FirmAE: Towards Large-Scale Emulation of IoT Firmware for Dynamic Analysis

Mingeun Kim[1], Dongkwan Kim[2], Eunsoo Kim[2], Suryeon Kim[3], Yeongjin Jang[4], and Yongdae Kim[2]

[1]The affiliated institute of ETRI
[2]KAIST
[3]Ministry of National Defense
[4]Oregon State University

IoT Devices are in danger

- 34.2 billion embedded devices will be in use in 2025*
  - Wireless routers, IP cameras, …

- IoT Devices are an alluring target
  - Satori botnet using 0-days (Dec. 2017)
  - Crypto mining botnet (May. 2018)
  - ECHOBOT, a variant of Mirai (Dec. 2019)
  - New Mirai variant targeting Comtrend routers (July 2020)

- Many IoT devices are exposed to the Internet, especially their web interfaces
  - Shodan, ZoomEye
  - Over 30 exploits used in ECHOBOT target web services
  - Web service RCE (CVE-2020-10173) used for Mirai variants

*https://iot-analytics.com/state-of-the-iot-update-q1-q2-2018-number-of-iot-devices-now-7b/

Analyzing device firmware

- Statically analyze device firmware ➔ Many false positives
  - Crack default passwords or find backdoor strings: Costin et al. (SEC '14), …
  - Symbolic execution to find vulnerabilities: FIE (SEC'13), Firmalice (NDSS'15), …
- User-level emulation
  - Emulate only the target program, not the entire environment
  - Utilize "chroot" on the firmware filesystem: Costin et al. (AsiaCCS '16)

  ➔ Cannot reflect system-wide behavior (e.g., device initialization)
- System-level emulation
  - Emulating the entire environment, including the kernel
    - Firmadyne (NDSS'16), FirmPin (BLACKHAT US'18), Firm-AFL (SEC'19), …

  ➔ Many approaches take this and analyze vulnerabilities
- Modeling accurate peripherals
  - MMIO, GPIO, DMA: Pretender (RAID'19), HALucinator (SEC'20), P2IM (SEC'20), …

  ➔ Promising, but immature to support large-scale analysis

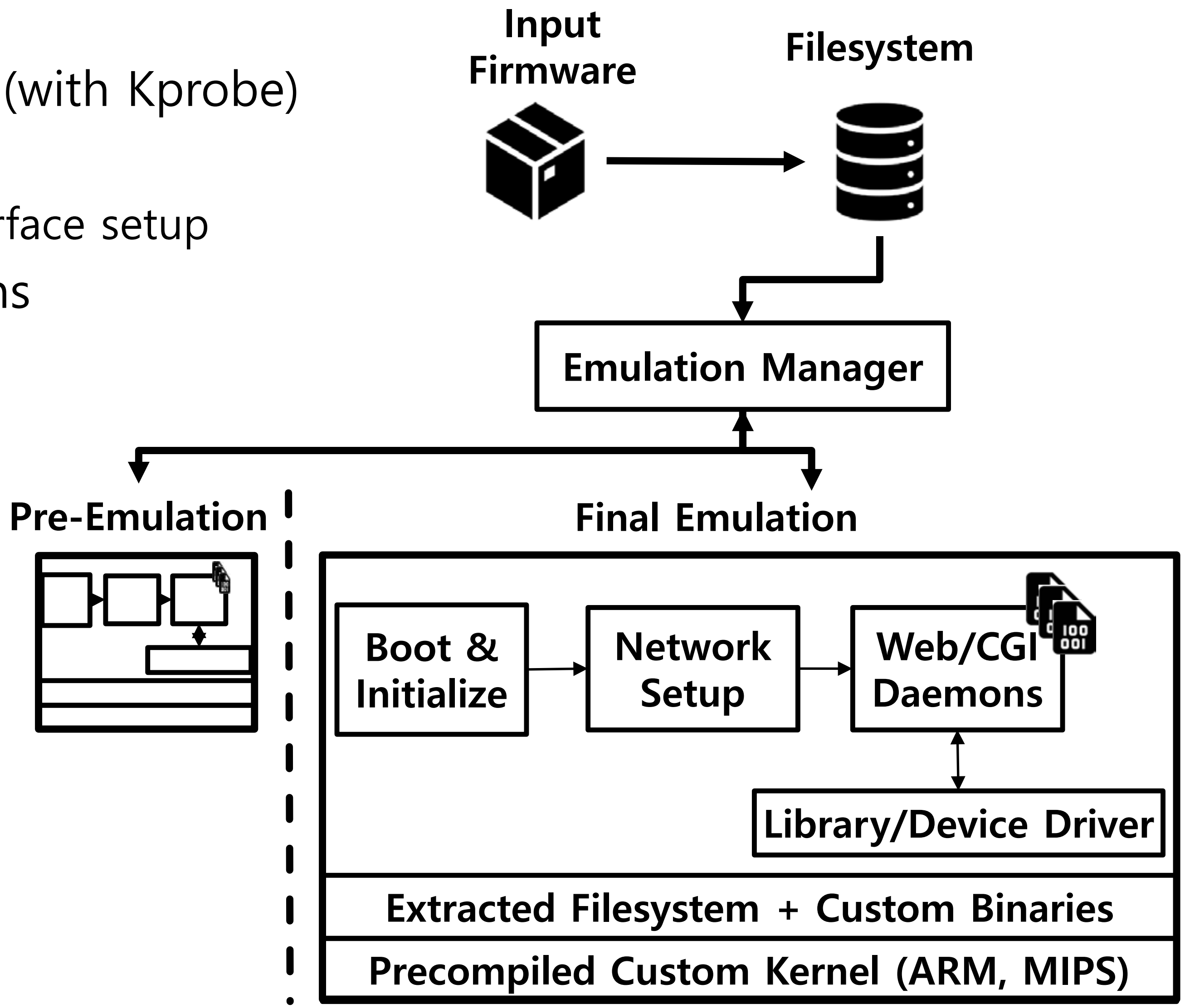
Analyzing device firmware

- ❖ Statically analyze device firmware ➔ Many false positives
  - – Crack default passwords or find backdoor strings: Costin et al. (SEC '14), …
  - – Symbolic execution to find vulnerabilities: FIE (SEC'13), Firmalice (NDSS'15), …
- ❖ User-level emulation
  - – Emulate only the target program, not the enti
  - – Utilize "chroot" on the firmware filesystem: Co
  - ➔ Cannot reflect system-wide behavior (e.g., dev
- ❖ System-level emulation
  - – Emulating the entire environment, including the kernel
    - ▪ Firmadyne (NDSS'16), FirmPin (BLACKHAT US'18), Firm-AFL (SEC'19), …
  - ➔ Many approaches take this and analyze vulnerabilities
- ❖ Modeling accurate peripherals
  - – MMIO, GPIO, DMA: Pretender (RAID'19), HALucinator (SEC'20), P2IM (SEC'20), …
  - ➔ Promising, but immature to support large-scale analysis

Is this sufficient to check vulnerabilities on a large-scale?

