

Using Nepenthes Honeypots to Detect Common Malware

Jamie Riden 2006-11-07

Introduction

In the past few years, a number of serious flaws in Windows have been exposed, including MS03-026 [ref 1], the flaw that Blaster [ref 2] used to spread in 2003, right up to the recent Mocabot/Wargbot worm [ref 3] which exploited MS06-040 [ref 4] from August 2006. The number of distinct pieces of malware exploiting these flaws has rapidly increased over the same time period. There are several variants of most worms and many more than that of most of the bot families, such as Agobot, Phatbot, Sdbot, and so on. As is now well-known, bots are collections of compromised "zombie" computers used together in a botnet network for nefarious purposes.

In "The Nepenthes Platform: An Efficient Approach to Collect Malware" [ref 5] Baecher *et al* note the following:

"In a four month period, we have collected more than 15,500 unique binaries, corresponding to about 1,400 MB of data. Uniqueness in this context is based on different MD5 sums of the collected binaries."

In the paper, they give detection rates for newly capture malware range between 73% and 84% across four different antivirus engines. Clearly, relying on antivirus software is not going to work for everyone, all the time.

In this paper we describe how a particular low-interaction honeypot, Nepenthes [ref 6], can be used to quickly alert an administrator to a network compromise. It captures malware and can assist in containing and removing the infection.

Useful IDS alerts for finding scanning-based worms

Some of the most damaging worms of recent years have been based on flaws in Windows services. For example, Blaster, Sasser, Welchia and Slammer have all caused large amounts of downtime and lost productivity to businesses around the world.

Hopefully, the reader's Intrusion Detection System (IDS) vendor has signatures for known worms such as these, and also has portscan detection to help discover new worms. In the case of Blaster, each infected host would send out around 10 packets every second to port 135/tcp, which was enough to trigger a Snort [ref 7] alert that discovered the problem even before the Blaster signature was created.

Some issues discovering bots

The main distinction between a "bot" and a worm is that a bot has some central control channel that issues commands to the infected computer - often this is accomplished via IRC. These "bots" can exhibit similar behaviour to the previously mentioned worms when scanning;

however, they are typically controlled via an IRC channel and will only begin scanning/exploiting on command. In this particular instance, an IDS could also have picked up the C&C traffic - successful exploitations are reported as: [SCAN]: Exploited yyy.yyy.123.45.

In general a bot will often be fairly quiet until it is ordered to scan a particular network:

```
#(4 - 1329104) [2005-03-25 03:39:49.297] [snort/2001372]
BLEEDING-EDGE IRC Trojan Reporting (Scan)
IPv4: yyy.yyy.231.32 -> zzz.zzz.163.59
      hlen=5 TOS=0 dlen=168 ID=18140 flags=0 offset=0 TTL=127 chksum=56572
TCP:  port=3023 -> dport: 8000  flags=***AP*** seq=1483308911
      ack=501861482 off=5 res=0 win=64331 urp=0 chksum=51363

Payload:  length = 128
000 : 50 52 49 56 4D 53 47 20 23 61 73 74 72 6F 20 3A  PRIVMSG #astro :
010 : 5B 53 43 41 4E 5D 3A 20 52 61 6E 64 6F 6D 20 50  [SCAN]: Random P
020 : 6F 72 74 20 53 63 61 6E 20 73 74 61 72 74 65 64  ort Scan started
030 : 20 6F 6E 20 yy yy yy 2E yy yy yy 2E 78 2E 78 3A  on yyy.yyy.x.x:
040 : 34 34 35 20 77 69 74 68 20 61 20 64 65 6C 61 79  445 with a delay
050 : 20 6F 66 20 35 20 73 65 63 6F 6E 64 73 20 66 6F  of 5 seconds fo
060 : 72 20 30 20 6D 69 6E 75 74 65 73 20 75 73 69 6E  r 0 minutes usin
070 : 67 20 32 30 30 20 74 68 72 65 61 64 73 2E 0D 0A  g 200 threads...
```

Another bot, reported by Daniel Cid [ref 8], used Google to search for potentially vulnerable Mambo installations; this would mean no portscanning would be necessary to find new targets. Therefore, as an administrator one should not rely solely on portscanning activity to find bots. The reports were made over IRC again and would have looked something like this:

```
"PRIVMSG #ch :[GOOGLE] Trying to exploit http://www.example.com/index.php"
```

Bots are not always as easy to spot as scanning worms because they can lie dormant for long periods and will only start to trigger IDS alerts when they are instructed to start spreading to other computers.

