

Implementing Rootkits to Address Operating System Vulnerabilities

Prof. Basie Von Solms and Manuel Corregedor



UNIVERSITY
OF
JOHANNESBURG
Academy of Computer Science
and Software Engineering

Introduction

- Are current anti-malware products working?



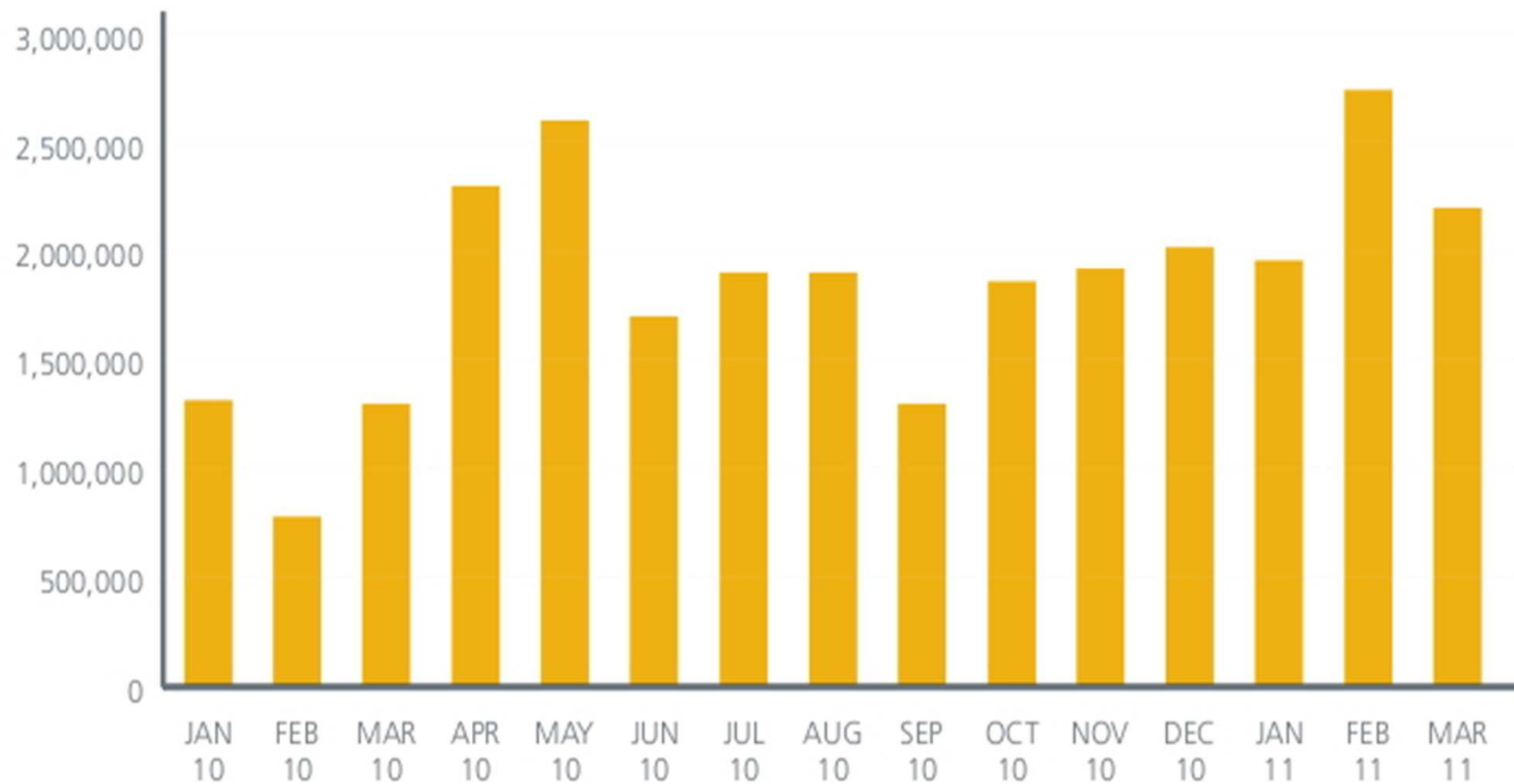
Introduction (Cont.)

