Towards Automated Dynamic Analysis for Linux-based Embedded Firmware

Dominic Chen[1], Manuel Egele[2], Maverick Woo[1], David Brumley[1]

[1]Carnegie Mellon University, [2]Boston University

{ddchen, pooh, dbrumley}@cmu.edu, megele@bu.edu

LG

wemo

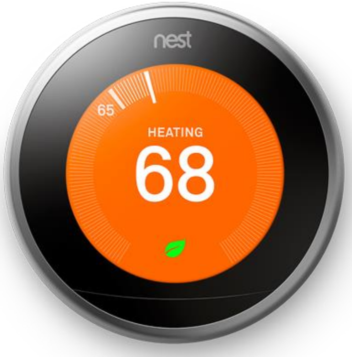
iRobot

SURFboard

THE INTERNET of THINGS

FOSCAM

LINKSYS


nest
65
HEATING
68

86

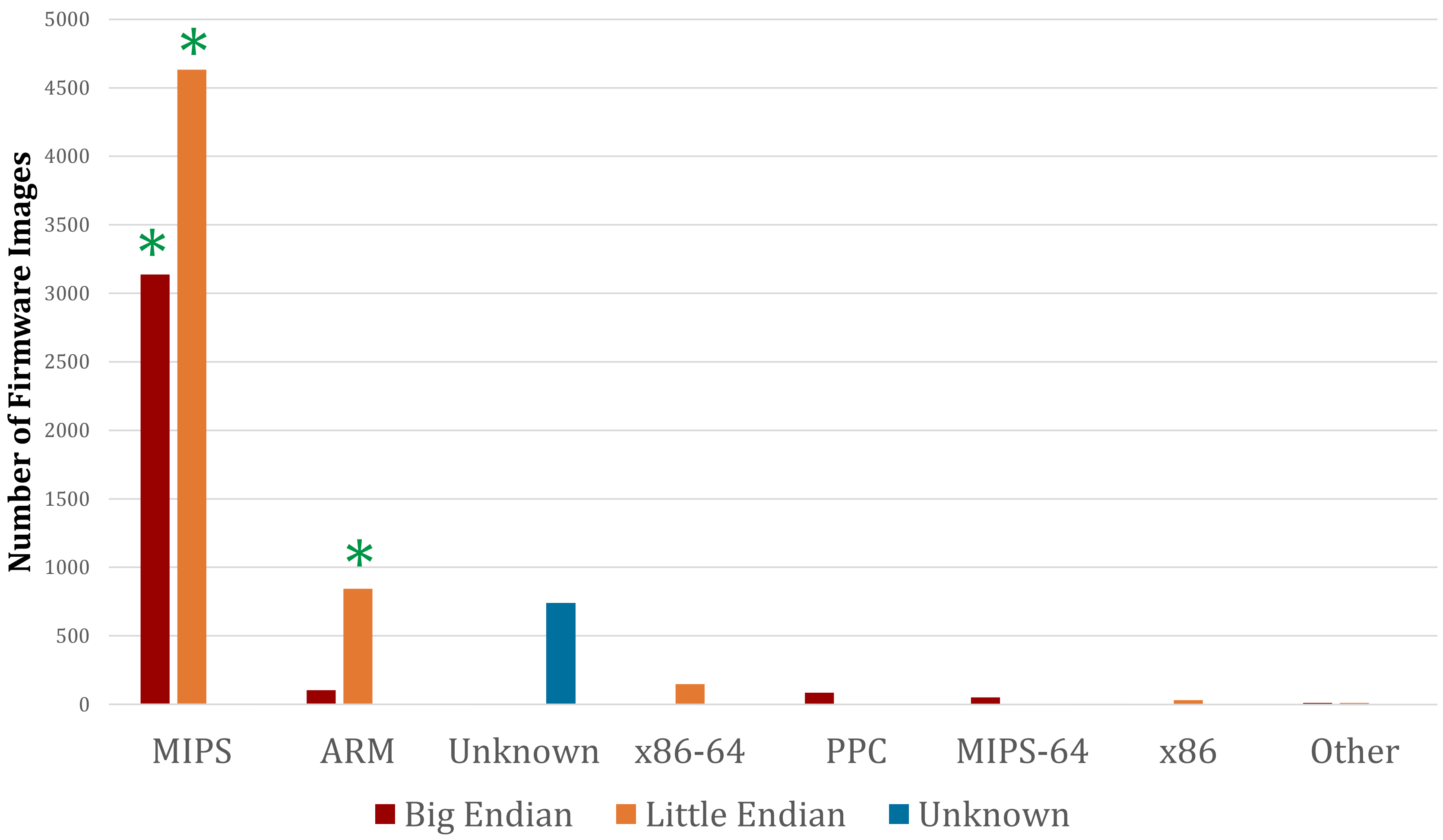
FIRMADYNE

- First system for full-system emulation of embedded Linux-based firmware
- Provides large-scale automated dynamic analysis
  - Built-in vulnerability detection
  - Tested on 9.5k extracted firmware images
- Objective: Continuous integration for firmware

