Improving Scalable, Automated Baremetal Malware Analysis

Adam Allred
Paul Royal
Agenda

• Modern Malware
  – Obfuscations, Server-side Polymorphism, Collection Volume
• Malware Analysis Detection
  – Commoditization, Popularity, Transparency
  – Detecting QEMU, VMware, KVM
• Baremetal Malware Analysis
  – Hardware, Technologies
• Improving Baremetal Malware Analysis
  – Reliability Testing
• Conclusion/Future Work
an introduction to

MODERN MALWARE
Modern Malware

• At the center of current threats on the Internet
  – Commodity botnets (spamming, DDOS, etc.)
  – Theft of financial information and intellectual property
  – Espionage and sabotage

• Used by criminals
  – Well-funded, large-scale operations

• Used by nation states
  – Effect political agenda
Malware Cont’d

• There is a pronounced need to understand malware behavior
  – Threat discovery and analysis
  – Compromise detection
  – Forensics and remediation

• Malware authors make analysis challenging
  – Various motivations
Malware Obfuscations

• Pictorial Overview

- Push EBP
- MOV EBP, ESP
- SUB ESP, 8
- CALL 00401170

```
Push EBP
MOV EBP, ESP
SUB ESP, 8
CALL 00401170
```

Program A

- Machine Code

- Encrypt/Compress/Transform

Program A’

- Transformed Machine Code (Appears as Data)

- <Unpack Code>

- ... (Appears as Data)

- Result: 38/39 (97.44%)

• Project ZeroPack

- File bifrose.exe received on 02.25.2009 18:41:57 (CET)
  Current status: finished
  Result: 38/39 (97.44%)

- File bifrose_zp.exe received on 02.25.2009 18:42:11 (CET)
  Current status: finished
  Result: 0/39 (0%)
Obfuscations Cont’d

• Server-side Polymorphism
  – Automate mutations

• When done professionally: Waledac
Why Automation?

• Vastly increased volume of samples
• GTISC averages > 100,000 new samples/day
  – Higher for commercial security organizations
• Volume makes manual analysis untenable
an overview of

MALWARE ANALYSIS DETECTION
Malware Analysis Detection

• Environment-aware Malware
  – Conficker
    • Checks for relocated LDT
  – TDL4
    • Checks for device emulation via WQL
  – Bredolab
    • Checks for device emulation via DeviceIoControl()
Detection Cont’d

- Analysis tool/environment detection is a standard, inexpensive option
Detection Cont’d

• In-Guest Tools
  – No higher privilege
  – Non-privileged side effects
  – Exception handling issues

• Reduced Privilege Guests (VMware, etc)
  – Non-privileged side effects

• Emulation (QEMU, Simics)
  – No identical instruction execution semantics
Detecting QEMU

- IRETD with 0x26 prefix

```c
#include <stdlib.h>
#include <stdio.h>
#include <windows.h>

int seh_handler(struct _EXCEPTION_RECORD *exception_record,
    void *established_frame,
    struct _CONTEXT *context_record,
    void *dispatcher_context)
{
    printf("Malicious code here.\n");
    exit(0);
}

int main(int argc, char *argv[]) {
    unsigned int handler = (unsigned int) seh_handler;
    printf("Attempting QEMU detection.\n");

    __asm("movl %0, %%eax\n          pushl %%eax\n          "r" (handler): "%eax");

    __asm("pushl %fs:0\n          movl %esp, %fs:0\n          ");

    __asm(".byte 0x26, 0xcf\n          "movl %esp, %fs:0\n          ");

    return EXIT_SUCCESS;
}
```
Detecting VMware, KVM

- VMware
  - Older versions primarily use binary software translation
    - SYSRET treated as NOP when executed in ring 3
- KVM
  - Uses hardware virtualization extensions
    - Certain instructions cause VMExits
    - Older versions terminate with unhandled exit on guest execution of VMREAD
Why Transparency?