- Statistics taken from Microsoft Security Intelligence Report Volume 10
 - detections and removals by Microsoft desktop anti-malware products in 2010 (Infection rates)

	Country/Region	1Q10	2Q10	3Q10	4Q10	Change from 3Q to 4Q	Change in 2010
1	United States	11,025,811	9,609,215	11,340,751	11,817,437	4.2% ▲	7.2% ▲
2	Brazil	2,026,578	2,354,709	2,985,999	2,922,695	-2.1% ▼	44.2% ▲
3	China	2,168,810	1,943,154	2,059,052	1,882,460	-8.6% ▼	-13.2% ▼
5	United Kingdom	1,490,594	1,285,570	1,563,102	1,857,905	18.9% ▲	24.6% ▲
4	France	1,943,841	1,510,857	1,601,786	1,794,953	12.1% ▲	-7.7% ▼
7	Korea	962,624	1,015,173	1,070,163	1,678,368	56.8% ▲	74.4% ▲
6	Spain	1,358,584	1,348,683	1,588,712	1,526,491	-3.9% ▼	12.4% ▲
9	Russia	700,685	783,210	928,066	1,311,665	41.3% ▲	87.2% ▲
8	Germany	949,625	925,332	1,177,414	1,302,406	10.6% ▲	37.1% ▲
10	Italy	836,593	794,099	900,964	998,458	10.8% ▲	19.3% ▲

Introduction (Cont.)

- New malware samples discovered (Jan 2010 – March 2011)



Introduction (Cont.)

- Why current anti-malware products are not working
 - Code Obfuscation
 - Metamorphic Malware
 - Polymorphic Malware
 - Exploits and Vulnerabilities in software
 - Social Engineering
 - It gets worse..



Rootkits

- Intended purpose
 - Hide itself (long term) and other malware
 - Hide malicious activities
 - Is that all?
- Privilege Level
 - Kernel mode vs User Mode

Rootkits

- Techniques used by rootkits
 - Patching (Run-time or binary)
 - Hooking call tables (alter kernel control flow)
 - User Mode (Import Address Table)
 - Kernel Mode (IDT, GDT, SYSENTER, SSDT, IRP Dispatch Table)
 - DKOM

Why Implement Rootkits?

- White Hat Perspective
 - Implementing helps with understanding
 - Identify the vulnerabilities that rootkits exploit (current and new ones)
 - Prevent or minimize the vulnerabilities

Our Rootkits

- Two rootkits implemented
 - Evader and Sabotager
 - Target OS
 - Windows XP Professional (32 bit)
 - Windows 7 Professional (32 bit)
 - Target Demographic
 - Average home user
 - Architecture
 - Hybrid
 - Objective is not stealth but instead to cause damage!



Rootkit Development Tools

- Rootkits – kernel mode drivers
- Tools freely available and supported:
 - Windows Driver Kit (kernel mode drivers)
 - Windows SDK (User mode components)
 - Sysinternals Suite (Debugview)
 - Windows Debugging Tools (windDbg, KD)

Steps for Implementing “Useful” Rootkits

- Install Development Tools
- Get into the Kernel
- Choose Rootkit(s) Installation Method
- Manipulate Kernel Structures

Getting into the kernel: 3 Ways

- Service Control Manager (SCM)
- Using the system call ZwSetSystemInformation
 - Exported by ntdll.dll
- Injecting code into the kernel



Getting into the kernel (Cont.)