Using honeypots to find bots

With most scanning worms, and most bots, a lot of the traffic will be directed externally. In these cases, you should be able to spot the patterns of large-scale scanning in your IDS logs. However, eventually there will come a time when they scan your internal network looking for more vulnerable systems to infect.

My favourite honeypot for these purposes is known as [Nepenthes](#) [ref 6], named after a genus of pitcher plants [ref 9]. See the RAID '06 paper [ref 5] for more information. Nepenthes runs on a UNIX server and provides enough emulation of common Windows services to fool most automated attacks. Nepenthes will attempt to download the malicious payload and has an

option to submit it automatically to the Norman sandbox [ref 10]. You will then receive a report of the characteristics of the malware to your given email address.

If you run Nepenthes on an exposed machine, you will quickly discover how much malware there is floating around on the net. A lot of it is different variants of a few main families of bots: SpyBot, Agobot, and others at the time of this writing. A fair number of these may be undetected by a particular antivirus product. This won't be of interest to most people, but it can be valuable to run a Nepenthes sensor within your organization to detect worms spreading internally.

Installing and configuring Nepenthes

Those readers using Debian Linux have a pre-built package available, and in the unstable branch one can simply do a `apt-get install nepenthes`. Users of other systems can read the documentation [ref 11] which includes details on building the package. If you don't feel like building the package yourself, there are also prebuilt Debian images for VMWare [ref 12] which only require minimal extra work to install Nepenthes. (Please be kind and use the BitTorrent links if you're going to download from this site.)

Once Nepenthes is installed, consider editing `/etc/nepenthes/nepenthes.conf` and uncomment the line `"submitnorman.so", "submit-norman.conf", ""` to use the Norman sandbox. The contents of the file `submit-norman.conf` should look like this:

```
submit-norman
{
    // this is the adress where norman sandbox reports will be sent
    email    "my.email@example.com";
};
```

This will send each submission to Norman's excellent on-line sandbox, which will perform a run-time analysis and send you a copy of the results in email. This can give you very useful information on what the binary does without having to execute and trace it in your own virtual machine, or having to reverse engineering it.

When you have Nepenthes up and running, it should be listening on a large number of common TCP/IP ports, as we can see below:

```
#lsof -i
nepenthes 25917  nepenthes  6u  IPv4 162588      TCP *:smtp (LISTEN)
nepenthes 25917  nepenthes  7u  IPv4 162589      TCP *:pop3 (LISTEN)
nepenthes 25917  nepenthes  8u  IPv4 162590      TCP *:imap2 (LISTEN)
nepenthes 25917  nepenthes  9u  IPv4 162591      TCP *:imap3 (LISTEN)
nepenthes 25917  nepenthes 10u  IPv4 162592      TCP *:ssmtp (LISTEN)
...
```

Using Nepenthes

Once there is an attempt to infect the Nepenthes sensor, Nepenthes will try to download a copy of the malware and submit it to the Norman sandbox. Here is part of a report on an IRC bot:

```
[ Network services ]
* Looks for an Internet connection.
* Connects to xxx.example.net on port 7654 (TCP).
* Sends data stream (24 bytes) to remote address xxx.example.net, port 7654.
* Connects to IRC Server.
* IRC: Uses nickname xxx.
* IRC: Uses username xxx.
* IRC: Joins channel #xxx with password xxx.
* IRC: Sets the usermode for user xxx to ...
```

As you can see, this is much easier than performing a similar analysis by tracing code or reverse engineering the malware. Some malware, such as Agobot, has anti-debug code which will prevent the sandbox from doing a useful analysis. In this case, you can try your favorite antivirus engine, or if that fails it is recommended that you submit the downloaded binary to Virus Total [[ref 13](#)] which will give you reports on the binary from the leading twenty or so antivirus products.

