eEye® Digital Security

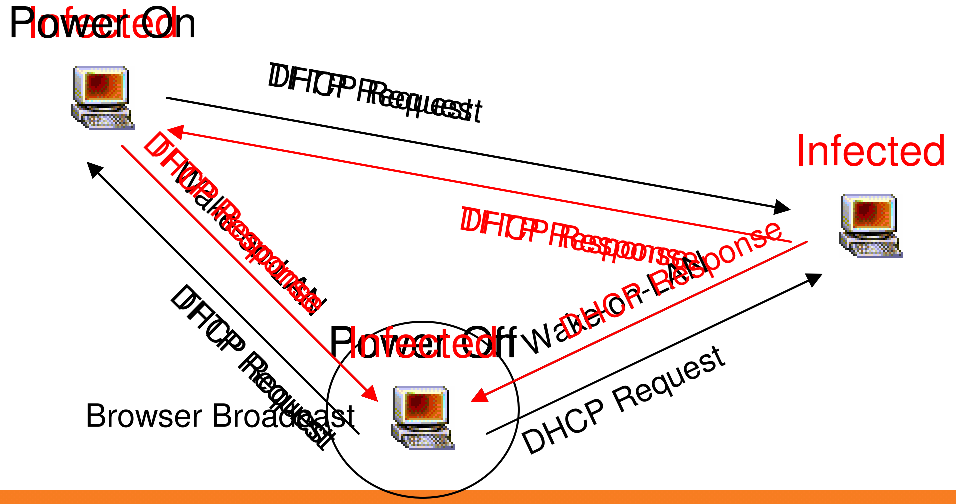
PiXiE: A Self-Propagating Network Boot Virus for Windows

Derek Soeder
Software Engineer, eEye Digital Security

February 2006

What is PiXiE?

- **About PiXiE**
  - Proof-of-concept (harmless) virus
  - Spreads to Windows 2000+ systems via network boot
  - Sends code to BIOS PXE agent of booting systems
  - Activates powered-off systems using Wake-on-LAN



Overview

- **Stage 1: Bootstrap / Kernel Code**
  - Based on eEye BootRoot v2.0
  - Executes before Windows; infiltrates kernel as it loads
  - Hooks NDIS.SYS to sniff network traffic

Starting Wind

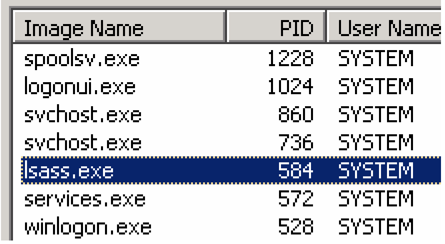


Image Name	PID	User Name
spoolsv.exe	1228	SYSTEM
logonui.exe	1024	SYSTEM
svchost.exe	860	SYSTEM
svchost.exe	736	SYSTEM
lsass.exe	584	SYSTEM
services.exe	572	SYSTEM
winlogon.exe	528	SYSTEM

- **Stage 2: User-Mode DLL**
  - Injected into a system process by kernel-mode code
  - Hosts viral DHCP and TFTP servers for network boot
  - Sends Wake-on-LAN packets to systems that shut down

eEye BootRoot: Background

- **Bootstrap code that subverts Windows NT-family kernel**
  - Presented at Black Hat USA 2005
  - First known public implementation of concept
  - eEye BootRoot v1.0
    - Step 1: Patch OSLOADER as it loads by hooking INT 13h (Disk)
    - Step 2: Traverse loaded boot driver list to patch kernel / drivers

- **BootRootKit v1.0**
  - Uses eEye BootRoot v1.0 techniques
  - Hooks NDIS.SYS to execute kernel code from packets with a specific signature

```
*** STOP: 0x0000001E (0xC00000
KMODE_EXCEPTION_NOT_HANDLED

*** Address 8047179C base at 8
```