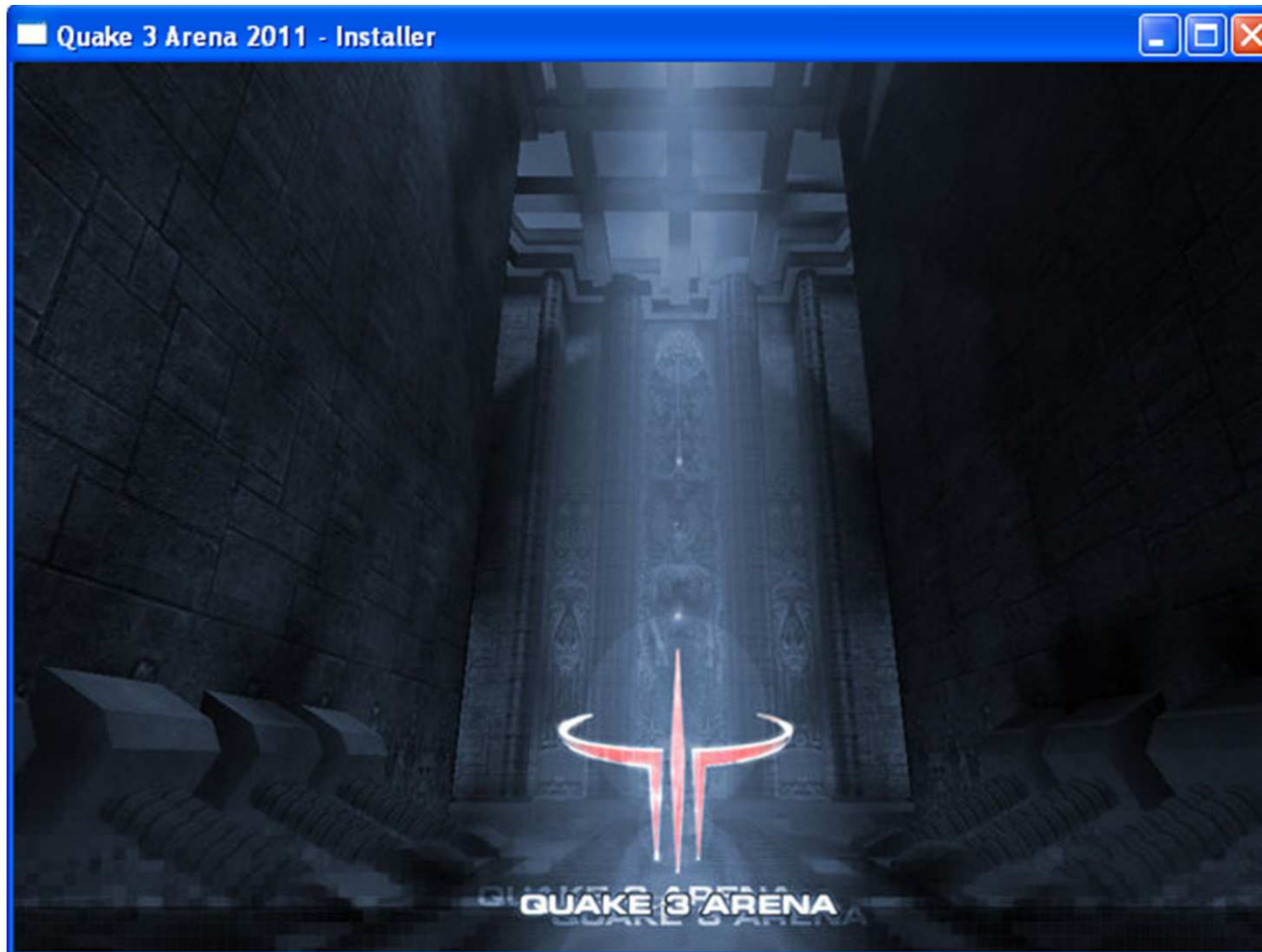
- Chosen method for our implementation
 - SCM
 - Load using system boot loader
 - Disguise rootkits
 - Name: msusb.sys
 - Directory: %windir%\system32\Drivers
 - Description: SDDL subsystem for Windows
USB Resource – Microsoft(C)
 - » Security Descriptor Definition Language
 - » USB



