Advanced Malware Trends

September 22 – 24, 2008

Michael Berg <mjberg@sandia.gov>
Senior Member of Technical Staff
Sandia National Laboratories

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company,
for the United States Department of Energy under contract DE-AC04-94AL85000.

Sandia National Laboratories

Purpose

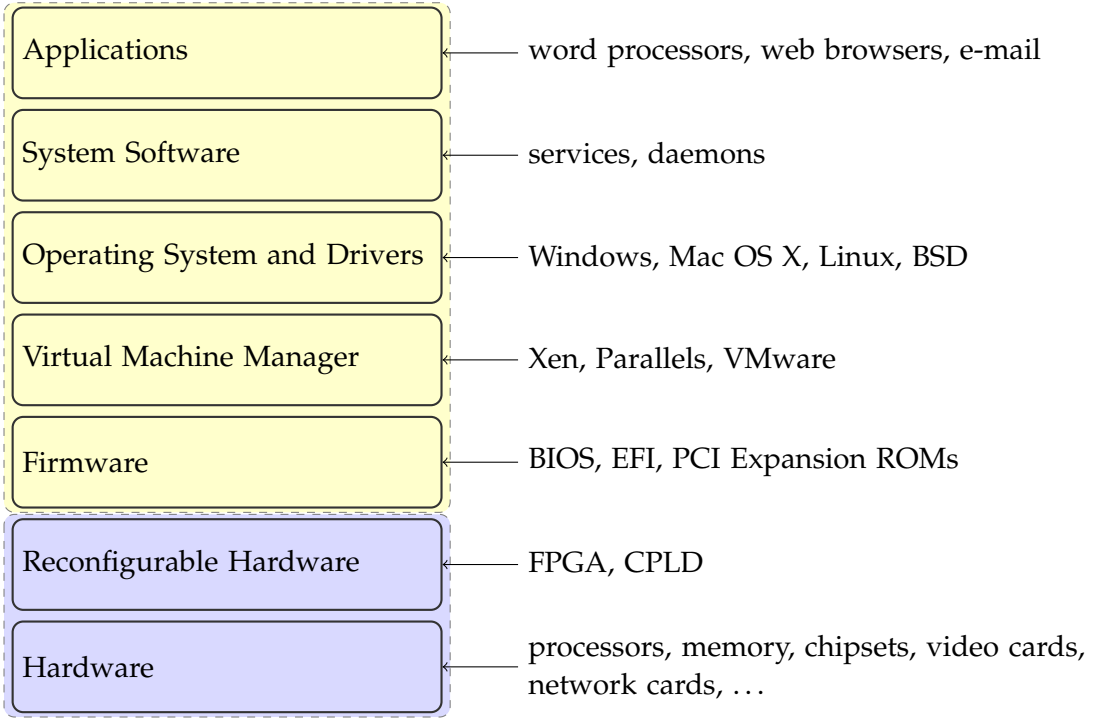
This presentation will cover:

- An overview of computer technologies and trends
- The trend of malware moving into lower-level software and hardware
- The evolving skill-set needed by information assurance (IA) professionals

In the hope of:

- Informing people about emerging threats to information systems
- Inspiring new ideas for protecting information systems

The Hardware/Software Stack



Computer Architecture Trends (1)

- Migration from shared busses to point-to-point connections
  - Allows higher-speed communication
  - PCI (shared bus segments) → PCI Express (point-to-point)
  - Parallel ATA (shared bus) → Serial ATA (point-to-point)
- Migration from parallel interfaces to serial interfaces
  - Allows higher-speed communication
  - Simplifies connectors and cabling
  - Smaller form factors
  - PCI (parallel) → PCI Express (serial)
  - Parallel ATA → Serial ATA
- Migration from discrete signaling to frame-/packet-based communication
  - PCI Express: Physical, Data Link, and Transport Layer Packets (PLP, DLLP, TLP)
  - Serial ATA: Frame Information Structure (FIS)

Internal computer connectors and protocols now look like network connectors and protocols. Attackers have years of experience with network protocols.

