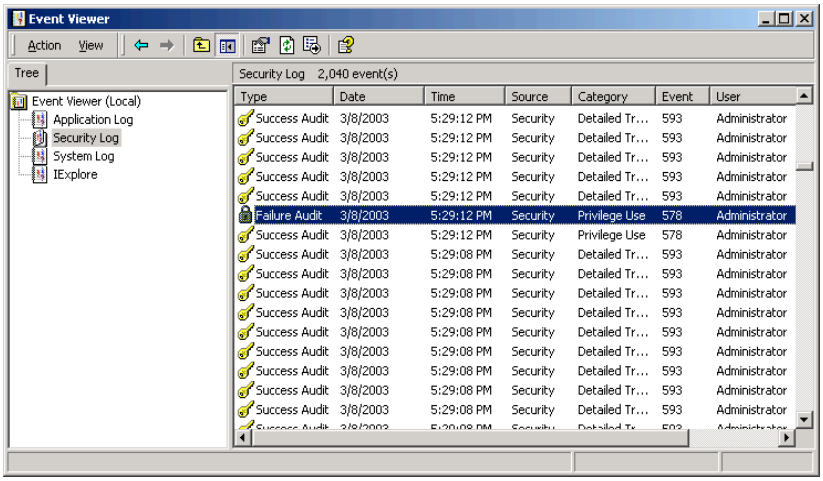
tcpdump audit logs

- Keep logs in operational store for one week (variable)
- Save logs to write once media (DVD, CDROM)
- Binary files can be read back with tcpdump, snort, ethereal, virtually any libpcap aware application.
- Shadow can help organize and maintain your tcpdump audit trail
- Shadow is available for free, from http://www.nswc.navy.mil/ISSEC/CID/

These numbers are subjective, your mileage may vary. The length of time that you keep logs in operational store (online) will vary according to the amount of data that you collect and the storage size allotted to your online log storage.

Binary tcpdump logs are compact and very versatile. Once saved they can be read by any libpcap aware application, which adds to its universal appeal.

Windows 2000 Event Viewer



This space intentionally left blank

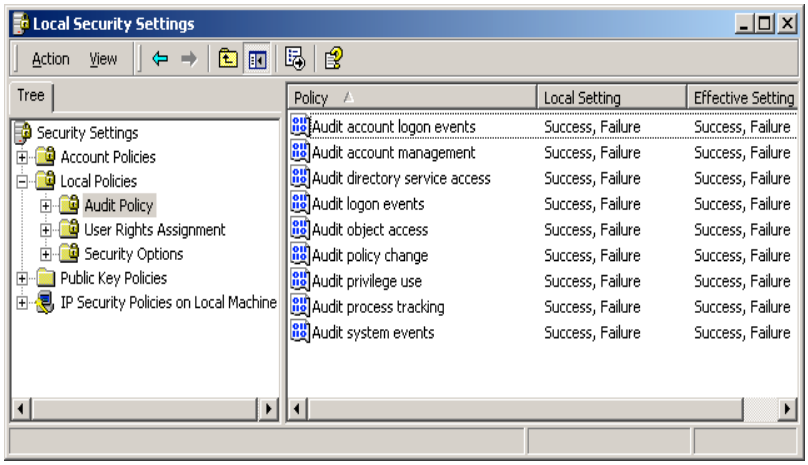
Microsoft Events

Windows 2000 Professional Event Viewer
Security Accounting turned off by default
Default Event Viewer file size of 512 Bytes
Very powerful process accounting
Very detailed event logging
Largely overlooked in security auditing

Different tools can be used to export Microsoft Events to Syslog. Two of the most powerful and popular are NTEventlogger and NTsyslog. NTEventlogger is a commercial product, and NTsyslog is released under the GPL.

