DFRWS Forensics Challenge 2016

Robert Beverly, Brian Greunke,
Michael McCarrin[#]
Joe Sylve[*]
Vassil Roussev[+]

[#]Naval Postgraduate School
[*]BlackBag Technologies
[+]University of New Orleans

DFRWS
DIGITAL FORENSIC RESEARCH WORKSHOP

Forensics Challenge 2016

- Background
- Scenario
- Results

DFRWS Forensics Challenge Goals

1. Advance research in new and emerging areas of digital forensics
2. Spur development of new tools and techniques

Rich History of Offering Timely Forensics Challenges to the Community…

Year	Challenge
2015	GPU Malware
2014	Mobile Malware
2013	Block Classifier
2012	Block Classifier
2011	Android Forensics
2010	Flash Memory Forensics
2009	Playstation Forensics
2008	Linux Memory Analysis

Rich History of Offering Timely Forensics Challenges to the Community…

Year	Challenge
2016	Software Defined Networks (SDN)
2015	GPU Malware
2014	Mobile Malware
2013	Block Classifier
2012	Block Classifier
2011	Android Forensics
2010	Flash Memory Forensics
2009	Playstation Forensics
2008	Linux Memory Analysis

Software Defined Networks

- New Model for Building/Operating Networks:
  - Move away from proprietary network OS
  - Open/programmable network switches
  - Standards-based protocol (OpenFlow)
  - Commodity hardware
  - Centralized control
- Promise:
  - Lower-cost, multi-vendor
  - Correctness
  - Enable innovation within network
  - Enable virtualization

SDN Abstraction

- Switches maintain a “flow table:”
  - Packet matching rules and actions
  - Hard and soft state
  - If no match, packet is sent to a controller
- E.g.
  - `in_port=2,nw_src=42.59.142.200/30 actions=mod_dl_src:41:31:3a:38:42:3a idle_timeout=45, out_port=7`
- Controllers:
  - OpenFlow-speaking software on a PC
  - Proactively or reactively install flow rules
  - Include sophisticated logic

State of SDN

- Implementations:
  - Hardware from major vendors and startups
  - Software switches (e.g., Open vSwitch in Linux Kernel)
  - Open software controllers (e.g., Ryu, Pox, Floodlight, etc)
- Deployments:
  - Google B4, Amazon, enterprises, etc.
  - *More virtual switch ports in existence today than physical!*
- But, security:
  - Only nascent research
  - *No work on SDN forensics*

Forensics Challenge 2016

- Background
- Scenario
- Results

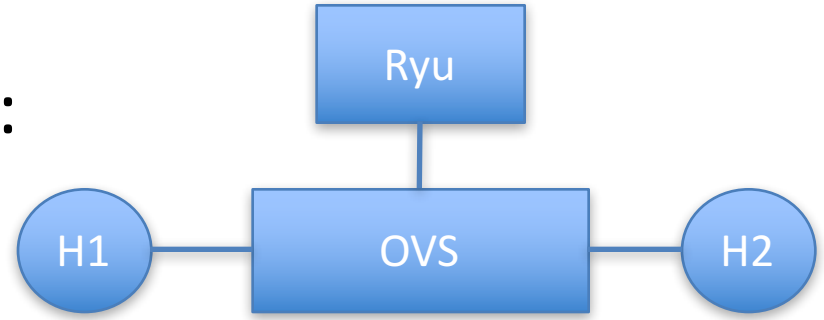
SDN Challenge

- Participants given:
  - Switch memory image
  - pcap of net traffic between controller and switch
  - No other knowledge or clues of scenario setup

- Forensic questions:
  - What type of SDN switch and controller?
  - What hosts were connected to which ports?
  - What traffic did hosts send?
  - What flow rules were installed on switch?
  - What actions did switch take, and when?

Scenario

- 4 Physically distinct devices:
  - Ryu OpenFlow controller
  - OVS Linux SDN switch
  - Two hosts



- OpenFlow TLS ECDHE between controller and switch, with cert identifying info removed
- LiME raw memory image of Linux OVS after reboot*
  - Pre-installed and dynamic flow table rules
  - Expired and non-expired flow table rules

Levels of Complexity

- What type of SDN switch and controller?
  - LiME memory dump, Linux/OVS signatures
  - Controller capabilities negotiation
- What hosts were connected to which ports?
  - Flow rules (w/ MAC and IPs), including residual, present in memory dumps
  - But where in memory…
- What actions did switch/controller take, and when?
  - Some activity revealed in encrypted traffic

Forensics Challenge 2016

- Background
- Scenario
- Results

Participants (thanks!)

- Four submissions:
  - Korea University
  - Booze Allen Hamilton (BAH)
  - University of Newhaven
  - Salford University

Booz | Allen | Hamilton

University of Salford MANCHESTER

고려대학교 KOREA UNIVERSITY

Challenge Approaches

- Misunderstand challenge:
  - e.g., IP addresses of controller and switch rather than hosts and flow rules
- Analyze encrypted pcap:
  - Difficult and limited
- Carve OVS data structures/logs from memory image
- Obtain ECDHE private key, pre-master key from memory image, decrypt southbound pcap

2016 Winners

Booz | Allen | Hamilton

Congrats!!

BAH Submission (or, what makes a winning submission)

- Well documented:
  - Approach
  - Findings
  - How to reproduce
- Correctly answered all challenge questions
- Created volatility plugins
- Worked toward automating system

BAH Approach

1. Determine format/type of memory dump (using strings)
   - Linux machine, LiME dump, but raw format
2. Recreate physical to virtual address mapping
   - From BIOS artifacts
3. Create new memory image
4. Run volatility with correct Linux profile

BAH Approach (con't)

- Recover private key from memory
- Analyze TLS handshake
- Yara on memory image to obtain pre-master key
- Identify TLS handshake messages with random time and bytes, obtain master key
- Decrypt pcap, recover OpenFlow messages
- (Also, obtain OVS log messages)

Wrapping Up

- Full solution to be posted on dfrws.org after this session
- Thanks to all the participants!

Challenge 2017

- Next year's challenge TBD:
  - Continue SDN?
  - Internet of things?
  - UEFI malware?
  - Windows 10?
  - Drones?
  - Cloud?
  - Other?

Please send feedback/flames:
challenge@dfrws.org