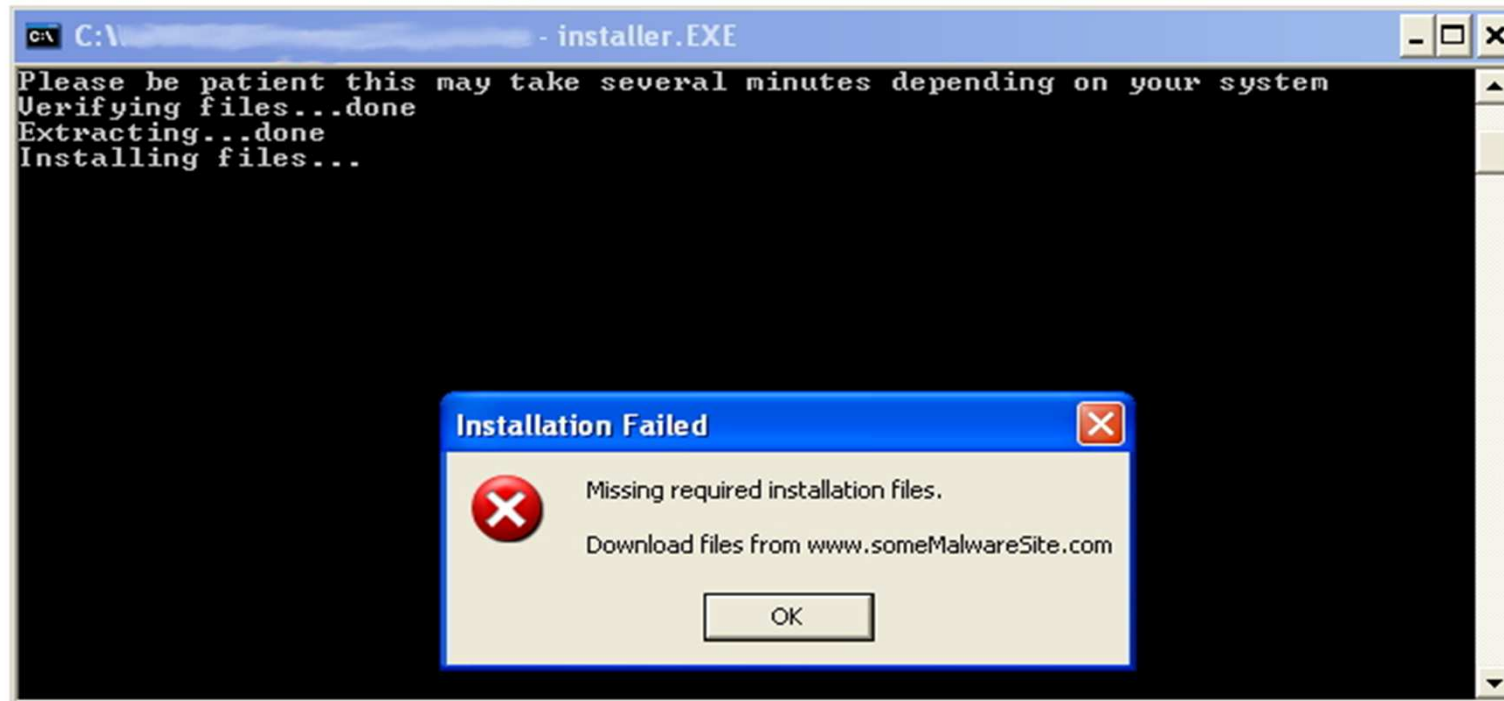
Installation of the rootkits

- Manipulate other binaries or the OS to execute the installer
- Social Engineering
 - Installer for a game
 - Avoid suspicion

Installation of the rootkits (Cont.)



Installation of the rootkits (Cont.)



Hooking system routines

- Import routines to hook from existing system modules
 - During compilation get address
- Modify SSDT
 - But its write protected
 - No problem we have kernel privileges
 - Disable write protection (CR0 Processor Register 17th bit)
- Hook routines
- Evader Rootkit uses DKOM



Sabotager Rootkit

- Goal: Cause the OS to generate an error or a bug check (BSOD) every time the computer reboots
- How?
 - Hook ZwSetValueKey (Session Manager)
 - Disable Recovery Features
 - Boot from last known configuration
 - Safe Mode (loaded by system boot loader)
 - Windows 7 Startup Repair (can launch system restore)
 - System Restore
 - User mode component uses SrClient.dll

Sabotager Rootkit (Cont.)

- Results/Error Messages:
 - Windows XP Professional (32 bit) – Normal Boot
 - Lsass.exe is the local security authority subsystem



Sabotager Rootkit (Cont.)

- Results/Error Messages (Cont.):
 - Windows XP Professional (32 bit) and Windows 7 Professional 7 (32 bit) – Safe Mode
 - Generates Bug Check

```
A problem has been detected and windows has been shut down to prevent damage to your computer.
```

```
BAD_POOL_CALLER
```

```
If this is the first time you've seen this stop error screen, restart your computer. If this screen appears again, follow these steps:
```

```
Check to make sure any new hardware or software is properly installed. If this is a new installation, ask your hardware or software manufacturer for any windows updates you might need.
```

```
If problems continue, disable or remove any newly installed hardware or software. Disable BIOS memory options such as caching or shadowing. If you need to use safe mode to remove or disable components, restart your computer, press F8 to select Advanced Startup options, and then select safe mode.
```

```
Technical information:
```

```
*** STOP: 0x000000C2 (0x00000007,0x00^
```



Sabotager Rootkit (Cont.)