Background

- Embedded devices are important
  - Low visibility by end-users
  - Critical network infrastructure
  - Software rarely upgraded
- Difficult to analyze
  - RISC-based architectures: MIPS, ARM, etc.
  - No direct interface into device firmware
  - Fixed hardware peripherals; no ‘Plug and Play’
  - Significant variety; hard to scale

Firmware Architectures

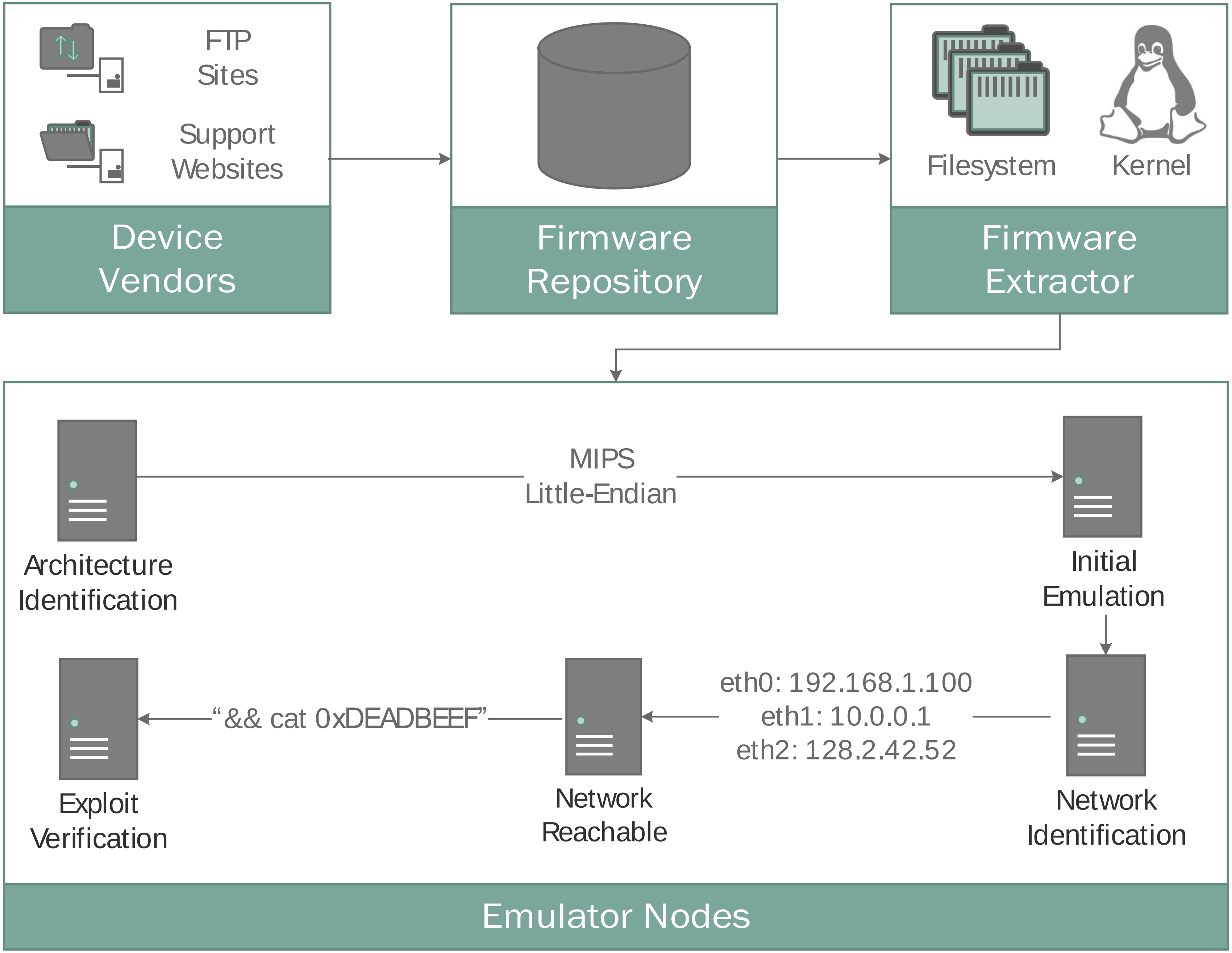


Related Work

- Zaddach et al., "*Avatar: A framework to support dynamic security analysis of embedded systems' firmwares*", NDSS 2014
  - Software emulation with partial offload to hardware
  - Doesn't scale: requires hardware and connection to debug port
- Costin et al., "*A large-scale analysis of the security of embedded firmwares*", USENIX 2014
  - Static extraction and analysis of firmware
  - Relatively cursory analysis and can't verify results; classic trade-offs of false positives vs. false negatives

Dynamic Approaches

- Application-level
  - Extract webpages and perform analysis
  - Custom interpreter modifications
- Process-level
  - Emulate original applications in user-mode
  - Different hardware and execution environment
- **System-level**
  - Boots entire filesystem with modified kernel
  - Supports all applications using original environment



FTP Sites

Support Websites

Device Vendors

Firmware Repository

Filesystem

Kernel

Firmware Extractor

Architecture Identification

MIPS Little-Endian

Initial Emulation

Network Identification

eth0: 192.168.1.100
eth1: 10.0.0.1
eth2: 128.2.42.52

Network Reachable

“&& cat 0xDEADBEEF”

Exploit Verification

Emulator Nodes

Filesystem Recovery

- Firmware format is not standardized
  - Can be compressed, include photos, etc.
- **Solution**: Develop custom extractor for filesystems
  - Searches for UNIX-like filesystems
  - Includes heuristics to avoid recursive extraction
- Improved existing unpacking tools
  - jefferson: User-mode extractor for JFFS2
  - sasquatch: Heuristic-based extractor for SquashFS

Device Configuration

- Firmware requires NVRAM peripheral to boot
  - Used as volatile configuration store
- **Solution**: Emulate NVRAM peripheral with userspace library
  - Compatible with different C runtime libraries
  - Self-initializes with default NVRAM values used during factory reset