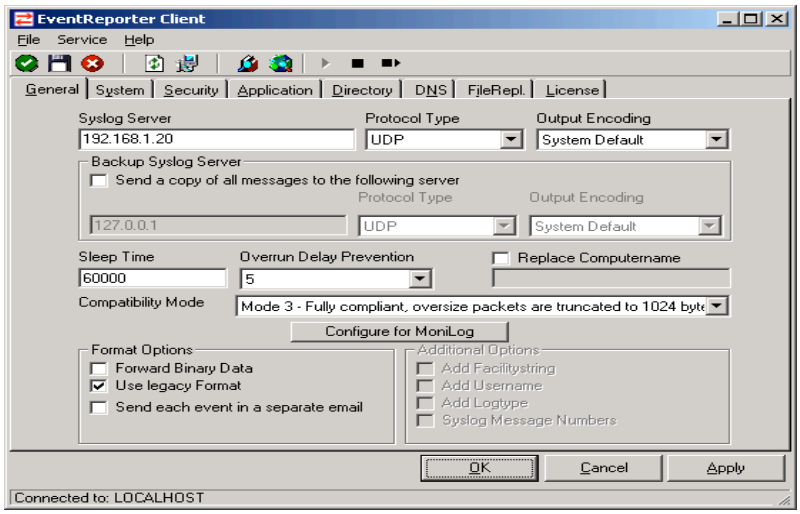
For a complete list of available projects, check http://www.loganalysis.org/sections/syslog/windows-to-syslog/index.html

First step: Enable Auditing



The first step that must be done is to enable Security Auditing. This is turned off by default. Next you want to increase the maximum size of your Event logs, as the default is 512 bytes. This value must be set in 64 byte increments, but Windows will automagically resize them for you. The default setting is to overwrite events as needed. If you have your events exported to syslog, this is probably ok. Otherwise, you have two other choices, clear logs manually (which leaves you in the uncomfortable position of missing new events until you do), or to clear logs after X amount of days (same problem as above).

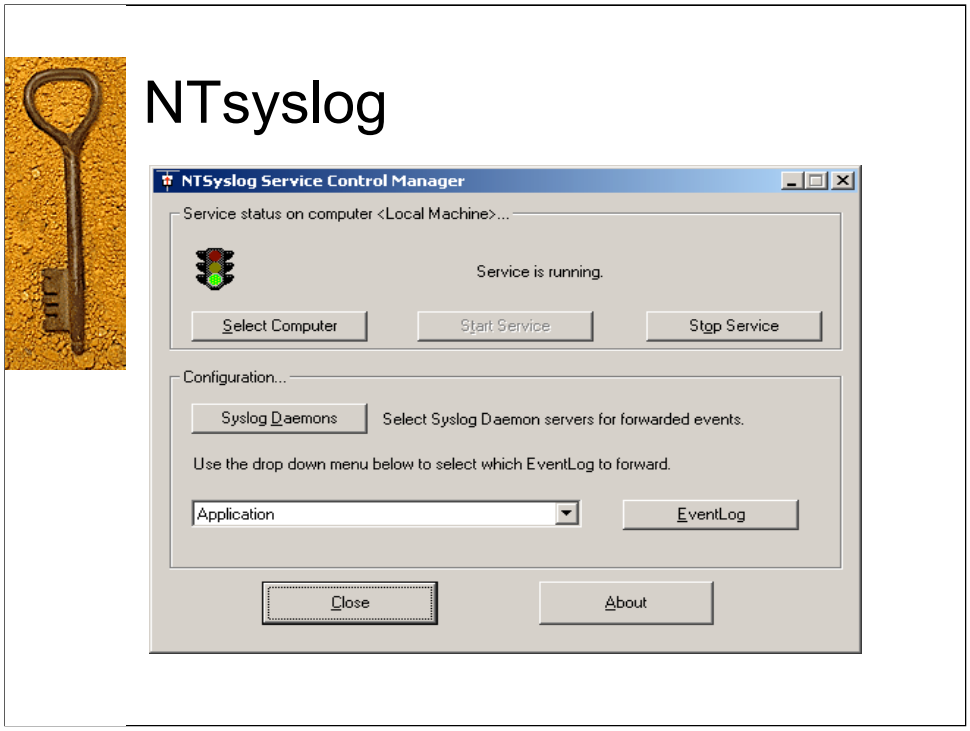
Event Reporter Client



Two different ways to export your NT event logs to centralized syslog host. 1: Event Reporter (shown above). 2. NTsyslog (next slide).

Event reporter is a great tool, and very useful in the enterprise. It is full featured, supports numerous syslog/syslog-ng options, including different compatibility modes as well as sending messages to email.