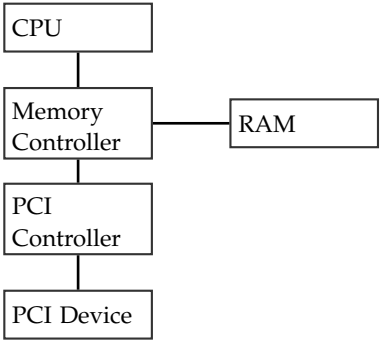
Computer Architecture Trends (2)

- Embedded microcontrollers with associated flash storage
  - "Intelligent end-points"
  - Mobile devices
- Reconfigurable hardware
  - Upgrade deployed hardware designs
  - Function-agile hardware in space-constrained environments
- Enterprise management of hardware and software
  - Centralized management of patches within corporations
  - Automated inventory of network connected systems
- Virtual machines
  - Development and testing environments
  - More efficient use of server resources
  - Reduce physical server count
  - Separate security contexts on a system

New architectures create new opportunities for defenders AND attackers.

Direct Memory Access (DMA)

- Allows hardware components to read and write system memory independently of the CPU
- Frees the CPU to perform other tasks while a DMA transfer occurs



Trusted Platform Module (TPM)

Specialized microcontroller and flash storage that provides:

- Hardware-based random number generator
- Protected generation and storage of cryptographic keys
- "Attestation" of hardware, boot, and OS configuration
- "Sealing" (encrypting) data based on attestation values

See `https://www.trustedcomputinggroup.org/` for full details.

Intel Active Management Technology (AMT)

http://www.intel.com/technology/platform-technology/intel-amt/index.htm
http://softwarecommunity.intel.com/articles/eng/1032.htm

Special microcontroller embedded in the north bridge with interfaces to the:

- CPU
- Memory Management Unit (MMU)
- Network card
- Other parts of the system

Intel's AMT architecture was designed to facilitate:

- Automated inventory of computer hardware plugged into the network
- Automated inventory of software on computers
- Deployment of patches and anti-virus updates to computers
- Quarantining malware-infected computers with their own network card

History: Open Publication to Deployment (1)

Stack Overflow Vulnerabilities:

- 1988 Deployment: Morris worm
- 1996 Publication: **"Smashing the Stack for Fun and Profit"**, "Aleph One"
- 1998 Deployment: Linux/ADM worm
- 2001 Deployment: CodeRed worm

Heap Overflow Vulnerabilities:

- 1999 Publication: **"Heap Overflow Tutorial"**, "w00w00"
- 2001 Mixed: **"Vudo"**, Michael "MaXX" Kaempf
- 2002 Deployment: Linux/Slapper worm

Format String Vulnerabilities:

- 1999 Publication: **"Exploit for proftpd 1.2.0pre6"**, Tymm Twillman
- 2000 Publication: **"WU-FTPD Remote Format String Stack Overwrite Vulnerability"**, "tf8"
- 2000 Deployment: WU-FTPD and LPRng exploits

History: Open Publication to Deployment (2)

SQL Injection Vulnerabilities:

- 1998 Publication: **"NT Web Technology Vulnerabilities"**, "rain.forest.puppy"
- 2001 Deployment: data exfiltration reports

There seems to be a 1 – 3 year "gestation" period from open publication of a new vulnerability category to widespread use of that vulnerability category by attackers "in the wild".

- Developing and refining the new exploit techniques
- Incorporating the new techniques into malware toolchains
- Testing

2003-2004: DMA Used to Protect a System

"A Hardware-Based Memory Acquisition Procedure for Digital Investigations"
Brian D. Carrier, Joe Grand, 2003.
`http://www.digital-evidence.org/papers/tribble-preprint.pdf`

- Describes hardware-based procedures for capturing forensic images of volatile system memory
- Proof-of-concept PCI device called "Tribble" uses DMA to read (image) the host's memory

"Copilot - a Coprocessor-based Kernel Runtime Integrity Monitor"
Nick L. Petroni, Timothy Fraser, Jesus Molina, William A. Arbaugh, University of Maryland, 2004.
`http://www.usenix.org/events/sec04/tech/full_papers/petroni/petroni.pdf`

- Prototype PCI device that uses DMA to track MD5 hashes of:
  - OS kernel
  - Loaded modules (drivers)
  - Critical data structures
- Changes from "known good" values are reported out-of-band to an administration station

2005: DMA Used to Compromise a System

"FireWire: all your memory are belong to us"
Maximillian Dornseif, 2005.
`http://md.hudora.de/presentations/firewire/2005-firewire-cansecwest.pdf`