- Analysis environment detection commoditized, popular
- Detection vulnerability trend does not suggest decrease over time
- Certain types of detection vulnerabilities automatically discoverable
an introduction to

BAREMETAL MALWARE ANALYSIS
Baremetal Challenges

• Conceptual
  – Physicalizing virtual machine

• Scalability
  – Cost of hardware
  – Efficiency of processing

• Automation
  – Managing system state
  – Ensuring longevity of hardware
Baremetal Cluster Hardware (2012)

• Baremetal Controller
  – Supermicro 5016I-MTF
    • X3430 Processor, 8GB RAM, 4 x 1TB HDDs
    • Debian 6 “Squeeze”

• Baremetal Non-Virtual Machine (NVM)
  – Supermicro 5015A-PHF
    • Integrated Atom processor, 1GB RAM

• Cluster Networking
  – Cisco WS-C2960-24TC-S
    • 24 10/100Mb, 2 1Gb Ethernet ports
Baremetal Cluster Technologies (2012)

- Linux Device Mapper
  - Create Copy-on-Write (CoW) block device
- ATA-over-Ethernet (AoE)
  - Make CoW device available over network
- g Preboot eXecution Environment (gPXE)
  - Boot NVM into OS on network CoW device
- Intelligent Platform Management Interface (IPMI)
  - Manage NVM system state
NVMTrace

• Software controller for automated baremetal malware analysis
  – Executes each sample in its own sterile, isolated non-virtual machine

• Provides access to NVM disk contents and network traffic
  – Use with your favorite network traffic and disk forensic tools
Implementation Corner Cases

• System Clock
  – Sample can modify system time
  – Modify gPXE to set sane value, sync immediately prior to sample execution

• NVM PSU Lifetime
  – Turning NVM on, off hundreds of times each day can impact PSU longevity
  – Use resets instead
GTISC NVMTrace Deployment
GTISC Deployment Cont’d
evaluation and enhancement of

BAREMETAL MALWARE ANALYSIS
NVMTrace Reliability Testing

• Anecdotal observation indicated potential issue in sample processing
• Sought to verify by using a well-known set of samples that make use of the DNS
  – Count samples that make DNS queries
  – Make a change, reprocess set
• Accept changes that increase percentage the number of samples that query DNS
Observed Symptom

• Windows boot hang
  – Nondeterministic across nodes
• Manually reproducible
• ATA over Ethernet suspected
  – Problem occurred right after activation
ATA over Ethernet

• Simple (12 page specification)
  – TFTP-like connection
• Unreliable
  – No packet retransmission, checksumming
• Network analysis confirmed AoE traffic ceases at hang
  – Packet loss or corruption impedes node execution
iSCSI

• Proposed as replacement for AoE
  – Provides reliable transport via TCP

• Candidate implementation must handle atypical use
  – Constant iSCSI LUN add/remove
iSCSI Reliability Testing

• Evaluated several iSCSI implementation candidates that did not work
  – SCST, STGT, Open-iSCSI
• Eventually tried LIO, which did work
• Also tested AoE under Debian 7
  – No change in reliability
Results

• > 99% of samples in well-known set successfully processed using LIO
  – Verified via multiple rounds of testing
• Additional testing with separate, ~200,000 sample dataset
  – Represented 24 hours of real-world collection
  – Virtualization-based processing results used as reference
  – Results reaffirm > 99% success rate
• Subsequent stable production use for months
Conclusion

• Analysis environment detection commoditized, increasingly popular
  – Virtualization still a valuable analysis tool, but can be supplemented
• Advances in hardware make scalable baremetal malware analysis possible
• Baremetal analysis systems must be carefully engineered for reliability
Future Work

- Increase cluster density via Supermicro MicroClouds
  - Yields three-fold increase in processing density
- Chainload into i/g PXE via PXE
  - Remove need for read-only USB devices
- Disk Forensics
  - Examine controller-NVM iSCSI network traffic
  - Record disk-level events as they occur
Acknowledgements

• Artem Dinaburg
  – Environment detection

• Robert Edmonds
  – System architecture

• David Dagon
  – System concept
Questions?

NVMTrace Source Code, Build Instructions
http://code.google.com/p/nvmtrace