Firmadyne: state-of-the-art firmware emulator

- ❖ QEMU for a virtual environment
- ❖ Prebuilt kernel for hooking system calls (with Kprobe)
- ❖ Emulating target firmware twice
  - – Collect system call logs for network interface setup
- ❖ NVRAM* library to wrap related functions

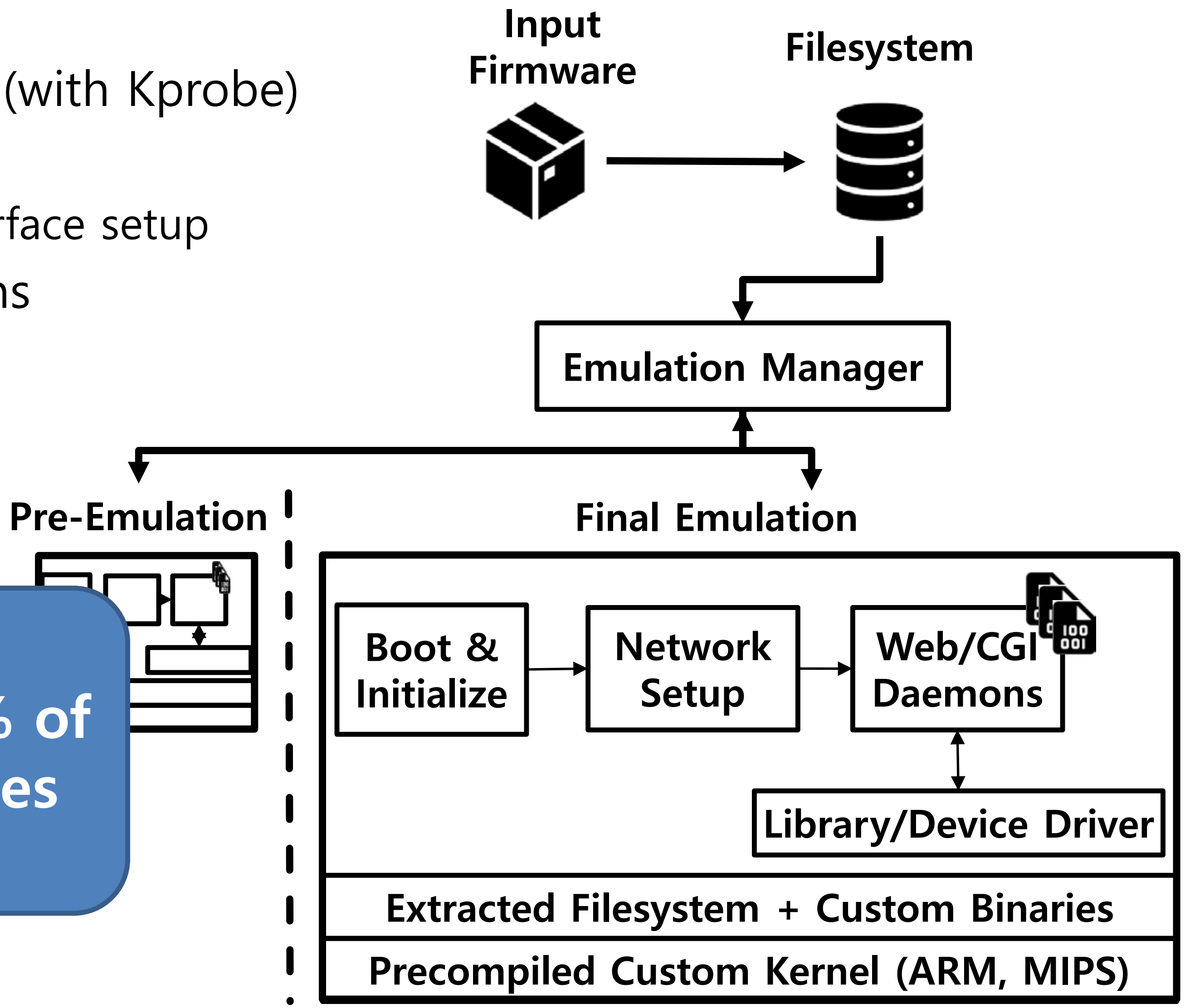


*NVRAM (Non-Volatile RAM) stores configuration key-value pairs

Firmadyne: state-of-the-art firmware emulator

- QEMU for a virtual environment
- Prebuilt kernel for hooking system calls (with Kprobe)
- Emulating target firmware twice
  - Collect system call logs for network interface setup
- NVRAM* library to wrap related functions

Firmadyne can emulate only 16% of firmware images for web services



*NVRAM (Non-Volatile RAM) stores configuration key-value pairs

Practical large-scale emulation for analyzing IoT devices
➔ Web services, typical attack targets

Randomness of embedded device implementation

➔ Difficulty of catching precise failure causes

➔ No need to be accurate for dynamic analysis

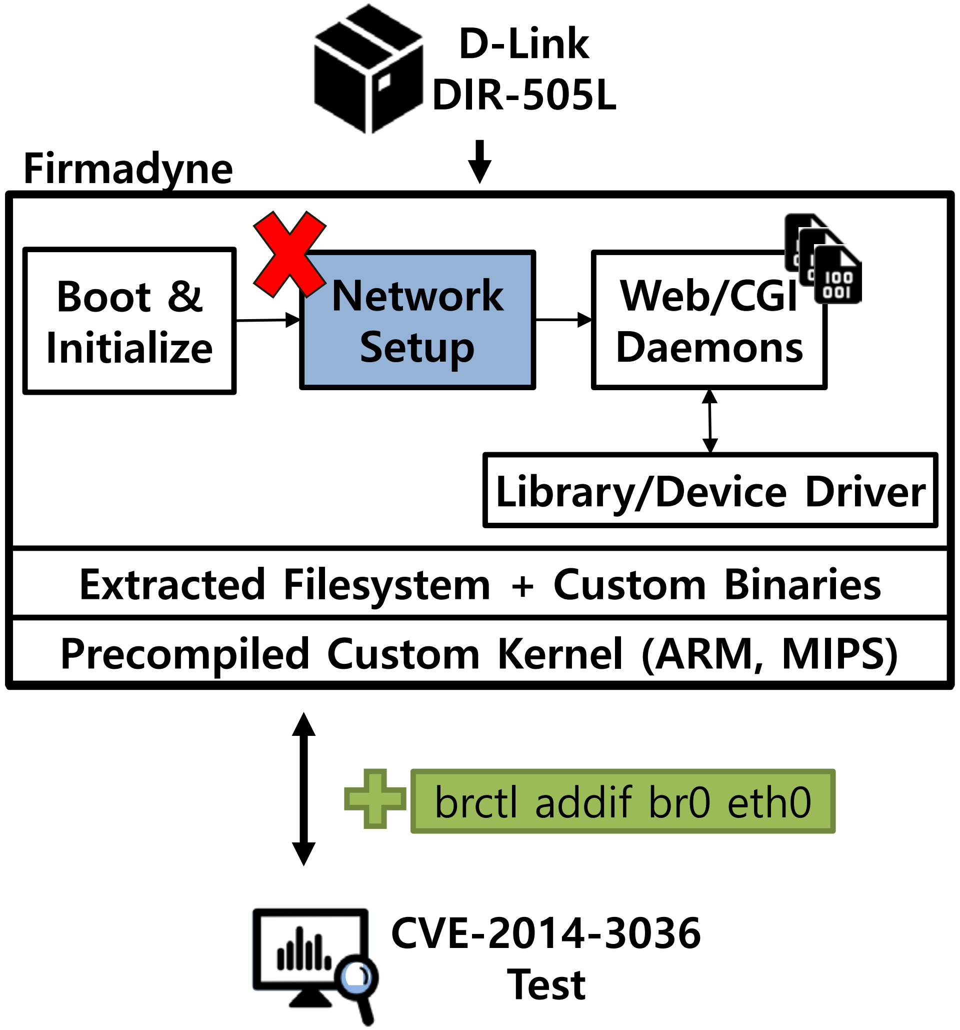
➔ Subtle efforts can address many failure cases