Event reporter is commercial software, which costs $50 per client, with bulk discounts.



NTsyslog is a free open source project, released under GNU General Public License (free as in free beer).

NTsyslog is a full featured service that exports NT/win2k events to a centralized syslog server. NTsyslog installs as a service on the Windows host, so no need for human interaction after setup is needed. The export format is fully compatible with syslog protocol.

Windows Events

Feb 23 21:45:22 192.168.1.10 borg EvntSLog: [AUS] BORG/Security (624) - "User Account Created: New Account Name: evil0ne New Domain: BORG New Account ID: %{S-1-5-21-1708537768-746137067-854245398-1004} Caller User Name: Administrator Caller Domain: BORG Caller Logon ID: (0x0,0x9699) Privileges - “

New user: evil0ne created

In notes, see that the group this user is added to is the Administrators group

Note event id # 624

Feb 23 21:45:22 192.168.1.10 borg EvntSLog: [AUS] BORG/Security (624) - "User Account Created: New Account Name: evil0ne New Domain: BORG New Account ID: %{S-1-5-21-1708537768-746137067-854245398-1004} Caller User Name: Administrator Caller Domain: BORG Caller Logon ID: (0x0,0x9699) Privileges - "

Feb 23 21:45:22 192.168.1.10 borg EvntSLog: [AUS] BORG/Security (642) - "User Account Changed: Account Enabled. Target Account Name: evil0ne Target Domain: BORG Target Account ID: %{S-1-5-21-1708537768-746137067-854245398-1004} Caller User Name: Administrator Caller Domain: BORG Caller Logon ID: (0x0,0x9699) Privileges: - "

Feb 23 21:45:23 192.168.1.10 borg EvntSLog: [AUS] BORG/Security (636) - "Security Enabled Local Group Member Added: Member Name: - Member ID:%{S-1-5-21-1708537768-746137067-854245398-1004} Target Account Name: Administrators Target Domain: Builtin Target Account ID: %{S-1-5-32-544} Caller User Name: Administrator Caller Domain: BORG Caller Logon ID: (0x0,0x9699) Privileges: - "

Highly recommended: Security Operations Guide for Windows 2000 Server:

http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/prodtech/windows/windows2000/staysecure/secops06.asp

Event's of Interest

Logon Events
- Successful Network Logins (540)
- Failed Logins of any kind (529-534, 537)

Account Management Events
- Creation of new accounts (624)
- User account type changed (625)
- Change of password (627 failed/ 628 successful)
- Lockout of Account (644)

System Events
- Windows starting up (512)
- security logs cleared (517)

Highly recommended: Security Operations Guide for Windows 2000 Server:

http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/prodtech/windows/windows2000/staysecure/default.asp

For a complete list of Intrusion Detection and Auditing related Event ID Numbers:

http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/prodtech/windows/windows2000/staysecure/secops06.asp

Code Red, Nimda, & Worm d'jour

Grepping through snort alerts you are faced with this:

```
$ grep -v 'spp' alert | grep '\[\*\*\]' | sort | uniq -c | sort -rn
2908 [**] [1:1002:3] WEB-IIS cmd.exe access [**]
202 [**] [1:1256:4] WEB-IIS CodeRed v2 root.exe access [**]
```

How do you differentiate between successful and failed attempts to exploit these vulnerabilities?

```
[**] [1:1002:3] WEB-IIS cmd.exe access [**]
[Classification: Web Application Attack] [Priority: 1]
05/25-15:53:39.997260 64.86.5.8:4147 -> 192.168.1.5:80
TCP TTL:116 TOS:0x0 ID:2411 IpLen:20 DgmLen:120 DF
***AP*** Seq: 0x7F2D92  Ack: 0x4D605F7D  Win: 0x2238  TcpLen:
   20
```

This is a simple grep through a random days worth of logs. We see that we are constantly being scanned, probed and prodded for vulnerable IIS web servers. The fact of the matter is, many of us do not know which hosts are running what on our network, much less what patch level they are at.