- Results/Error Messages (Cont.):
 - Windows 7 Professional (32 bit) – Normal Boot
 - Generates Bug Check

```
A problem has been detected and windows has been shut down to prevent damage
to your computer.

The video driver failed to initialize

if this is the first time you've seen this stop error screen,
restart your computer. If this screen appears again, follow
these steps:

Check to make sure any new hardware or software is properly installed.
if this is a new installation, ask your hardware or software manufacturer
for any windows updates you might need.

If problems continue, disable or remove any newly installed hardware
or software. Disable BIOS memory options such as caching or shadowing.
If you need to use Safe Mode to remove or disable components, restart
your computer, press F8 to select Advanced Startup options, and then
select Safe Mode.

Technical information:

*** STOP: 0x000000B4 (0x85EE8D20,0x8621B000,0x8621E000,0x00000000)

Collecting data for crash dump ...
Initializing disk for crash dump ...
Beginning dump of physical memory.
Dumping physical memory to disk: 70
```



Sabotager Rootkit (Cont.)

- Practical Uses
 - Extortion
 - Ransomware

Evader Rootkit

- Goals: Disable anti-malware software and log keys
- How to log keys pressed
 - Use Microsoft's layered driver model architecture to Intercept I/O Request Packets
 - How to store keys logged?
 - Dispatch level vs Passive Level
 - Two threads one memory location

Evader Rootkit (Cont.)

- How to disable anti-malware software
 - Hook ZwQuerySystemInformation routine (undocumented)
 - Intercept system process information structures
 - Iterate through system process information structures
 - Compare process names to stored list
 - If match found use process ID to terminate process
 - Keep looping
 - Problem with privilege levels (Dispatch and Passive)
 - Use two threads one memory location



Evader Rootkit (Cont.)

- Disabling anti-malware software setup
 - Windows XP Professional 32 bit
 - All products fully updated (11 March 2011)
 - Testing done using EICAR test file



Evader Rootkit (Cont.)

- Disabled following anti-malware products:
 - Avira AntiVir Personal
 - AVG Internet Security 2011
 - BitDefender Total Security 2011
 - F-Secure Internet Security 2011
 - Avast! Free Antivirus
 - Microsoft Security Essentials

Evader Rootkit (Cont.)

- Avira AntiVir Personal detected rootkit automatically
 - Used generic detection routine
 - Scanned automatically due to .sys extension
 - Change extension
- Practical uses
 - Steal confidential information using key logger
 - Install other malware after disabling anti-malware program

Virustotal Results

Scanned on the 17th of April 2011:

A total of 42 anti-malware programs were used namely:

AhnLab-V3, AntiVir, Antiy-AVL, Avast, Avast5, AVG, BitDefender, CAT-QuickHeal, ClamAV, Commtouch, Comodo, DrWeb, Emsisoft, eSafe, eTrust-Vet, F-Prot, F-Secure, Fortinet, Gdata, Ikarus, Jiangmin, K7AntiVirus, Kaspersky, McAfee, McAfee-GW-Edition, Microsoft, NOD32, Norman, nProtect, Panda, PCTools, Prevx 3.0, Rising, Sophos, SUPERAntiSpyware, Symantec, TheHacker, TrendMicro, TrendMicro-HouseCall, VIPRE, ViRobot and VirusBuster.



Vulnerabilities Identified

- Kernel Security
 - Hooking easy/disabling write protection
 - All code in kernel is equally trusted/has same privilege levels
- Sharing memory
 - Lower privilege level process can read memory locations written to by higher privilege level processes



Vulnerabilities Identified (Cont.)

- Windows Registry
 - Windows central hierarchy db for software, user profiles etc.
- Windows boot loader
 - Loaded rootkit before session manager



Vulnerabilities Identified (Cont.)

- System messages
 - Bug checks and errors are misleading/not helpful
 - No warning given to user that all restore points were being deleted.



Vulnerabilities Identified (Cont.)