➔ Once implemented, such experience can build up

➔ Successful emulation of 892 firmware images!

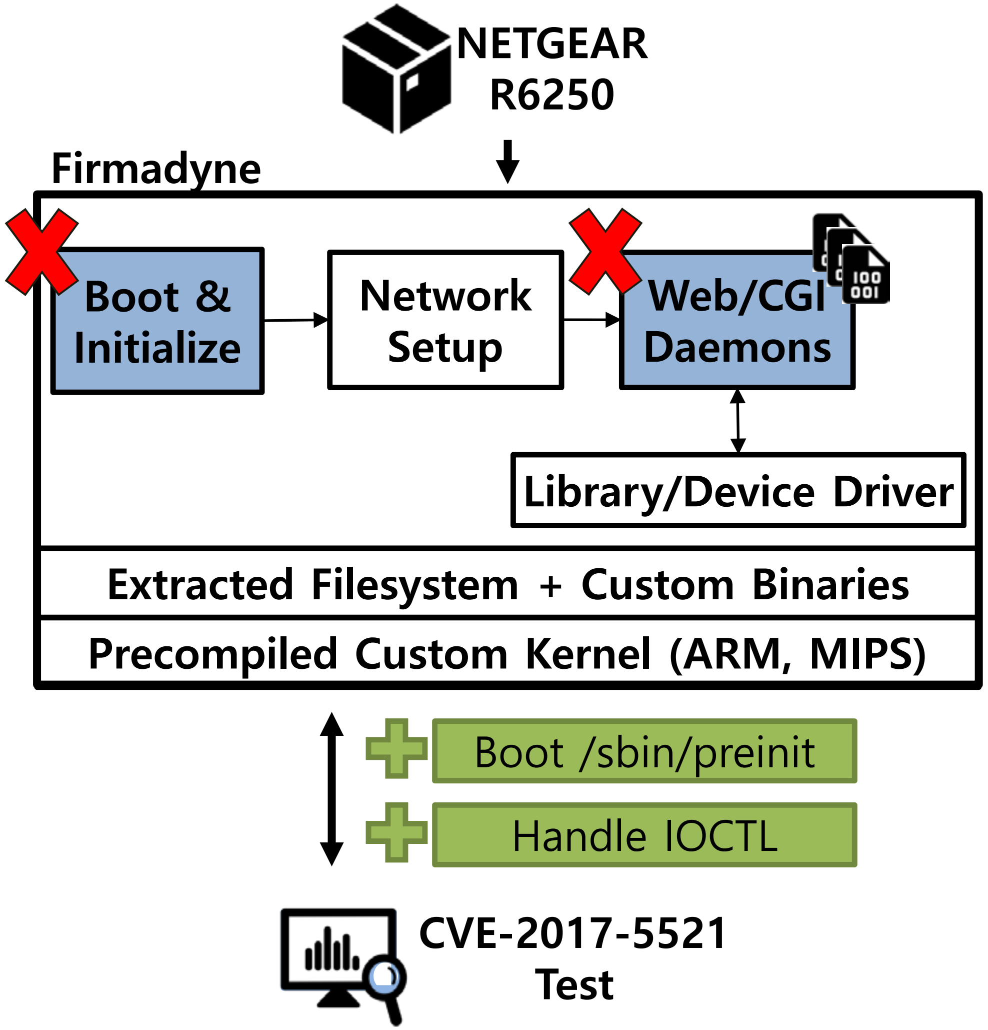
Motivating example 1: CVE-2014-3936

- Target
  - D-Link DIR-505L
- Symptom
  - Fails to configure network connection
    - Missing bridge interface to communicate with the host
- Possible causes
  - Access to unsupported peripherals
  - Missing NVRAM configuration value
- How to address
  - Run a single command that links the bridge interface



Motivating example 2: CVE-2017-5521

- Target
  - NETGEAR R6250
- Symptom
  - Fails to boot
    - Diverse initializing program paths
  - Fails to run the web service
    - Missing IOCTL functions
- Possible causes
  - Incorrect initializing program path
  - Missing kernel module
- How to address
  - Change the initializing program path to “/sbin/preinit”
  - Add IOCTL wrappers



Our approach

- Key observation
  - Emulating high-level behaviors can be sufficient to conduct dynamic analysis
  - Relatively easy and does not need to address the exact causes of emulation failures

- Arbitrated emulation
  - Ensures high-level conditions to run target programs by injecting interventions*
  - Focuses on emulating target program to conduct dynamic analysis

- Goal
  - Emulating **web services** in firmware for **dynamic analysis** (i.e., bug hunting) in a large scale
  - Targeting wireless routers and IP-cameras
    - Popular attack targets and still have many vulnerabilities

- High-level conditions to analyze web services
  - A device should be booted without kernel panic
  - Its network should be reachable from the host
  - Its internal web services should be available