Snort Log of a cmd.exe event

```
05/25-14:53:39.997260 64.86.5.8:4147 -> 192.168.1.5:80
TCP TTL:116 TOS:0x0 ID:2411 IpLen:20 DgmLen:120 DF
***AP*** Seq: 0x7F27CD92  Ack: 0x4D605F7D  Win: 0x2238  TcpLen:
    20
47 45 54 20 2F 64 2F 77 69 6E 6E 74 2F 73 79 73  GET /d/winnt/sys
74 65 6D 33 32 2F 63 6D 64 2E 65 78 65 3F 2F 63  tem32/cmd.exe?/c
2B 64 69 72 20 48 54 54 50 2F 31 2E 30 0D 0A 48  +dir HTTP/1.0..H
6F 73 74 3A 20 77 77 77 0D 0A 43 6F 6E 6E 6E 65  ost: www..Connne
63 74 69 6F 6E 3A 20 63 6C 6F 73 65 0D 0A 0D 0A  ction: close....
```

Again, this is great information, but it does not give us much to go on

To get to the bottom of this mystery, we must go to the webserver logs

This shows a packet capture from Snort IDS, showing an attempt to pass parameters to the windows command shell.

Web Server Sanity Check

First on a vulnerable Windows IIS Webserver:

$ grep 'cmd.exe' ex030216.log | wc -l
3025
$ grep 'cmd.exe' ex030216.log | grep ' 200' | wc -l
1362

Next on a non-vulnerable Apache Webserver

$ grep 'cmd.exe' access_log | wc -l
623461
$ grep 'cmd.exe' access_log | grep ‘ 200' | wc -l
0

Everyone involved in network and computer security should have the opportunity to do a sanity check on a web server and run two simple greps:

grep ‘cmd.exe’ <filename>

grep ‘/etc/passwd’ <filename>

From an Apache web server running on Linux (not vulnerable):

X.X.X.X - - [30/Jan/2003:14:55:17 -0500] "GET /scripts/..%c0%2f../winnt/system32/cmd.exe?/c+dir

HTTP/1.0" 404 –

X.X.X.X - - [30/Jan/2003:14:55:18 -0500] "GET /scripts/..%c0%af../winnt/system32/cmd.exe?/c+dir

HTTP/1.0" 404 -

X.X.X.X - - [30/Jan/2003:14:55:18 -0500] "GET /scripts/..%c1%9c../winnt/system32/cmd.exe?/c+dir

HTTP/1.0" 404 –

From a vuln windows webserver, first we see an attempt that didn’t work, and then one that did:

14:45:45 X.X.X.X GET /scripts../../../../winnt/system32/cmd.exe 404

14:46:21 X.X.X.X GET /scripts../../../../winnt/system32/cmd.exe 200

Web server Intrusion Data

192.168.1.101 GET /scripts/../../../../winnt/system32/cmd.exe 200
ip address | http command | URI | HTTP server status code

HTTP/1.1 Server Status Codes commonly seen during attacks:
- **200 OK "The request has succeeded".**
- **403 Forbidden**
- **404 File Not Found**
- **500 Internal Server Error**

2xx Codes are indications of success
4xx Codes are indications of client error
5xx Codes are indications of server error

14:26:18 192.168.1.100 GET /scripts/.Á□../winnt/system32/cmd.exe 500

>> 500 Internal Server Error "The Server encountered an unexpected condition which prevented it from fullfilling the request.

14:28:57 192.168.1.100 GET /.Á/scripts/.Á□../winnt/system32/cmd.exe 404

>> 404 not found "The server has not found anything matching the Request – URI.

15:06:00 192.168.1.101 GET /scripts/../../../../winnt/system32/cmd.exe 200

>> 200 OK. "The request has succeeded. The information returned with the response is dependent on the method used in the request."

15:06:41 192.168.1.101 GET /scripts/../../../../winnt/system32/cmd.exe 200

See http://www.w3.org/Protocols/rfc2616-sec10.html or
view the rfc directly at ftp://ftp.rfc-editor.org/in-notes/rfc2616.txt

Nmap Baseline Analysis

- Nmap is a great tool for attackers and defenders alike
- Best to see your network as the attackers do
- Free, open source network mapper available from http://www.insecure.org
- Can be used to create a baseline for auditing your network

Nmap is a great tool for auditing your network. Before going down this road, I must caution you however to the dangers of network scanning. Plenty of people have broken networks, caused network outages and disruptions, and lost their jobs or faced prosecution due to unauthorized network mapping. GET PERMISSION IN WRITING.

