FEATURE

MALWARE IN A PIG PEN – PART 2

Martin Overton
Independent Researcher, UK

In the first part of this feature (see *VB*, October 2004, p.10) I covered the use of SNORT to detect malware using simple binary and MIME strings. This part will cover more complex malware and SNORT signatures/rules to detect them. It will also cover some other parts of the SNORT rule structure (or nomenclature), as well as some of the other directives including some non-malware related rules/signatures that show how the directive/keyword is used. Finally, the article will look at some of the things that can go wrong when signatures are chosen poorly.

FLAVOURSOME PACKETS

As well as the use of signatures/rules that act on TCP packets, SNORT can act on UDP, ICMP and IP packets. Other packet types (such as IGMP) may be supported in future versions of SNORT.

THIS LITTLE PIGGY WENT TO …

The rule/signature for W32/Netsky.p described in part one of this article showed how to detect infected email coming from an external network to your internal network. But what if you want to reverse this test, or even test both directions at the same time?

You simply change the original part of the signature/rule from:

```
$EXTERNAL_NET any -> $HOME_NET any
```

which is used to detect inbound packets (from an IP not on our internal network), to:

```
$HOME_NET any -> $EXTERNAL_NET any
```

This is used to detect outbound packets (from an IP on our internal network). To reverse the direction of the test you cannot just use '<-', as this is not supported by SNORT.

Alternatively, the following is what you would use if you wanted to test data going in either direction (both inbound and outbound) with a single signature/rule:

```
$EXTERNAL_NET any <> $HOME_NET any
```

GO WITH THE FLOW

A useful keyword is the 'flow' directive. This can be used to limit rules to client or server traffic. For example:

```
alert tcp $EXTERNAL_NET 110 -> $HOME_NET any
(msg:"VIRUS Klez Incoming";
flow:to_server,established; dsize:>120;
content:"MIME"; content:"VGhpcyBwcm9";
classtype:misc-activity; sid:1800; rev:2;)
```

This rule/signature will trigger only once a client has connected to a server (in this case a POP3 server), and will act on the data received from the server.

PIGGIN' WEB CONTENT

Let us imagine that you want SNORT to alert on web traffic that meets a specific signature. The following rule will trigger when a URL contains '/readme.eml' in any letter case (upper, lower or mixed – this is specified by the 'nocase' keyword):

```
alert tcp $HOME_NET any -> $EXTERNAL_NET $HTTP_PORTS
(msg:"WEB-MISC readme.eml download attempt"; flags:A+;
uricontent:"/readme.eml"; nocase; classtype:attempted-
user; sid:1284; reference:url,www.cert.org/advisories/
CA-2001-26.html; rev:8;)
```

This is an ideal solution for handling malware that downloads components and updates from the web, such as W32/Bagle.az@MM (see http://vil.nai.com/vil/content/v_128582.htm) or Downloader-PU (see http://vil.nai.com/vil/content/v_128464.htm).

TELL SID!

The 'sid' keyword is used as a unique identification of a specific rule/signature. However, before starting to number your own signatures you must bear in mind the following:

- Numbers <100 are reserved for future use.
- Numbers 100–1,000,000 are for use *only* for rules included with SNORT (i.e. 'official' rules).
- Numbers >1,000,000 can be used for local rules on a free-for-all basis.

MULTIPLE CONTENT

The first part of this article showed a SNORT signature/rule that contained one 'content' section (signature) to be matched against incoming data. However, SNORT signatures/rules are not limited to single 'content' sections and you can even mix content types, such as binary and text strings. For example:

```
alert udp $EXTERNAL_NET any -> $HOME_NET 1434
(msg:"W32.SQLEXP.Worm propagation (1434)"; con-
tent:"|68 2E 64 6C 6C 68 65 6C 33 32 68 6B 65 72
6E|"; content:"|04|"; offset:0; depth:1;)
```

You can even mix content types in the same content section, for example:

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 139
(msg:"NETBIOS SMB ADMIN$access";
flow:to_server,established; content:"\\ADMIN$|00 41
3a 00|"; reference:arachnids,340;
classtype:attempted-admin; sid:532; rev:4;)
```

OBFUSCATED AND ENCRYPTED SAMPLES

Obfuscated samples

These are samples that are often packed (as many as ten different packers may be used) or that display some mild form of polymorphism, such as adding random text or other garbage to the file in order to fool MD5 and other hash functions.

Encrypted samples

In the context of this article, encrypted samples refer to the password-protected zips that have been seen in many of the Bagle variants – both those with a plain text password in the body of the email and those that use the graphic password trick to try to slow down or stop the scanner from scanning the file held in the password-protected zip.

WHEN SNORT TELLS PORKIES

This section covers some of the possible problems you may encounter when using SNORT. These are not issues with SNORT, but issues you may encounter with the signatures/rules themselves.

FALSE POSITIVES

A false positive occurs when a rule is triggered on a file that is not malicious, but is flagged as if it were. For example, a beast that grunts like a pig and eats like a pig might be flagged as a pig, but actually be a frog (*Rana grylio*, aka the pig frog).

As with anti-virus products, false positives do occur, especially when signatures are selected in haste and are not sufficiently tested. To date I have found very few false positive issues with the signatures/rules I have created. I attribute this to the level of testing I carry out before making the signatures available.

If you use the 'Flexible Response' features (flexresp) in SNORT you could end up with a self-inflicted DoS, so do be careful when using this feature.

FALSE NEGATIVES

False negatives occur when a rule is not triggered on a file that is malicious. For example a beast that grunts like a pig, eats like a pig, and *is* a pig, might be misclassified due to

the fact that it is kept as a house pet (*Sus scrofa domestica,* aka the Vietnamese potbellied pig).

This is a more serious problem, as it means that the signature is flawed and misses 'real' infected files/content that should have been identified. In some cases this is difficult to resolve, especially with complex obfuscated or encrypted malware. Resolving this issue usually requires multiple signatures/rules to be created or a different approach, such as using header information rather than MIME body data.

MAIL HEADERS

Let us now look at a different way of detecting worms, not by the attachment, but by the manufactured headers.

PCRE (PERL-COMPATIBLE REGULAR EXPRESSIONS)

As mentioned, there are sometimes other ways to detect obfuscated or encrypted malware emails reliably, by looking at the manufactured mail headers they use (or do not use).

For example, both the MyDoom and Bagle families use manufactured headers which can be a reliable method of detecting them, without the need to create signatures to detect the attachment.

The following will detect MyDoom-constructed emails, even if they are corrupted, non-viable or truncated:

```
alert tcp $EXTERNAL_NET any -> any any (msg:"MyDoom
Mail Header Match/PCRE"; pcre:"/X-MIMEOLE\: Produced
By Microsoft MimeOLE V6\.00\.2600\.0000/"; pcre:"/
boundary=["][-]{4}\=\_NextPart\_000\_\d{4}\_.{8}\
..{8}/"; pcre:"/filename=["]\S{1,}[.](bat|scr|com|
cmd|exe|pif|zip)/";classtype:misc-activity; rev:1;)
```

The following will reliably detect Bagle-constructed emails under the same conditions as above:

```
alert tcp $EXTERNAL_NET any -> any any (msg:"Bagle
Mail Header Match/PCRE"; pcre:"/Message-
ID\:\W{1,}[<][a-z]{19}[@]/"; pcre:"/boundary=["][-
]{8}[a-z]{20}/"; classtype:misc-activity; rev:1;)
```

You can also use this technique to detect/block unwanted attachments in email:

```
alert tcp $EXTERNAL_NET any -> any any (msg:"Bad
Extensions Match/PCRE"; pcre:"/
attachment\;\W{1,}filename=["]\S{1,}[.](scr|com|exe|cpl|pif|
hta|vbs)/"; classtype:misc-activity; rev:1;)
```

The signature above does not include all recommended 'bad extensions' to block, just a small subset – so feel free to add any you want to include.

The signature/rule below will usually trigger only on password-protected zip files created by email worms:

```
alert tcp $EXTERNAL_NET any -> any any
(msg:"Encrypted PKZip - SUSPECT/PCRE";
flow:to_server,established; pcre:"/
UEsDBAoAA\S{10,}[A]{4,}/"; classtype:misc-activity;
rev:1;)
```

To date, this signature has not triggered on password-protected zips that contain samples sent to me from other researchers. However, this is still a 'test' rule and it should be used with care.

NETWORK WORMS, BLASTER, ETC.

SNORT is extremely useful for detecting, tracing and blocking many of the network worms that have become part of the background noise on the Internet, as well as the vast array of bot families and their numerous offspring.

EXPLOIT ME!

Although the SNORT maintainers no longer supply (or support) the 'virus.rules' signature set for the product, they do offer signatures that can be used to identify the use of most of the exploits upon which a reasonable percentage of worms, viruses and bots depend to allow them to auto-run when previewed in *Outlook*, or to get onto a system via a known exploit in, say, DCOM, LSASS or GDI.

```
alert tcp any any -> any 135 (msg:"DCOM Exploit
(MS03-026) targeting Windows XP SP1"; content:"|BA 26
E6 77 CC E0 FD 7F CC E0 FD 7F|"; classtype:attempted-
admin; sid:1100007; reference:URL,www.microsoft.com/
security/security_bulletins/ms03-026.asp;
reference:URL,jackhammer.org/rules/1100007; rev:1;)
```

BLOCKING INSTEAD

In the first part of this article I covered only the 'alert' directive, which will send an alert to the SNORT logs, Syslog, database or other configured storage options when a signature is matched.

There are other options for what action to take when a signature is matched. These include the ability to terminate the session, at either the originator or destination end of the conversation, or both at the same time.

The advantage of this is that you can stop an infection attempt dead in its tracks.

Below is an example:

```
alert tcp $EXTERNAL_NET any -> $HOME_NET any
(msg:"Backdoor.GoBot.p [KAV] - SMB"; content: "|B6 B9
ED ED CD 77 5E 11 75 1B 8B BB 01 7E 05 29 54 BF 0D B6
F0 83 7B 0C 3F 44 64 EB 96 0A 8B 72|"; classtype:
misc-activity; resp:rst_all;)
```

This would terminate the connection between the source and destination IP addresses when the signature was matched. Obviously, this 'power' should be used with

caution as it can cause problems with some applications, especially if there is a false positive problem with the signature itself.

However, using this feature effectively turns SNORT into a so-called IPS (Intrusion Prevention System), rather than an IDS (Intrusion Detection System).

SNIFFING WITH SNORT

SNORT is not just an IDS; it can also be used as a Sniffer – simply run SNORT in one of the following ways to achieve this:

```
Snort -v
```

This will show TCP/IP packet headers (TCP, UDP and ICMP) on the console.

```
Snort -vd
```

This will show TCP/IP packet headers and application data on the console.

```
Snort -vde
```

This will show TCP/IP packet headers, application data and data link layer headers on the console.

SNORT SIGS BOARD

Readers who would like access to the latest malware rule/signatures that I maintain for SNORT might like to create an account on my SNORT Sigs Board. This can be found at http://arachnid.homeip.net/cgi-bin/blah/Blah.pl. (Please note: those who do not supply the requested information when signing up will not be granted access.)

CONCLUSIONS

I hope that I have whet your appetite and shown that SNORT does indeed have its place in the anti-malware toolbox. This is increasingly true when we consider the merging of many technologies between the spammers, scammers, malware and hacking (cracking) communities.

This two-part article should not be considered as an exhaustive or complete look at SNORT. I have merely scratched the surface of the pig – and there is plenty more goodness under the crackling … dig in and pig out! (*Samson the pig appears courtesy of Farm Sanctuary, http://www.farmsanctuary.com.*)