Check violation cases

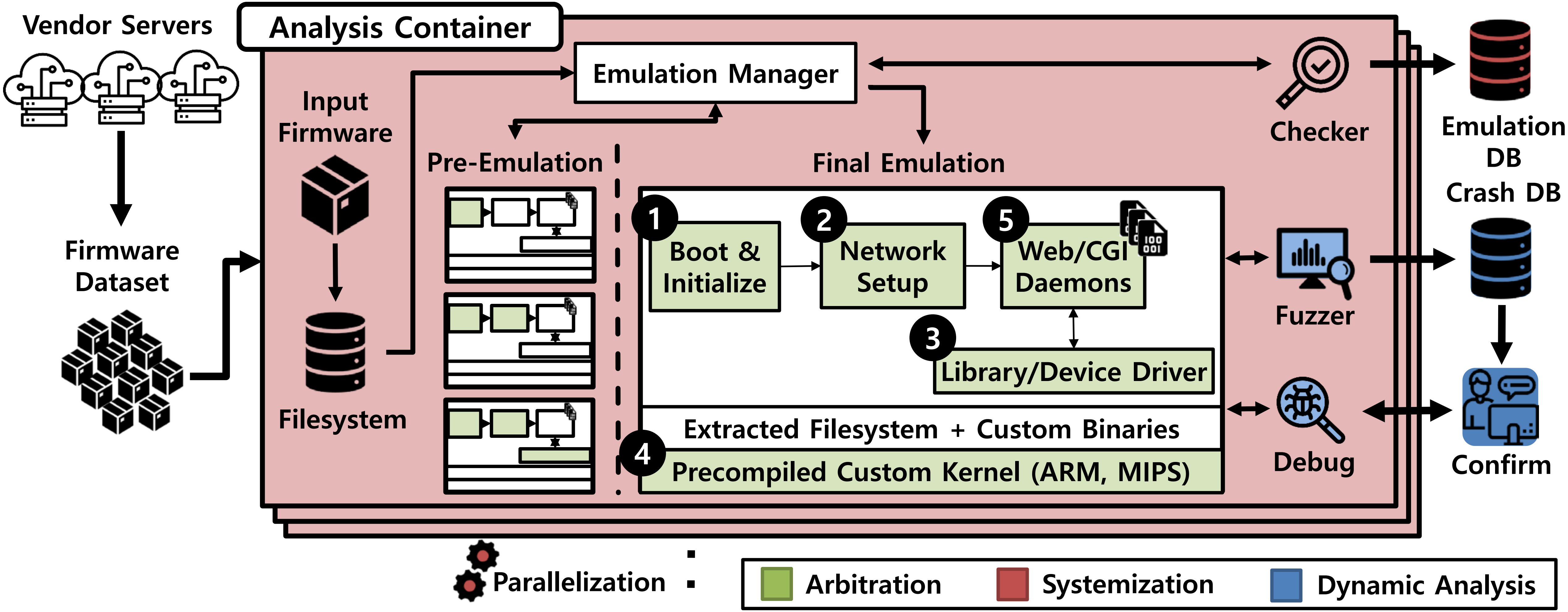
→ Boot environment

→ Network configuration

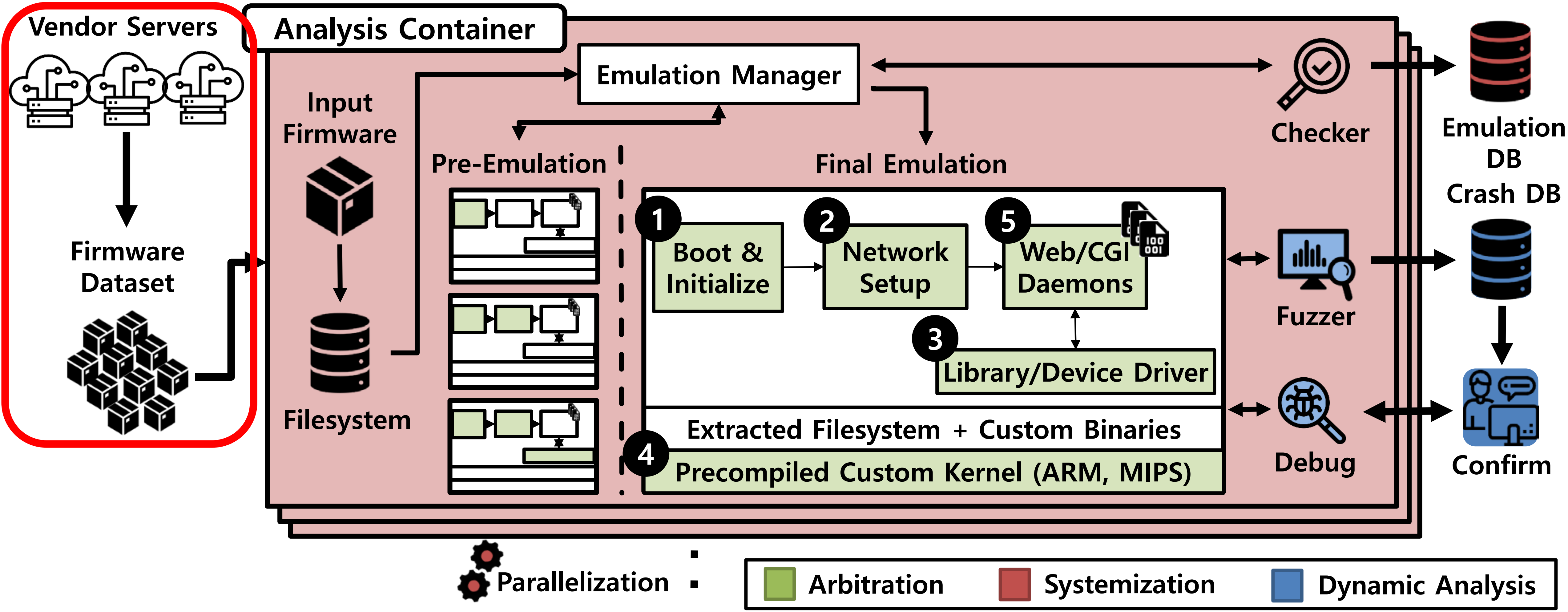
→ Library, device driver, etc.

*Intervention: an intentionally added action

FirmAE overview



FirmAE overview



Dataset building

- Firmware collection
  - Collect firmware from vendor servers
    - Customized scraper based on Firmadyne's + Manual download
  - Extract the filesystem
    - Binwalk: Signature-based file search
  - Target architecture: ARMel, MIPSel, MIPSeb