That said, I find that using nmap to create a known baseline for your listening services on known hosts, then monitoring your network against this baseline (using tools like ndiff) can be of great service in detecting intrusions.

Ndiff

- Free Perl module for comparing nmap scans
- Very useful for catching backdoor listeners
- Can output plain text or html reports
- Comes with a variety of tools to manage and run nmap scans
- Built for Linux (but probably easily portable to other Unices)

Ndiff is written and maintained by James Levine (jdl@vinecorp.com).

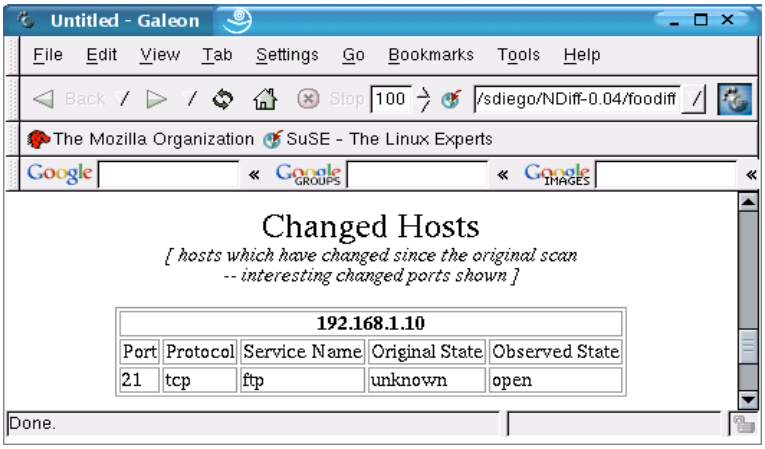
From www.vinecorp.com/ndiff/:

Ndiff compares two nmap scans and outputs the differences.

Ndiff can easily be run as a cron job (see man pages for ndiff and nrun that comes with the package) to automatically run nmap and ndiff, and log the differences from the baseline.

Ndiff (cont.)

Ndiff output in HTML indicating new port listening on TCP 21



Wu-ftpd woes

Snort detects a format string attack coming across the wire directed at port 21

[**] [1:1971:1] FTP SITE EXEC format string attempt [**]
[Classification: Potentially Bad Traffic] [Priority: 2]
02/18-23:54:56.583689 192.168.1.40:5247 -> 192.168.1.120:21
TCP TTL:64 TOS:0x0 ID:18709 IpLen:20 DgmLen:563 DF
AP Seq: 0xCBF0F49A Ack: 0x1EA7F50A Win: 0x81D0
TcpLen: 32
TCP Options (3) => NOP NOP TS: 14407154 900325

* See notes for the other alerts

ftp.rules:alert tcp $EXTERNAL_NET any -> $HOME_NET 21 (msg:"FTP SITE EXEC format string attempt"; flow:to_server,established; content:"SITE"; nocase; content:"EXEC "; nocase; distance:0; content:"%"; distance:1; content:"%"; distance:1; classtype:bad-unknown; sid:1971; rev:1;)

[**] [1:1971:1] FTP SITE EXEC format string attempt [**]
[Classification: Potentially Bad Traffic] [Priority: 2]
02/18-23:54:56.583689 192.168.1.40:5247 -> 192.168.1.120:21
TCP TTL:64 TOS:0x0 ID:18709 IpLen:20 DgmLen:563 DF
AP Seq: 0xCBF0F49A Ack: 0x1EA7F50A Win: 0x81D0 TcpLen: 32
TCP Options (3) => NOP NOP TS: 14407154 900325

[**] [1:1748:4] FTP command overflow attempt [**]
[Classification: Generic Protocol Command Decode] [Priority: 3]
02/18-23:54:57.602011 192.168.1.40:5247 -> 192.168.1.120:21
TCP TTL:64 TOS:0x0 ID:18711 IpLen:20 DgmLen:201 DF
AP Seq: 0xCBF0F699 Ack: 0x1EA7F8F0 Win: 0x8C58 TcpLen: 32
TCP Options (3) => NOP NOP TS: 14407256 900326
[Xref => bugtraq 4638]

[**] [1:498:3] ATTACK RESPONSES id check returned root [**]

Wuftpd woes (cont.)