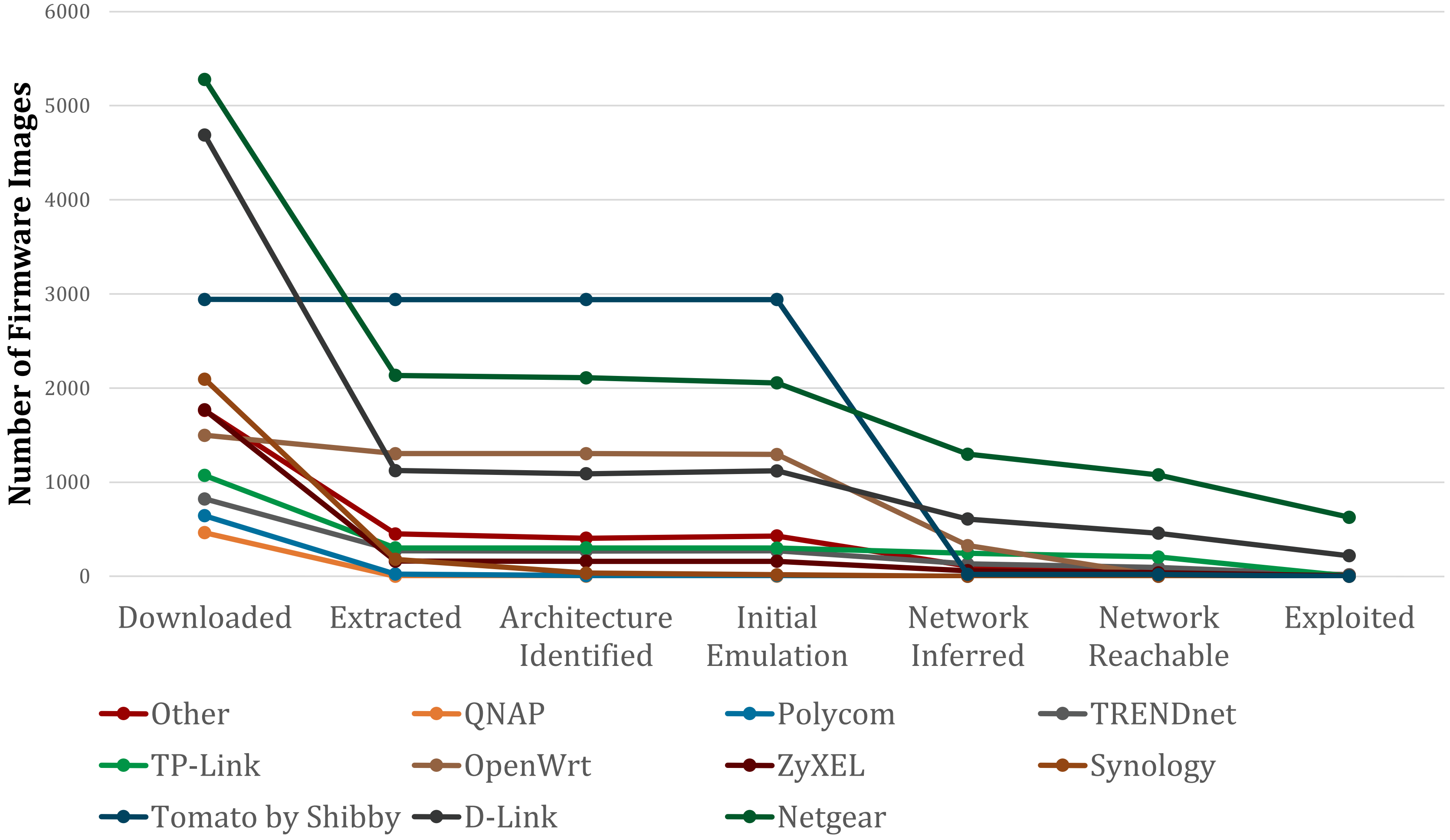
Network Inference

- Devices expect different network configuration
  - eth0 vs. lan0, wlan0, wan0, vs. ath0, br0, etc.
- **Solution**: Use custom kernel with software instrumentation to infer networking
  - Parse kernel log to infer expected configuration
  - Track IP addresses, bridges, and VLANs
  - Restart with new configuration

Automated Analyses

- Accessible Webpages
  - Checks for unauthenticated webpages
  - Command injection/information disclosure
- SNMP Information
  - Dumps public SNMP data
  - Information disclosure
- Vulnerability Detection
  - Checks for presence of vulnerabilities

Firmware Analysis Progress by Vendor



Vulnerability Analysis

- Discovered 14 previously-unknown vulnerabilities
  - New vulnerabilities can be automatically tested across entire dataset
  - Selected 60 applicable vulnerabilities from Metasploit
- Of 1,971 firmware images that were network reachable, 43%* (846) were vulnerable to at least one exploit
  - Estimated to affect 89+ different products

* Corrected

Unknown Vulnerabilities

- Discovered 14 unknown vulnerabilities that affect 69 firmware images across 12+ products using our analyses
  - Command Injection (Netgear)
  - Buffer Overflow (D-Link)
  - Information Disclosure (D-Link & Netgear)
- Responsible disclosure to vendors and CERT
  - VU#548680: Affected D-Link devices
  - VU#615808: Affected Netgear devices
    - Fix is expected by end of February/mid-March

Netgear Command Injection (CVE-2016-1555)

- Unauthenticated webpages with debug functionality were accidentally included
  - Used to write manufacturing data, e.g. MAC addresses, firmware region, and serial number
  - Can detect with our instrumentation
- Form input is passed directly as command-line argument to shell
  - Affects 65 firmware images across 7+ products

D-Link Buffer Overflow (CVE-2016-1558)

- Web server sets *dlink_uid* cookie to track sessions for authenticated users
  - Value is passed to strlen() then memcpy()
- Setting the cookie to a long string crashes the web server at e.g. 0x41414141
  - Affects 13 firmware images across 5+ products