Captured binaries are named after their md5sums, and on Debian they are found in `/var/lib/nepenthes/binaries`:

```
# ls /var/lib/nepenthes/binaries/
01a7b93e750ac9bb04c24c739b09c0b0  547765f9f26e62f5dfd785038bb4ec0b
99b5a3628fa33b8b4011785d0385766b  055690bcb9135a2086290130ae8627dc
54b27c050763667c2b476a1312bb49ea  ...
```

The log files also indicate how and where each binary is obtained:

```
# tail -1 /var/log/nepenthes/logged_submissions
[2006-07-05T20:37:52]
ftp://ftp:password@xxx.info:21/host.exe eb6f41b9b17158fa1b765aa9cb3f36a0
```

If your favorite antivirus vendor doesn't recognize the threat at this point, now is the time to submit a sample to them so they can get their definitions updated quickly. That allows you to delegate clean-up work to your technicians more easily, should the outbreak be reasonably widespread.

Results using Nepenthes

The New Zealand Honey net Project installed a Nepenthes honeypot using version 0.17 running on Debian unstable. This was listening on 255 IP addresses, a /24 network prefix. Over a period of five days, it had collected 74 different samples as distinguished by the MD5 hashes of the binaries. Of these, only 48 were identified as malware by a particular antivirus product at the end of the five day period. Of the known samples, many were worms such as Korgo, Doomjuice, Sasser and Mytob. The rest were IRC bots of one sort or another, like SDBot, Spybot, Mybot and Gobot. The majority of binaries, whether classified, as worms or bots had some kind of IRC backdoor functionality. Further analysis of these samples can also be performed by the reader as desired.

Conclusion

There are a large number of patches that have been issued for Windows which have fixed remote exploitation issues. Even with a good patch management system, some of the reader's internal hosts may be missing these patches, due to misconfiguration, human error or because they are in the process of being reinstalled. A large number of different malware binaries exist which can exploit some of these flaws to gain access to these computers. Since the source code for some of this malware is readily available to blackhats, a great many variations exist and not all are detected by common antivirus software.

A low-interaction honeypot like Nepenthes is easy to install and requires minimal maintenance. It may provide valuable information in the event of an infection within your organisation. When used in conjunction with an Intrusion Detection System, valuable information about the behavior of the malware, packet captures and the malware binary itself may be obtained.

References

[ref 1] Microsoft Security Bulletin [MS03-026](#), "Buffer Overrun in RPC Interface Could Allow Code Execution."

[ref 2] Description of the Blaster worm, www.symantec.com/security_response/writeup.jsp?docid=2003-081113-0229-99.

[ref 3] Description of the Mocabot/Wargbot worm, www.symantec.com/security_response/writeup.jsp?docid=2006-081312-3302-99.

[ref 4] Microsoft Security Bulletin [MS06-040](#), "Vulnerability in Server Service Could Allow Remote Code Execution."

[ref 5] RAID '06 paper, "The Nepenthes Platform: An Efficient Approach to Collect Malware" by Baecher *et al*, honeyblog.org/junkyard/paper/collecting-malware-final.pdf.

[ref 6] Nepenthes homepage, nepenthes.mwcollect.org.

[ref 7] Snort open-source IDS homepage, www.snort.org.

[ref 8] Daniel Cid's post about Mambo scans on the Incidents mailing list, securityfocus.com/archive/75/433959/30/0/threaded.

[ref 9] Wikipedia document about the origin of the Nepenthes name: a genus of pitcher plants, en.wikipedia.org/wiki/Nepenthes.

[ref 10] Norman sandbox, which receives and analyses malicious payloads, sandbox.norman.no.

[ref 11] Nepenthes documentation, nepenthes.mwcollect.org/documentation.

[ref 12] Prebuilt Debian images for VMWare, www.thoughtpolice.co.uk/vmware.

[ref 13] Virus Total, free service for scanning binaries with multiple antivirus products, www.virustotal.com.

About the author

Jamie Riden is a member of the [New Zealand Honey net Project](#).

Reprints or translations

Reprint or translation requests require [prior approval](#) from SecurityFocus.

© 2006 SecurityFocus

Comments?

Public comments for Infocus articles, below, require technical merit to be published. General comments, article suggestions and feedback are encouraged but should be sent to the [editorial team](#) instead.

[Privacy Statement](#)

Copyright 2006, SecurityFocus