Now that we have our snort alerts, we must validate the traffic

After examining the snort rule, its time to view the packet that triggered the alert

tcpdump –nnXr <logfile> 'port 21 and port 5247'

* See notes for packet dump

23:54:56.583689 192.168.1.40.5247 > 192.168.1.120.21: P 36557:37068(511) ack 69467 win 33232 <nop,no

p,timestamp 14407154 900325> (DF)

```
0x0000   4500 0233 4915 4000 4006 6bbf c0a8 0128        E..3I.@.@.k....(
0x0010   c0a8 0178 147f 0015 cbf0 f49a 1ea7 f50a        ...x............
0x0020   8018 81d0 9989 0000 0101 080a 00db d5f2        ................
0x0030   000d bce5 5349 5445 2045 5845 4320 3720        ....SITE.EXEC.7.
0x0040   fccb ffff bf50 7350 73fd cbff ffbf 5073        .....PsPs.....Ps
0x0050   5073 fecb ffff bf50 7350 73ff ffcb ffff        Ps.....PsPs.....
0x0060   bf25 2e66 252e 6625 2e66 252e 6625 2e66        .%.f%.f%.f%.f%.f
0x0070   252e 6625 2e66 252e 6625 2e66 252e 6625        %.f%.f%.f%.f%.f%
0x0080   2e66 252e 6625 2e66 252e 6625 2e66 252e        .f%.f%.f%.f%.f%.
0x0090   6625 2e66 252e 6625 2e66 252e 6625 2e66        f%.f%.f%.f%.f%.f
0x00a0   252e 6625 2e66 252e 6625 2e66 252e 6625        %.f%.f%.f%.f%.f%
0x00b0   2e66 252e 6625 2e66 252e 6625 2e66 252e        .f%.f%.f%.f%.f%.
0x00c0   6625 2e66 252e 6625 2e66 252e 6625 2e66        f%.f%.f%.f%.f%.f
0x00d0   252e 6625 2e66 252e 6625 2e66 252e 6625        %.f%.f%.f%.f%.f%
```

Wu-ftpd woes (cont.)

- Now we have validated that there was an attack
- Was it successful?
- What did the attacker do after that packet flew by our NIDS?
- Pull syslog data to find out

Now that we have confirmed that the attack matches a known pattern for an attack against a wu-ftdp server, versions 2.6.1 and below, we must now determine whether or not the attack was successful or not.

We know go to our central syslog server and pull the logs relating to that ftp server.

Wu-ftpd woes Syslog Data

Feb 18 00:17:27 192.168.1.120 xinetd[6204]: START: ftp pid=6208 from=192.168.1.40

Feb 18 00:17:30 192.168.1.120 ftpd[6208]: ANONYMOUS FTP LOGIN FROM 192.168.1.40 [192.168.1.40], mozilla@

Feb 18 00:19:53 192.168.1.120 useradd[6308]: new group: name=evil, gid=501

Feb 18 00:19:53 192.168.1.120 useradd[6308]: new user: name=evil, uid=0, gid=501, home=/home/evil, shell=/bin/bash

Feb 18 00:20:32 192.168.1.120 useradd[6309]: new group: name=evilnorm, gid=502

Feb 18 00:20:32 192.168.1.120 useradd[6309]: new user: name=evilnorm, uid=501, gid=502, home=/home/evilnorm, shell=/bin/bash

Feb 18 00:21:31 192.168.1.120 sshd[6313]: Could not reverse map address 192.168.1.40.

Feb 18 00:21:43 192.168.1.120 sshd[6313]: Accepted password for ROOT from 192.168.1.40 port 5242 ssh2

Feb 18 00:21:43 192.168.1.120 sshd(pam_unix)[6313]: session opened for user evil by (uid=0)

As you can see, we start with an anonymous ftp login from 192.168.1.40, no application crash, then he adds two users, one normal and one root. An hour later, we see the user return and login as root via ssh. Sign of Compromise? Now we know for sure.