- System messages (Cont.)
 - No warnings given to users regarding suspicious activity E.G. Sabotager Rootkit output in debugger DebugView (after update):
 - “A driver is mapping physical memory 0001F000->0001FFFF that it does not own. This can cause internal CPU corruption. A checked build will stop in the kernel debugger so this problem can be fully debugged.”
- The User

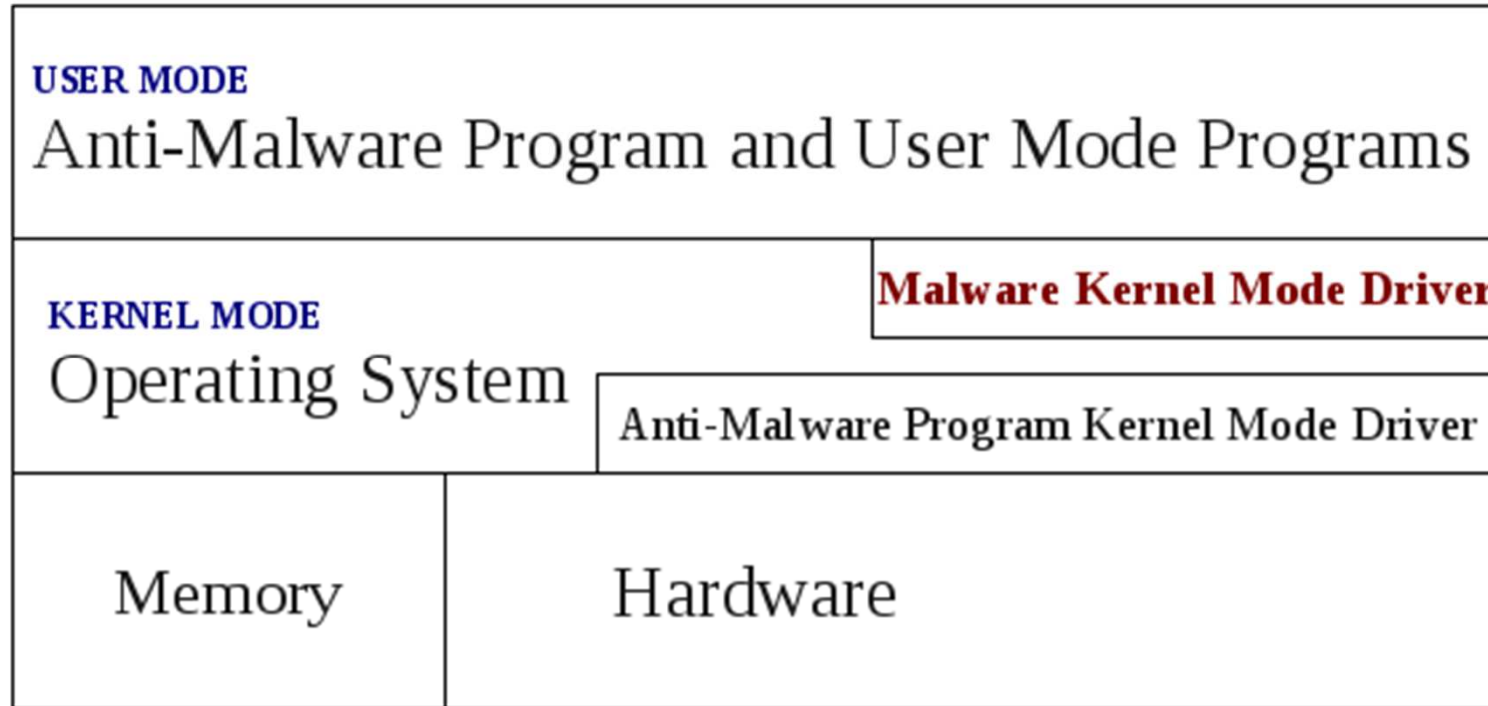


64 bit Versions of Windows

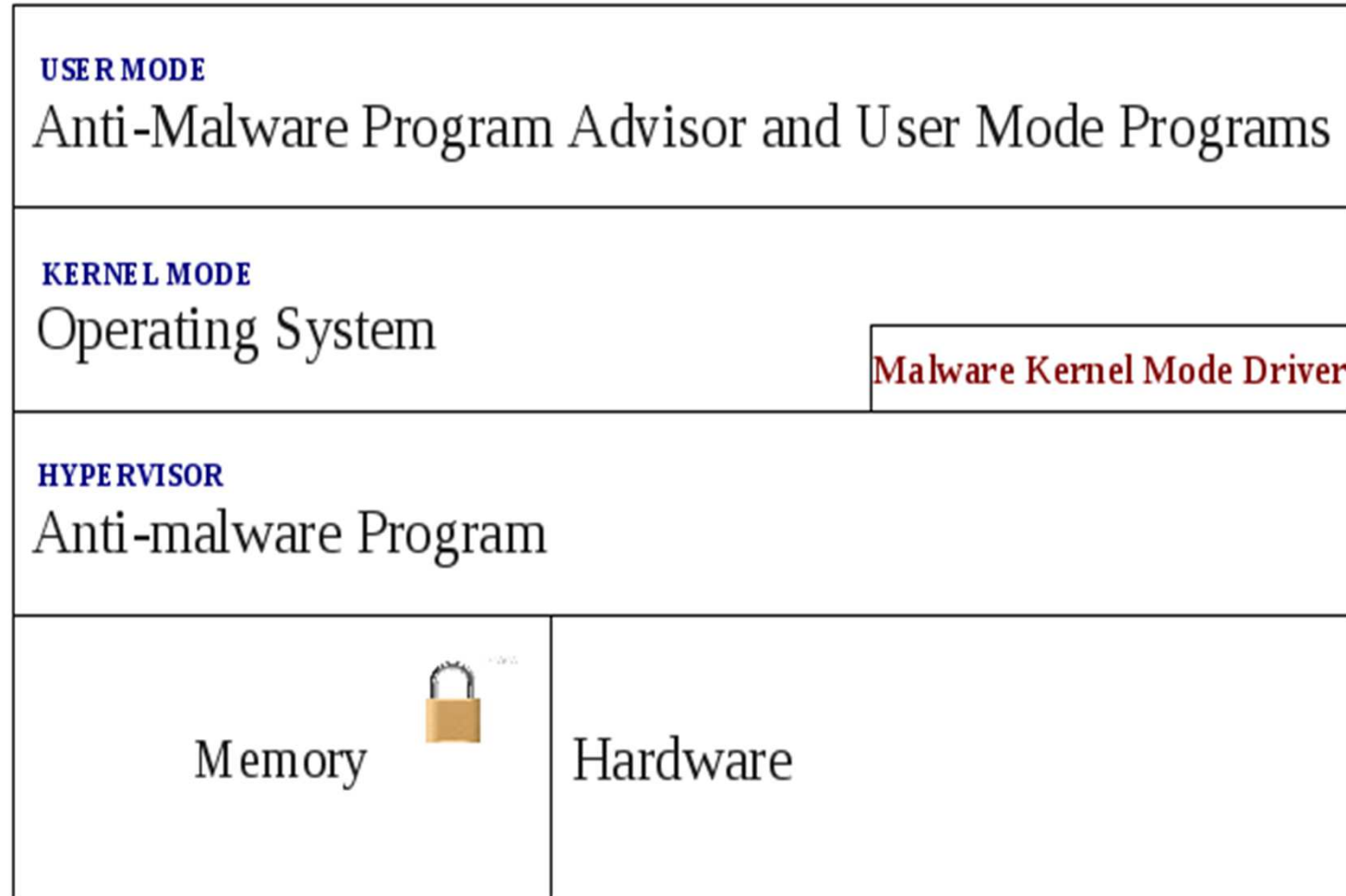
- Vulnerabilities Addressed in 64 bit Versions of Windows
 - Drivers must be digitally signed
 - Kernel Patch Protection aka Patch Guard prevents:
 - Modification of system service tables e.g. SSDT
- Bypassing new 64 bit improvements
 - Driver Signing: front company, exploit vulnerabilities in signed drivers or steal signing certificates
 - Patch Guard runs in the kernel therefore can be subverted (as already shown).



Current anti-malware architectures



Proposed anti-malware architecture



To Summarise

- Developing malware can help:
 - Better understand the malware
 - Identify the current vulnerabilities exploited
 - Identify new vulnerabilities
 - Focus on prevention rather than detection

Questions?