- Demonstrates the lack of FireWire/IEEE1394 DMA filtering by using an attached device (an iPod) to read from and write to system memory
  - Read screen contents
  - Search for strings or keys
  - Change data
  - Escalate privileges of a process
  - Inject code into the system
- Discusses capturing forensic images of memory via FireWire/IEEE1394

We haven't seen widespread use of DMA attacks for two reasons:

- Many operating systems now configure the IEEE1394 controllers to prevent this attack
- This type of DMA attack requires physical access – which attackers avoid where possible

2006: Malicious Virtual Machine Managers

"SubVirt: Implementing malware with virtual machines"
Samuel T. King, Peter M. Chen, Yi-Min Wang, Chad Verbowski, Helen J. Wang, Jacob R. Lorch, 2006.
`http://www.eecs.umich.edu/~pmchen/papers/king06.pdf`

"Hardware Virtualization Rootkits"
Dino A. Dai Zovi, Matasano, 2006.
`http://www.blackhat.com/presentations/bh-usa-06/BH-US-06-Zovi.pdf`

"Subverting Vista Kernel for Fun and Profit"
Joanna Rutkowska, COSEINC Advanced Malware Labs, 2006.
`http://www.blackhat.com/presentations/bh-usa-06/BH-US-06-Rutkowska.pdf`

Basic sequence of events:

1. Run malicious virtual machine manager/monitor (VMM)
2. Migrate the running OS into a virtual machine (original OS is now a "guest" OS)
3. Malicious VMM provides backdoor functionality that is invisible to the guest OS

If the publication to deployment trends hold, we could start to see malicious VMMs "in the wild" around 2009.

2006-2007: Malware Targets Device Firmware

"Implementing and Detecting an ACPI BIOS Rootkit"
John Heasman, Next-Generation Security (NGS) Software, 2006.
`http://www.blackhat.com/presentations/bh-federal-06/BH-Fed-06-Heasman.pdf`

"Implementing and Detecting a PCI Rootkit"
John Heasman, NGS Software, 2006.
`http://www.ngssoftware.com/research/papers/Implementing_And_Detecting_A_PCI_Rootkit.pdf`

"Hacking the Extensible Firmware Interface"
John Heasman, NGS Software, 2007.
`http://www.ngssoftware.com/research/papers/BH-VEGAS-07-Heasman.pdf`

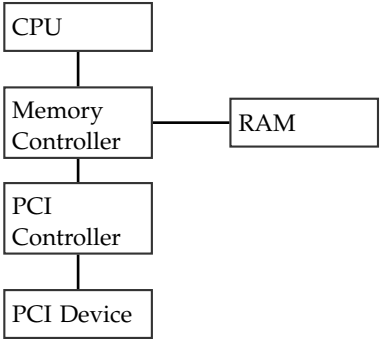
- Describes modification of boot-related firmware to bootstrap a rootkit before the OS loads
- Modifications to boot-related firmware should be detected by the TPM (if activated)

If the publication to deployment trends hold, we could start to see malicious firmware "in the wild" around 2010.

2007: Blocking DMA to Hide Malware

"Beyond the CPU: Defeating Hardware Based RAM Acquisition"
Joanna Rutkowska, COSEINC Advanced Malware Labs, 2007.
`http://www.blackhat.com/presentations/bh-dc-07/Rutkowska/Presentation/bh-dc-07-Rutkowska-up.pdf`

- The memory controller handles both DMA and memory mapping
- The memory controller handles access from the CPU and PCI devices in slightly different ways
- Malware configures registers to redirect DMA for certain address ranges back out to PCI space
- Malware can hide in these "masked" regions of memory



If the publication to deployment trends hold, we could start to see malware hiding via DMA blocking "in the wild" around 2010 – but only if DMA blocking is actually necessary.

2008: Malicious Hardware

"Designing and implementing malicious hardware"
Samuel King, Joseph Tucek, Anthony Cozzie, Chris Grier, Weihang Jiang, Yuanyuan Zhou, 2008.
`http://www.usenix.org/event/leet08/tech/full_papers/king/king.pdf`