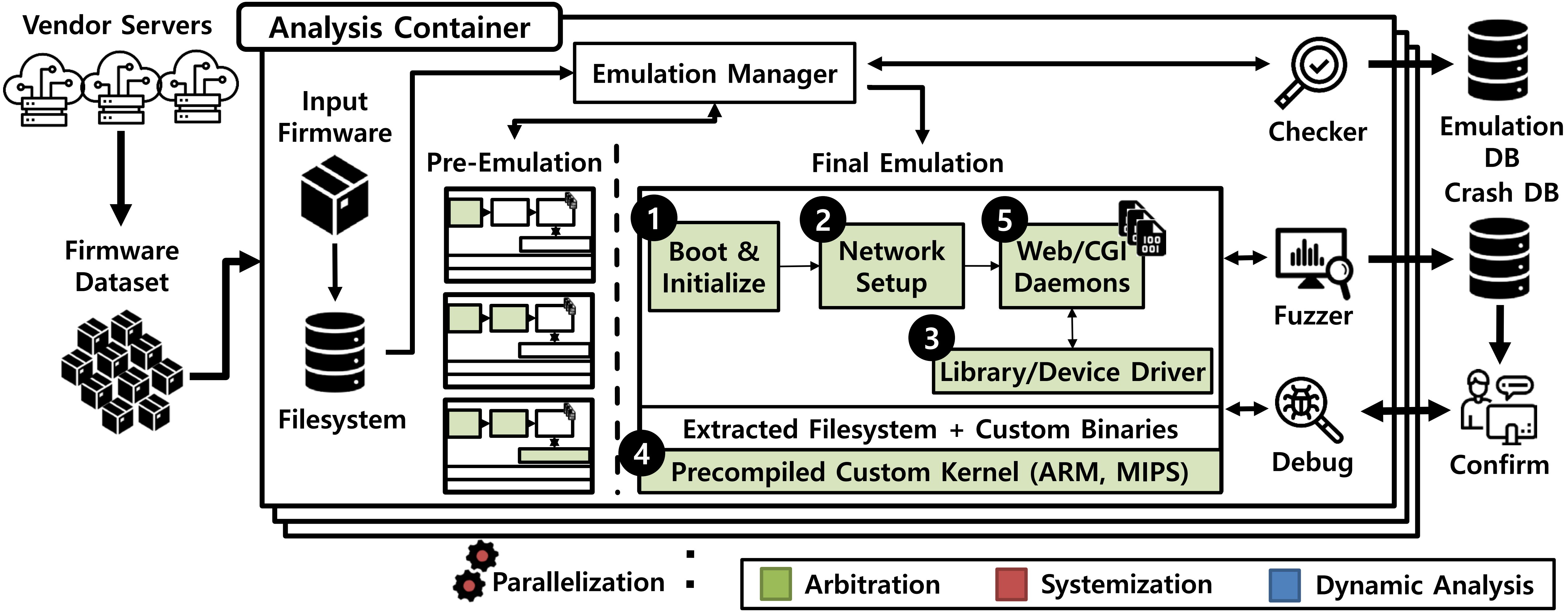
- Dataset (1124 images)
  - AnalysisSet (526 images)
    - Old images from 3 vendors to develop arbitrations
  - LatestSet (553 images)
    - Latest images* from 8 vendors to check the effectiveness of arbitrations
  - CamSet (45 images)
    - Latest images* to evaluate arbitrations in another, yet similar domain

Dataset	Vendor	Images
AnalysisSet	D-Link	179
	TP-Link	73
	NETGEAR	274
Sub Total		526
LatestSet	D-Link	58
	TP-Link	69
	NETGEAR	101
	TRENDnet	106
	ASUS	107
	Belkin	37
	Linksys	55
	Zyxel	20
Sub Total		553
CamSet	D-Link	26
	TP-Link	6
	TRENDnet	13
Sub Total		45
Total		**1124**

*Latest firmware images are checked as of Dec. 2018

FirmAE - Arbitration

Analyze emulation failure cases and resolve them with arbitrations



Arbitration summary

Type	High-level Condition Violation	Intervention
Boot	Improper booting sequence	Identify the initializing program from the kernel of firmware image
	Missing filesystem structure	Make necessary directories by extracting used paths from binaries
Network	Invalid IP alias handling	Fix routing rule to properly handle IP aliasing
	No network information	Add sequence of commands to set up default network interface
	Insufficient support of multiple network interfaces in QEMU ARM	Set a single network interface on QEMU ARM machine
	Insufficient VLAN setup	Fix VLAN configuration on the host system
	Blocked by rules in iptables	Flush the iptables rules
NVRAM	Unknown NVRAM default files	1. Search files that contain key names identified from pre-emulation 2. Initialize NVRAM with found default files
	Crash due to returned NULL pointer	Return an empty string instead of NULL pointer
Kernel	Insufficient support of kernel module	1. Supplement IOCTL handler in the kernel, it can be different by architecture 2. For generalization can be abstracted in LD_PRELOAD library as one function
	Improper kernel version	Upgrade MIPS kernel version to the 4.1, but set 'CONFIG_COMPAT_BRK' to prevent old libc crashes
Others	Unexecuted web servers	Forcibly execute the web servers with appropriate configuration files
	Timeout issues	Increase emulation timeout (Pre: 240s, Final: 360s)
	Lack of tools for emulation	Add full-featured busybox to deal with insufficient command in firmware

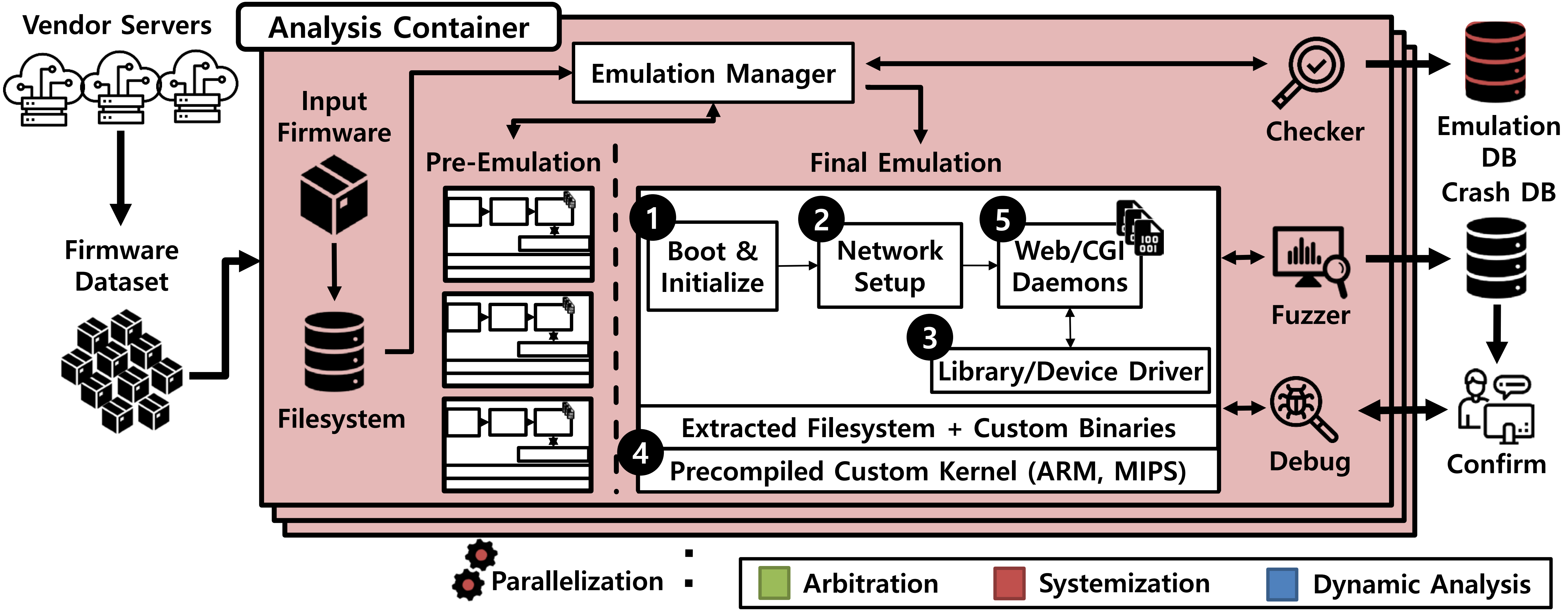
Side-effects of arbitration

- Arbitrations may result in different behaviors against the original hardware
  - It has only slight effect on the security analysis of web services
  - We indeed found several vulnerabilities

