Automating Linux Memory Capture

Hal Pomeranz

Deer Run Associates
ENGAGE DEMO!
Why Memory Forensics?

• Detect malware and rootkits
• Defeat encryption
• Speed up analysis
Memory Analysis Challenges

- Need to load a driver to capture RAM
- Need to locate kernel data structures

- *Incredibly OS version dependent*
- *Small changes break analysis tools*
Why Is Linux Hard?

Too Many Kernels!
Linux Memory Acquisition
(the short summary)

- Obtain driver source code
- Build driver for target system (Where?)
- Obtain administrative access on target system
- Determine RAM capture destination:
  - Portable device: attach and mount  
  - Network: configure remote destination
- Load driver
- Initiate capture
Linux Analysis Profile Creation

Dependencies: Volatility™, dwarfdump, appropriate kernel build environment...

- Dump locations of kernel data structures
- Obtain symbol table for target kernel
- Create profile archive (ZIP file)
- Determine appropriate profile name/location
We Need Leverage!

"Smart people could handle these steps"

• Smart people should be doing analysis
• Smart people may not be available
• Contains 3rd-party dependencies:
  – LiME kernel module source
  – dwarfdump
  – Volatility™

• Hal's "img" script:
  – Builds LiME
  – Captures RAM to USB device
  – Creates Volatility™ profile
Issues of Purity

• Attaching writable media to target
• Development environment required on target
• Executing programs from target OS
• Creates memory artifacts of its own
BACK TO DEMO!
Last Chance for Questions!

The tool – https://github.com/halpomeranz/lmg

Other stuff –
http://deer-run.com/~hal/
http://digital-forensics.sans.org/blog/author/halpomeranz/

Hal Pomeranz  Deer Run Associates
@hal_pomeranz  hal@deer-run.com