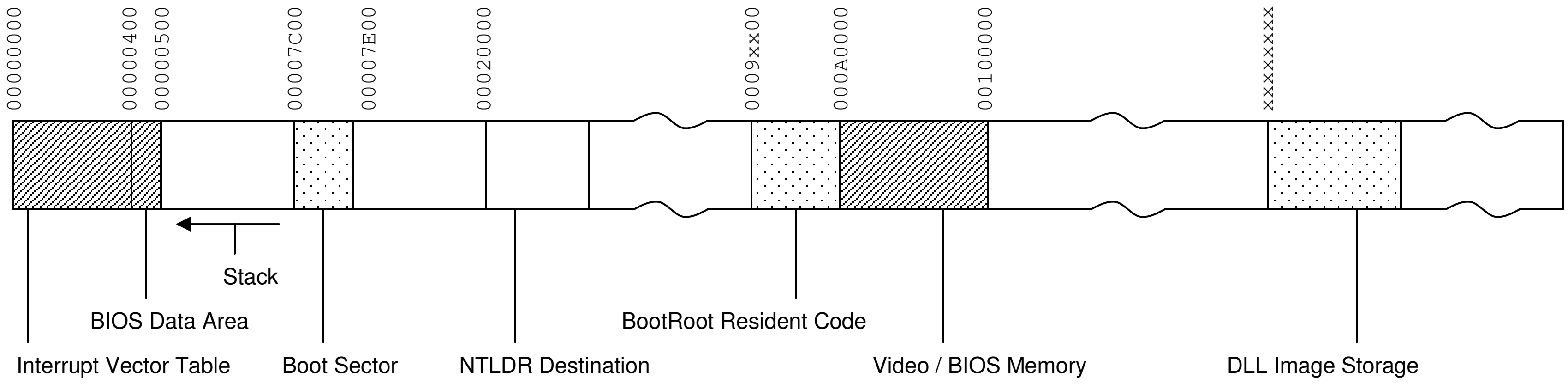
eEye BootRoot v2.0: Features

- **Compatibility and robustness**
  - No specific byte signatures or version-dependent structures
  - Only uses kernel APIs supported by NT4/2000/XP/2003
  - Compensates for buggy BIOSes that misreport conventional memory limit from INT 15h/AX=E820h

- **Showcases fun technology**
  - Pure memory (no file) DLL injection from kernel
  - NTOSKRNL export lookup using 8-bit name hashes
  - Disassembler engine for function entry point hooking
  - Hides physical memory with INT 15h hook

eEye BootRoot v2.0: Overview (1)

- **Phase 1: Bootstrap Code**
  - Reserves conventional memory
  - Makes modified system memory map to reserve memory
  - Loads DLL into reserved memory
  - Hooks INT 13h to modify image sizes on load
  - Hooks INT 15h to serve up modified memory map
  - Executes hard drive Master Boot Record

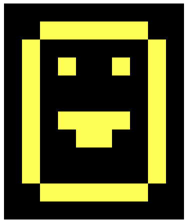


eEye BootRoot v2.0: Overview (2)

- **Phase 2: INT 15h Hook**
  - Provides NTLDR with a modified memory map
  - Also hooks "`LIDT [ofs32]`" instructions in OSLOADER code (loaded immediately after NTLDR)
    - Simple and generic-ish way to retain control across switch to protected mode
    - Allows us to modify IDT before it takes effect
- **Phase 3: LIDT Hook**
  - Hooks INT 0Dh (General Protection Fault) before doing LIDT
  - Sets code descriptor (GDT#0008h) limit = 0x7FFFFFFF
    - Allows us to catch transfer to NTOSKRNL entry point

eEye BootRoot v2.0: Overview (3)

- **Phase 4: INT 0Dh (#GP) Hook**
  - Restores CS descriptor limit = 0xFFFFFFFF
  - Searches module list for NTOSKRNL
    - OSLOADER’s _BlLoaderBlock is entry point’s stack argument
  - Expands last section of NTOSKRNL and copies in our code
  - Looks up imports from NTOSKRNL
  - Hooks MmMapViewOfSection and PspCreateThread
  - Displays yellow smiley
  - Resumes execution of NTOSKRNL entry point



eEye BootRoot v2.0: Overview (4)