- Examples
  - Returning empty string from NVRAM
    - As most values from NVRAM are used for configuration, this may direct the program to use the default value
    - Provides more chance to analyze programs than crashing due to NULL dereference
  - Changing network configuration
    - The network configuration can be different from the original environment
    - However, most vulnerabilities are independent to the network configuration (i.e., IP Address)

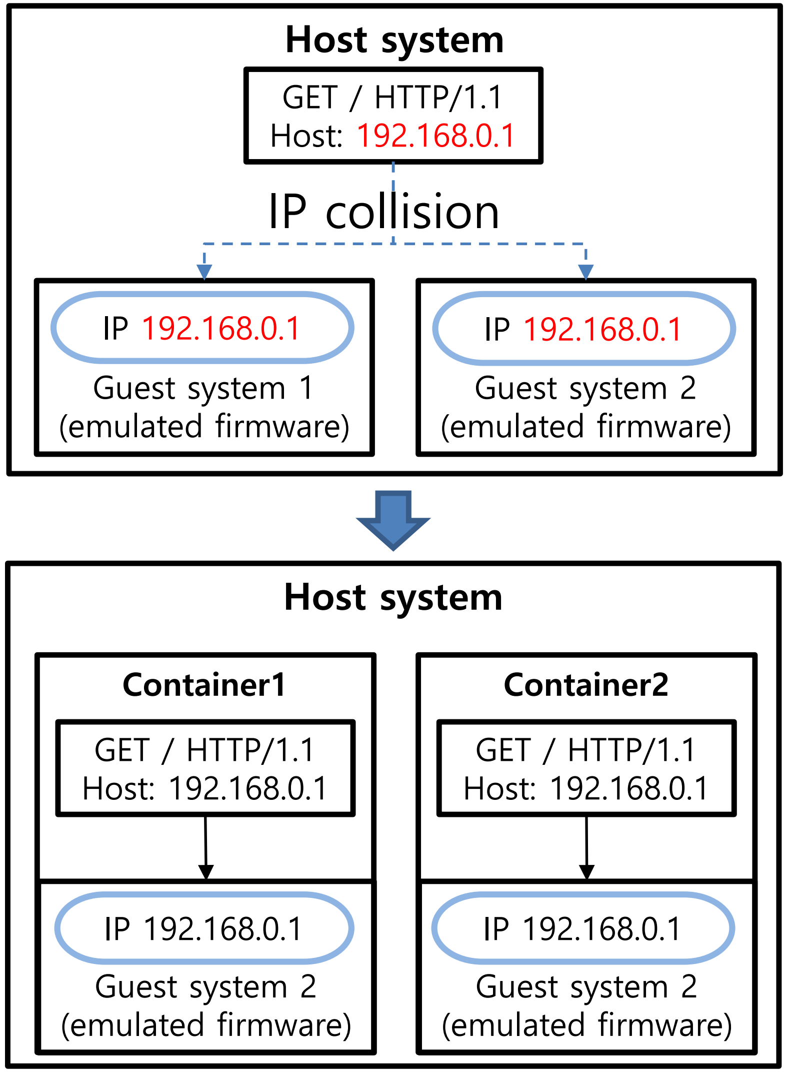
FirmAE - Systemization

Fully-automate and parallelize with containers



Systemization

- Full automation
  - Apply interventions
    - Analyze kernel and filesystem information
  - Check network and web server
    - Use "ping" and "curl"
  - Further analyze vulnerabilities
- Parallelization with containers
  - Make entire firmware emulation/analysis abstract
  - Build an independent network environment
    - Handle network collision from the hard-coded IP addresses



Emulation results

- ❖ Emulation check
  - Network reachability
  - Web service availability

- ❖ Results (vs Firmadyne)
  - AnalysisSet
    - 16.92% → 91.83%
  - LatestSet
    - 16.64% → 69.08%
  - CamSet
    - 4.44% → 60.00%
  - Total
    - 16.28% → 79.36%

