Solutions for Memory Forensics & Automated Malware Reversing
HBGary Background

• Founded in 2003
  • Government R&D

• Solutions:
  • Enterprise Malicious Code Detection
  • Live Windows Memory Forensics & Incident Response
  • Malicious Code Detection
  • Automated Reverse Engineering

R&D Funding

Air Force Research Labs
• Next Generation Software Reverse Engineering Tools
• Kernel Virtual Machine Host Analyzer
• Virtual Machine Debugger

Dept Homeland Security (HSARPA)
• Botnet Detection and Mitigation
• H/W Assisted System Security Monitor
• Subcontractor to AFCO Systems Development
3 Core Technologies

- Physical Memory Forensics
- Code Reverse Engineering
- Digital DNA (Behavioral Analysis)
Technology and Workflow

- Offline Physical Memory Analysis
  - Unprecedented Visibility
  - "Automated Crash Dump Analysis"
  - No code executing to "actively" fool our analysis

- Automated Malware Analysis
  - Rapidly identify the malicious code capabilities
  - Generate Report

- Enterprise Policy Change
  - URL's and IP address blocking
  - IDS/IPS – Detection and Blocking Rules
  - Identify Scope of Breach
  - Develop and Implement Optimal corrective action plan
The Core Technology

- Offline Physical Memory Analysis
- Rootkit Detection
- Automated Malware Disassembly
- Digital DNA
- Alerting & Reporting

This is The Advantage!
Rebuilds underlying undocumented data structures
Rebuilds running state of machine “exposes all objects”
Malware cannot hide itself actively
The Core Technology

- Offline Physical Memory Analysis
- Rootkit Detection
- Automated Malware Disassembly
- Digital DNA
- Alerting & Reporting

These tricks expose themselves by interacting with OS
- Direct Kernel Object Manipulation Detection
- Hook Detection IDT/SSDT/Driver Chains
- Crossview Based Analysis
The Core Technology

- Offline Physical Memory Analysis
- Rootkit Detection
- Automated Malware Disassembly
- Digital DNA
- Alerting & Reporting

- Suspicious Code is extracted from RAM
- Code is Disassembled, broken apart, and analyzed
- Integration with Flypaper & Flypaper Pro
- Code Control Flow Graphing
The Core Technology

- Offline Physical Memory Analysis: Identifies executable code behaviors
- Rootkit Detection: DDNA created for all executable code
- Automated Malware Disassembly: A Threat Score is provided for all code
- Digital DNA: White & Black List Code / Behaviors
- Alerting & Reporting
MD5 Doesn’t Work in Memory… so we created Digital DNA.
Why MD5’s Don’t Work in Memory

• A file executing in RAM is represented in a new way that cannot be easily be back referenced to a file checksum

• Digital DNA™ does not change, even if the underlying file does
  – Digital DNA is calculated from what the software DOES (it’s behavior), not how it was compiled or packaged
State of malicious code

Rootkit

Executable code is only visible in RAM and Pagefile

No representation of what is being stolen

Encrypted Covert Channel

Drive by Download

Filesystem

Runtime Memory
In memory, traditional checksums don’t work.

MD5 Checksum is not consistent

Digital DNA remains consistent

100% dynamic
Copied in full
Copied in part

DISK FILE

IN MEMORY IMAGE

MD5 Checksum reliable
MD5 Checksum is white listed

Process is trusted

White-listing on disk doesn’t prevent malware from being in memory

Whitelisted code does not mean secure code

Public Attack-kits have used memory-only injection for over 5 years
IN MEMORY IMAGE

Digital DNA remains consistent

OS Loader

Starting Malware

Packed Malware

Digital DNA defeats packers

Packer #1

Packer #2

Decrypted Original