Feb 18 00:17:27 192.168.1.120 xinetd[6204]: START: ftp pid=6208 from=192.168.1.40

Feb 18 00:17:30 192.168.1.120 ftpd[6208]: ANONYMOUS FTP LOGIN FROM 192.168.1.40 [192.168.1.40], mozilla@

Feb 18 00:19:53 192.168.1.120 useradd[6308]: new group: name=evil, gid=501

Feb 18 00:19:53 192.168.1.120 useradd[6308]: new user: name=evil, uid=0, gid=501, home=/home/evil, shell=/bin/bash

Feb 18 00:20:32 192.168.1.120 useradd[6309]: new group: name=evilnorm, gid=502

Feb 18 00:20:32 192.168.1.120 useradd[6309]: new user: name=evilnorm, uid=501, gid=502, home=/home/evilnorm, shell=/bin/bash

Feb 18 00:21:18 192.168.1.120 sshd[6312]: Connection closed by 192.168.1.40

Feb 18 00:21:31 192.168.1.120 sshd[6313]: Could not reverse map address 192.168.1.40.

Feb 18 00:21:43 192.168.1.120 sshd[6313]: Accepted password for ROOT from 192.168.1.40 port 5242 ssh2

Feb 18 00:21:43 192.168.1.120 sshd(pam_unix)[6313]: session opened for user evil by (uid=0)

Failed wu-ftpd Format Strings Attack

```
0x0000   4500 004c fcf2 4000 4006 b9dc c0a8 0114      E..L..@.@.......
0x0010   c0a8 0178 0402 0015 40e9 3cff e814 a6eb      ...x....@.<.....
0x0020   8018 16d0 8446 0000 0101 080a 008f c00d      .....F..........
0x0030   0006 312d 5349 5445 2045 5845 4320 2530      ..1-SITE.EXEC.%0
0x0040   3230 647c 252e 6625 2e66 7c0a                20d|%.f%.f|.
```

```
23:34:11.173795 192.168.1.120.ftp > foo.foo.1026: P 168:203(35) ack 48 win 5792
<nop,nop,timestamp 405806 9420813> (DF)
0x0000   4500 0057 9772 4000 4006 1f52 c0a8 0178      E..W.r@.@..R...x
0x0010   c0a8 0114 0015 0402 e814 a6eb 40e9 3d17      ............@.=.
0x0020   8018 16a0 783d 0000 0101 080a 0006 312e      ....x=........1.
0x0030   008f c00d 3530 3220 4558 4543 2063 6f6d      ....502.EXEC.com
0x0040   6d61 6e64 206e 6f74 2069 6d70 6c65 6d65      mand.not.impleme
0x0050   6e74 6564 2e0d 0a                            nted...
```

Once we apply patches to the wu-ftpd server, the exploit is tried again, and it does not work.

```
23:34:15.365843 192.168.1.120.ftp > foo.foo.1026: P 203:240(37) ack 49 win 5792
<nop,nop,timestamp 406226 9421232> (DF)
0x0000   4500 0059 9773 4000 4006 1f4f c0a8 0178      E..Y.s@.@..O...x
0x0010   c0a8 0114 0015 0402 e814 a70e 40e9 3d18      ............@.=.
0x0020   8018 16a0 e492 0000 0101 080a 0006 32d2      ..............2.
0x0030   008f c1b0 3232 3120 596f 7520 636f 756c      ....221.You.coul
0x0040   6420 6174 206c 6561 7374 2073 6179 2067      d.at.least.say.g
0x0050   6f6f 6462 7965 2e0d 0a                      oodbye...
```

Basic Correlation Parameters

- Event
- Attacker or Target
- Time
- Host
- Network
- Service

We've seen in this presentation how we can correlate basic information from existing system and application logs to further investigate and analyze the events that come across our ids.

Future Complex IDS Signature Research

We have reached a point where the system resources are available to do complex signature IDS

Need for a Complex Signature Database

Need for log parsing tools that will alert based on log signatures.

http://www.loganalysis.org

Thanks to the work of Tina Bird, Marcus Ranum and many others, we are slowly building up a database of system and application signatures and the tools to find them.

I highly recommend Tina and Marcus' website http://www.loganalysis.org as well as Tina's mailing list.