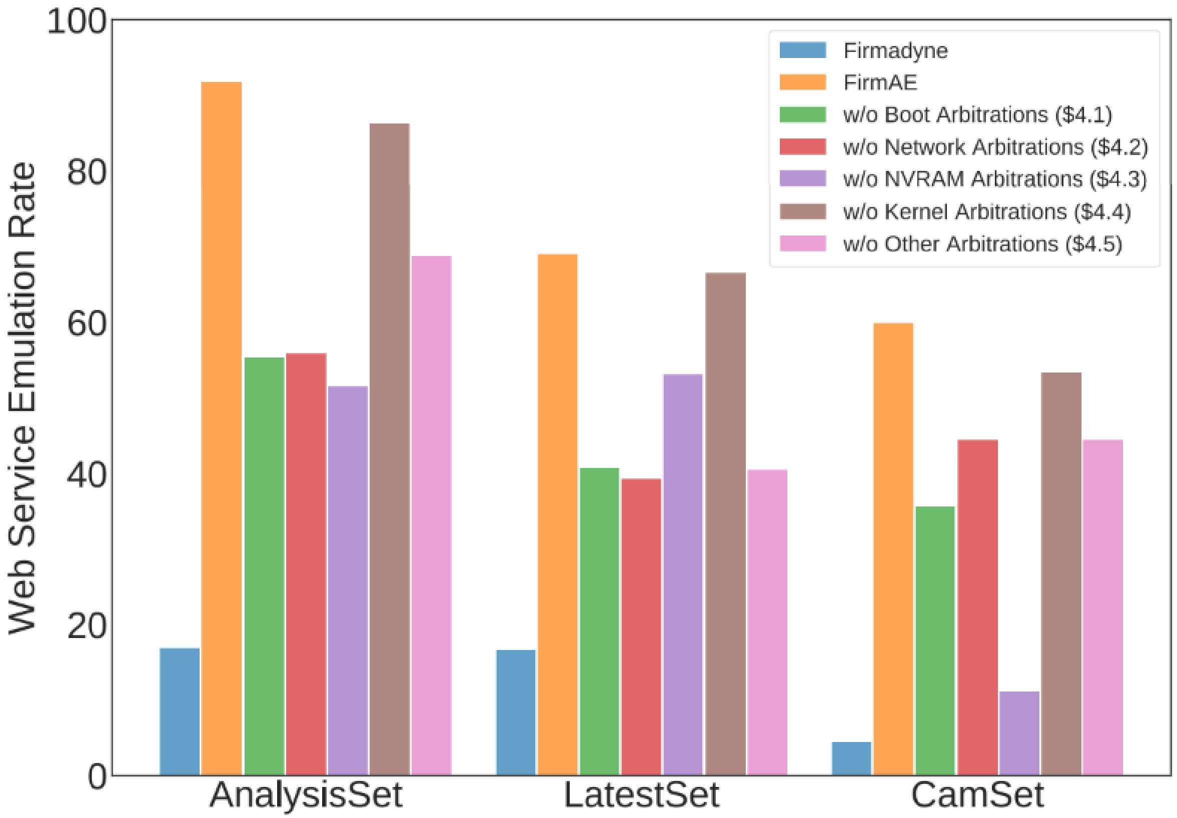
			Firmadyne		FirmAE	
Dataset	**Vendor**	**Images**	**Net**	**Web**	**Net**	**Web**
AnalysisSet	D-Link	179	55 (30.73%)	54 (30.17%)	177 (98.88%)	167 (93.30%)
	NETGEAR	73	26 (35.62%)	5 (6.85%)	73 (100%)	59 (80.82%)
	TP-Link	274	86 (31.39%)	30 (10.95%)	259 (94.52%)	257 (93.80%)
Sub Total		**526**	**167 (31.75%)**	**89 (16.92%)**	**509 (96.77%)**	**483 (91.83%)**
LatestSet	D-Link	58	18 (31.03%)	17 (29.31%)	54 (93.10%)	48 (82.76%)
	TP-Link	69	33 (47.83%)	10 (14.49%)	69 (100%)	54 (78.26%)
	NETGEAR	101	30 (29.70%)	7 (6.93%)	92 (91.09%)	79 (78.22%)
	TRENDnet	106	35 (33.02%)	23 (21.70%)	91 (85.85%)	63 (59.43%)
	ASUS	107	27 (25.23%)	25 (23.36%)	63 (58.88%)	62 (57.94%)
	Belkin	37	2 (5.41%)	2 (5.41%)	30 (81.08%)	22 (59.46%)
	Linksys	55	13 (23.64%)	8 (14.55%)	48 (87.27%)	44 (80.00%)
	Zyxel	20	3 (0.15%)	0 (0%)	18 (0.90%)	10 (50.00%)
Sub Total		**553**	**161 (29.11%)**	**92 (16.64%)**	**465 (84.09%)**	**382 (69.08%)**
CamSet	D-Link	26	0 (0%)	0 (0%)	19 (73.08%)	17 (65.38%)
	TP-Link	6	0 (0%)	0 (0%)	6 (100%)	0 (0%)
	TRENDnet	13	2 (15.38%)	2 (15.38%)	10 (76.92%)	10 (76.92%)
Sub Total		**45**	**2 (4.44%)**	**2 (4.44%)**	**35 (77.78%)**	**27 (60.00%)**
Total		**1124**	**330 (29.36%)**	**183 (16.28%)**	**1009 (89.77%)**	**892 (79.36%)**

Effectiveness of each arbitration

- How to check?
  - Remove each arbitration from full system
  - Check with web service availability

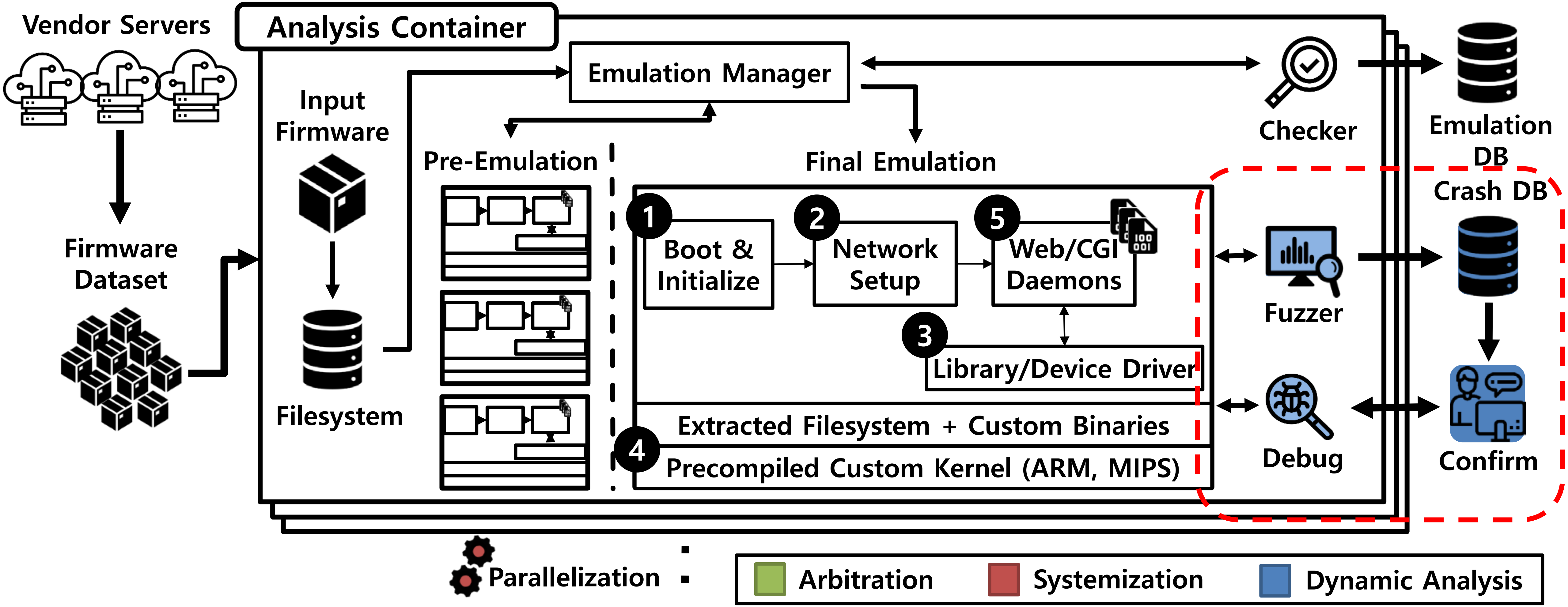
- Results
  - Boot & Network
    - 30% affected
  - NVRAM (the most effective)
    - 35% affected
  - Kernel
    - 4.88% affected
  - Other
    - 22.35% affected

- **All arbitrations are necessary!**



FirmAE - Dynamic Analysis

Dynamically analyze and find vulnerabilities with PoCs and a fuzzer



Conducting dynamic analysis

- For the emulated web services,
  - Initialize webpages by clicking HTML buttons or calling JavaScript functions with Selenium
  - Collect website information from the filesystem
  - Perform dynamic analysis
    - 1-day analysis: RouterSploit (Known PoCs like Metasploit) + Customized PoC
    - 0-day analysis: Our simple fuzzer targets command injection and buffer overflow

- Customized syscall logs
  - Firmadyne's prebuilt kernel significantly helped analyzing the bugs

- Analyses to show the emulation indeed works!
  - 1-day analysis, vs Firmadyne (with AnalysisSet)
  - 1-day analysis, on latest images (with LatestSet)
  - 0-day analysis, on latest images (with LatestSet)
    - CVE hunting!