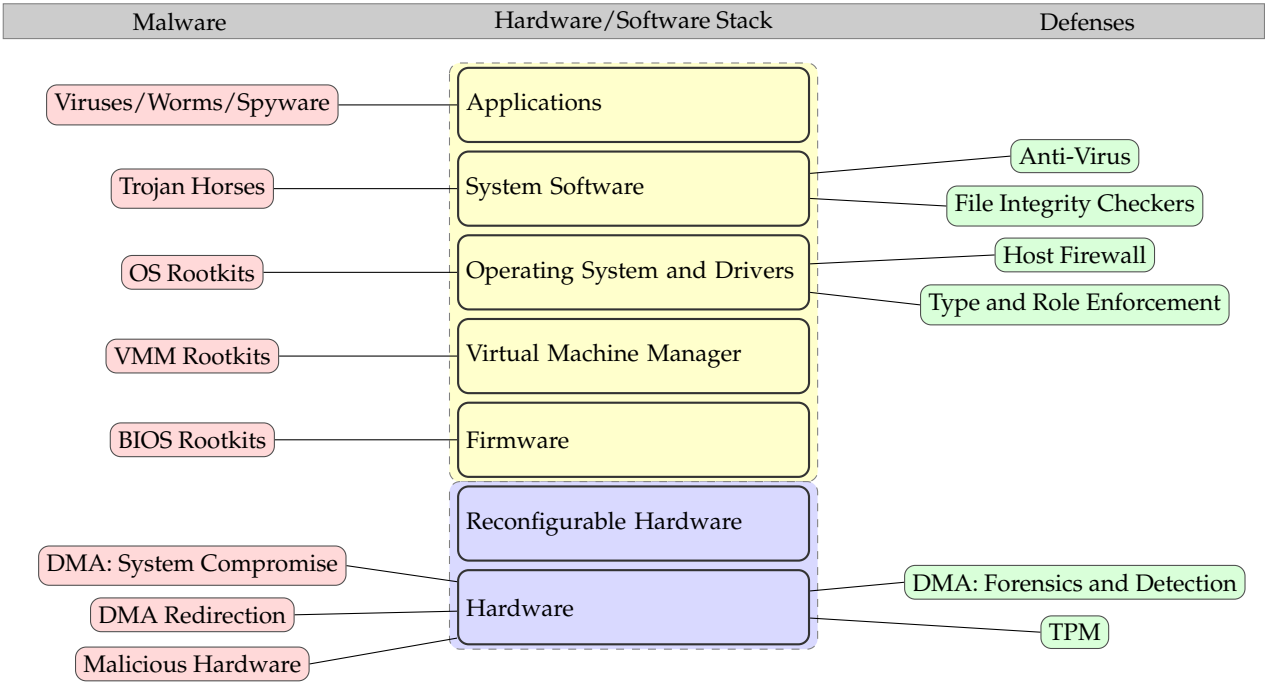
- 959 additional logic gates allow unprivileged software to access privileged memory regions
  - Placing a special sequence of bytes on the data bus disables memory protection
  - Enables privilege escalation, password stealing, etc
- 1,341 additional logic gates provide a new *shadow* execution mode
  - Reserves instruction and data cache lines for malware
  - Uses hardware debugging support to trap events from normal mode into shadow mode
  - Enables backdoors, covert monitoring, etc

Hardware supply chains create barriers to entry for casual attackers but not for advanced threats (see "Defense Science Board Task Force on High Performance Microchip Supply", 2005).

Predators Follow the Available Prey

- Protections against certain attacks are finally being built into systems
  - Non-executable memory and stack smashing detection address many buffer overflows
  - SELinux and Microsoft Vista have much stronger privilege enforcement
- New targets enter the technology space
  - Virtual machines
  - EFI
  - Reconfigurable hardware
  - Embedded hardware
- Common attackers will pursue the highest returns for the lowest investment
  - SQL injection and cross-site scripting have surpassed buffer and heap overflows
  - Heavy focus on VMM and firmware over the last few years
  - Focus is moving from OS and system software to application logic, software below the OS (VMM and firmware), and hardware
- Advanced attackers may choose additional paths
  - Can use any openly published attack vectors
  - May have unpublished or expensive attack vectors

The Current Situation



We need more defenses for application logic.
We need more defenses at the lower layers of the hardware/software stack.

Implications for Information Assurance Professionals

- Attackers are targeting low-level software and hardware
- IA professionals must start looking at low-level software and hardware to detect adversaries
- Security reviews of hardware are needed to detect flaws before the hardware is widely deployed
- Additional skills needed by IA professionals:
  - Virtual Machine Manager (VMM) programming
  - Firmware programming (BIOS, EFI, PCI option ROMs)
  - Hardware architecture and design (in FPGA and ASIC)
- Look for areas where a successful defense at one layer can be adapted to defend another layer

Questions?