- **Phase 5: PspCreateThread hook**
  - Located by scanning PsCreateSystemThread for “`CALL rel`”
  - Activates when first thread is created in target process
    - Finds process name offset by searching System Process object for “System” string
    - Checks VM_COUNTERS.QuotaPeakNonPagedPoolUse from NtQueryInformationProcess(ProcessVmCounters) to determine if this is first thread in process
    - If so...

eEye BootRoot v2.0: Overview (5)

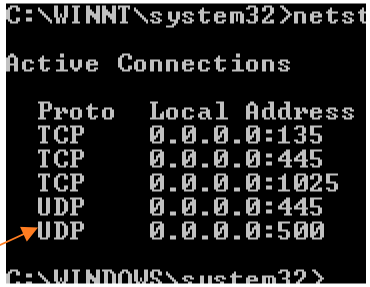
- **Phase 5a: DLL Injection**
  - Creates “\KnownDlls\XXXXXXXX.dll” memory section
    - Where “XXXXXXXX” is hexadecimal address of Process object
    - Creates and maps temporary view of section
    - Manually maps and copies DLL from physical memory into view
  - Allocates memory and copies in DLL injection code
    - Calls NTDLL.DLL!LdrLoadDll(“XXXXXXXX.dll”) to take advantage of native loader code (does imports, relocations, etc.)
    - LdrpMapDll tries to open “\KnownDlls\___.dll” section before accessing file (e.g., “\WINNT\system32\___.dll”), for performance
  - Hijacks EIP in new thread’s context
    - Originally pointed to EXE entry point or BaseProcessStartThunk
    - Now it points to our DLL injection code

eEye BootRoot v2.0: Overview (6)

- **Phase 6: MmMapViewOfSection hook**
  - If Section object is “\KnownDlls\XXXXXXXX.dll”:
  - Changes ‘Protect’ argument from PAGE_READWRITE to PAGE_EXECUTE_READWRITE
    - We must force +X since this is not a real SEC_IMAGE section
  - Invokes original MmMapViewOfSection
  - If STATUS_SUCCESS is returned, changes return value to STATUS_IMAGE_NOT_AT_BASE
    - This forces NTDLL loader to apply relocations

PiXiE: Kernel Code

- **“Kernel code” includes boot loader code as well**
- **Basically BootRoot v2.0, except:**
  - Hooks NDIS.SYS!ethFilterDprIndicateReceivePacket to sniff network traffic for Browser broadcasts
  - Communicates MAC addresses of powering-down hosts to user-mode DLL via memory section
  - Target process is “LSASS.EXE”
    - Starts early in boot sequence
    - Required for proper system operation
    - Always unique
    - Loads Winsock and hosts servers (ISAKMP, LDAP, etc.) normally



```
C:\WINNT\system32>netst

Active Connections

  Proto  Local Address
  TCP    0.0.0.0:135
  TCP    0.0.0.0:445
  TCP    0.0.0.0:1025
  UDP    0.0.0.0:445
  UDP    0.0.0.0:500

C:\WINDOWS\system32>
```

PiXiE: User-Mode DLL

- **Hosts majority of viral code**
  - Starts DHCP and TFTP servers as soon as possible
    - DHCP server answers requests asking for “Boot File Name”; other requests are ignored so real DHCP server can answer
    - TFTP server sends back PiXiE kernel code + DLL as requested file
  - Periodically polls list of shutting-down MAC addresses
    - Sends Wake-on-LAN packet for MAC address until a DHCP request is received, or entry becomes “stale”

- **Not too interesting technically, so...**

Demonstration

Let's see it in action!

- **One infected host on LAN...**
- **Another host attempts to use network boot...**
- **Another host powers down and is awakened...**

Questions?

E-mail me: dsoeder@eeye.com