1-day analysis results on AnalysisSet (vs Firmadyne)

- Is FirmAE effective to reproduce vulnerabilities?

Vulnerability Category	# of POC	**Firmadyne**	**FirmAE**
		# of Images (Unique)	# of Images (Unique)
Information leak	2	0 (0)	17 (17)
Command injection	9	10 (6)	152 (65)
Password disclosure	2	4 (3)	146 (99)
Authentication bypass	2	0 (0)	5 (5)
Total	**15**	**14 (9)**	**320 (128)**

1-day and 0-day analysis results on LatestSet

❖ Is FirmAE effective to find new/unpatched vulnerabilities?

Type	Vulnerability Category	# of Vulns	# of Devices
1-day	Information leak in PHP	1	19
	Information leak in CGI	1	13
	Command injection in UPnP	2	13
	Command injection in SOAP CGI	2	12
	Command injection in HNAP	1	3
	Command injection with backdoor (32764)	2	3
	Path traversal	2	9
Sub Total		11	72
0-day	Command injection in HNAP	6	13
	Command injection in CGI	1	3
	Buffer overflow in HNAP	1	1
	Buffer overflow in CGI	4	6
Sub Total		12	23
Total		**23**	**95**

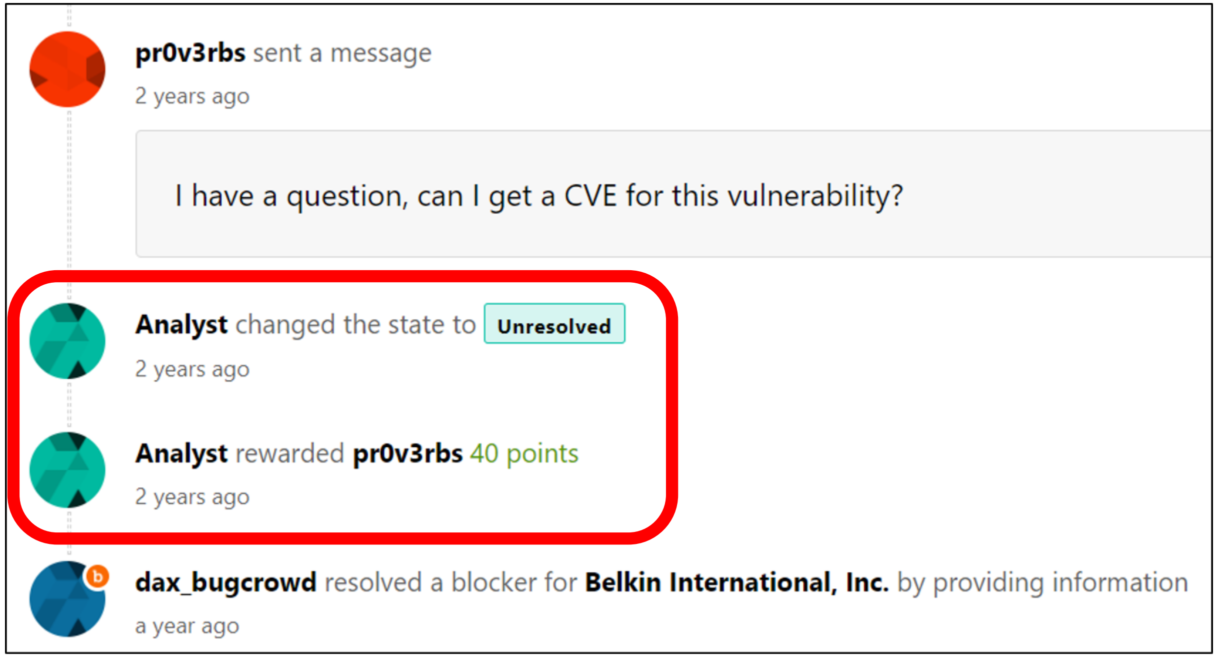
Responsible disclosure

- D-Link
  - HNAP (Command injection, Buffer overflow)
    - SetClientInfoDemo – Deprecated page, but can be identified from filesystem
    - All vulnerabilities are patched by the vendor

- ASUS
  - BOF: Hall of fame (Dec 2019)
    - Reported on Apr 2019
    - Confirmed on Jan 2020

- Belkin
  - Buffer overflow (P1, 40pts from Bugcrowd)
  - Two years passed, no more progress :(

- For more details
  - https://github.com/pr0v3rbs/CVE



pr0v3rbs sent a message
2 years ago

I have a question, can I get a CVE for this vulnerability?

Analyst changed the state to Unresolved
2 years ago

Analyst rewarded **pr0v3rbs** 40 points
2 years ago

dax_bugcrowd resolved a blocker for **Belkin International, Inc.** by providing information
a year ago

Discussion

- Improving emulation rates
  - Developing other arbitration techniques
  - Defining more NVRAM default values and IOCTL functions
  - Investigating other devices types such as Network Attached Storage (NAS)
  - Adopting promising peripheral modeling techniques

- Applying promising analysis techniques
  - Static + Dynamic analysis
  - Targeting other services
    - UPNP, SOAP-CGI, DHCP, and so on

- Developing a honeypot
  - Honware (Vetterl et al., Electronic Crime Research`19)

Conclusion

- What we have done
  - Proposed arbitrated emulation and investigated failure cases
  - Developed its prototype, FirmAE
  - Boosted emulation rate from 16.28% (Firmadyne's) to 79.36% (FirmAE) for 1,124 devices
  - Found 23 new bugs (11 1-days and 12 0-days) affecting 95 unique latest devices

- Lessons learned
  - Many failure cases can be easily resolved by arbitrating the high-level behaviors of firmware
  - This is sufficient for dynamic analysis
  - Emulating diverse embedded devices is challenging, which requires manual efforts

- To support community, we release our source code:
  - https://github.com/pr0v3rbs/FirmAE

Thank You!
Any Questions?

rla5072@nsr.re.kr