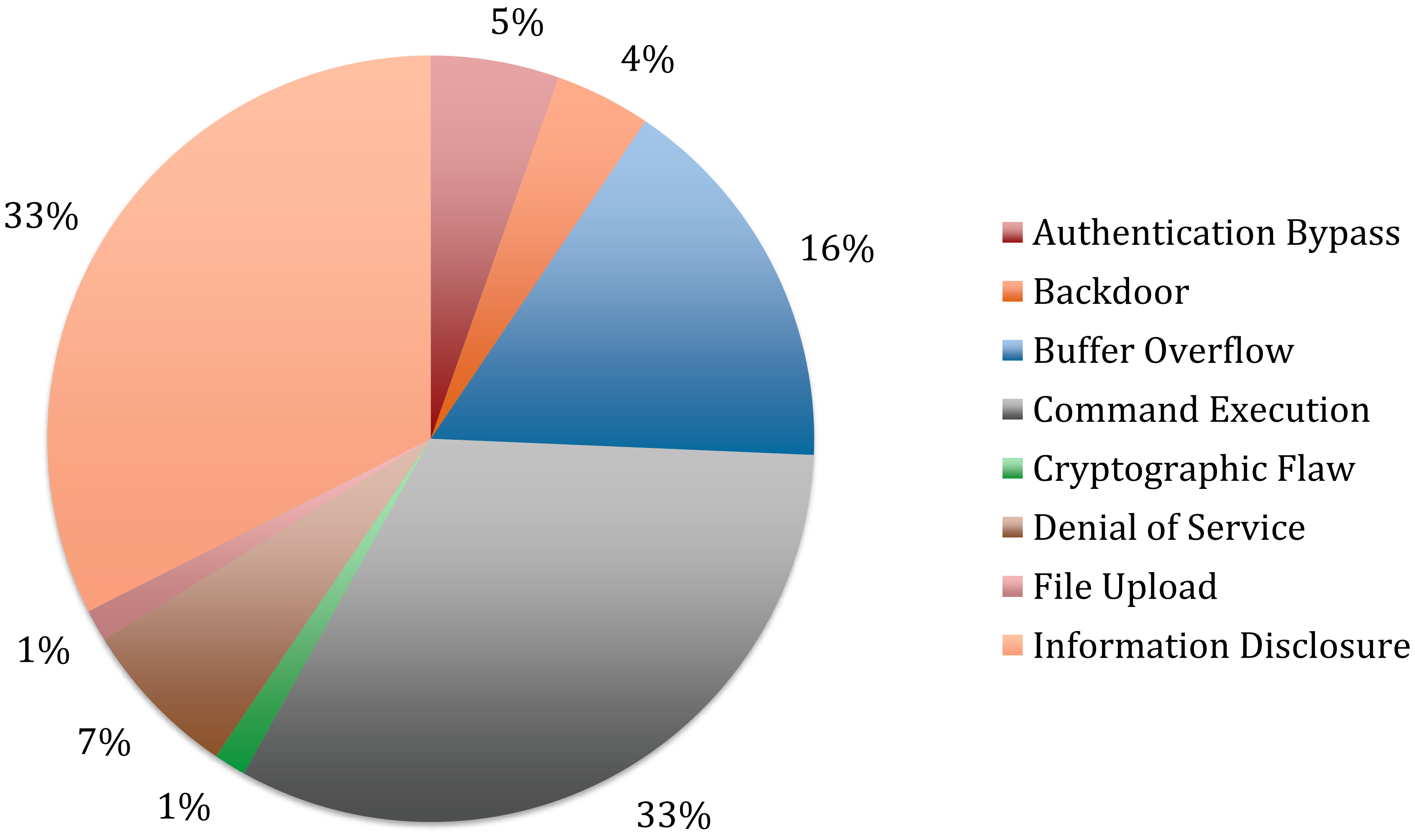
D-Link & Netgear Information Disclosure

- Unauthenticated services provide sensitive information
  - Web pages (CVE-2016-1556)
  - SNMP queries (CVE-2016-1557, CVE-2016-1559)
- Insecure default configuration
  - Affects 54 firmware images across 10+ products

Code Reuse

- Sercomm Backdoor (CVE-2014-0659)
  - Unauthenticated remote attackers can dump configuration
  - Affects 282 firmware images across 16+ products from our dataset
  - Our results show On Networks and TRENDnet are also affected
- MiniUPnPd Denial of Service (CVE-2013-0229)
  - Parsing flaws in open-source internet-facing UPnP daemon
  - Affects 169 firmware images across 14+ products from our dataset
- OpenSSL ChangeCipherSpec (CVE-2014-0224)
  - TLS implementation allows attacker to downgrade cipher
  - Affects 169 firmware images across 27+ products from our dataset

Classification of Tested Vulnerabilities



Conclusion

- FIRMADYNE allows full-system emulation and dynamic analysis of Linux-based firmware
  - Infers network configuration of firmware
  - Emulates hardware peripherals, e.g. NVRAM
  - Automatically checks for vulnerabilities across dataset
- 43% of all network reachable firmware images are vulnerable to at least one exploit
  - Future work in investigating code sharing among OEM's
- Open-source and available today
  - https://github.com/firmadyne
  - Patches welcome!

Questions

- Dominic Chen (ddchen@cmu.edu)