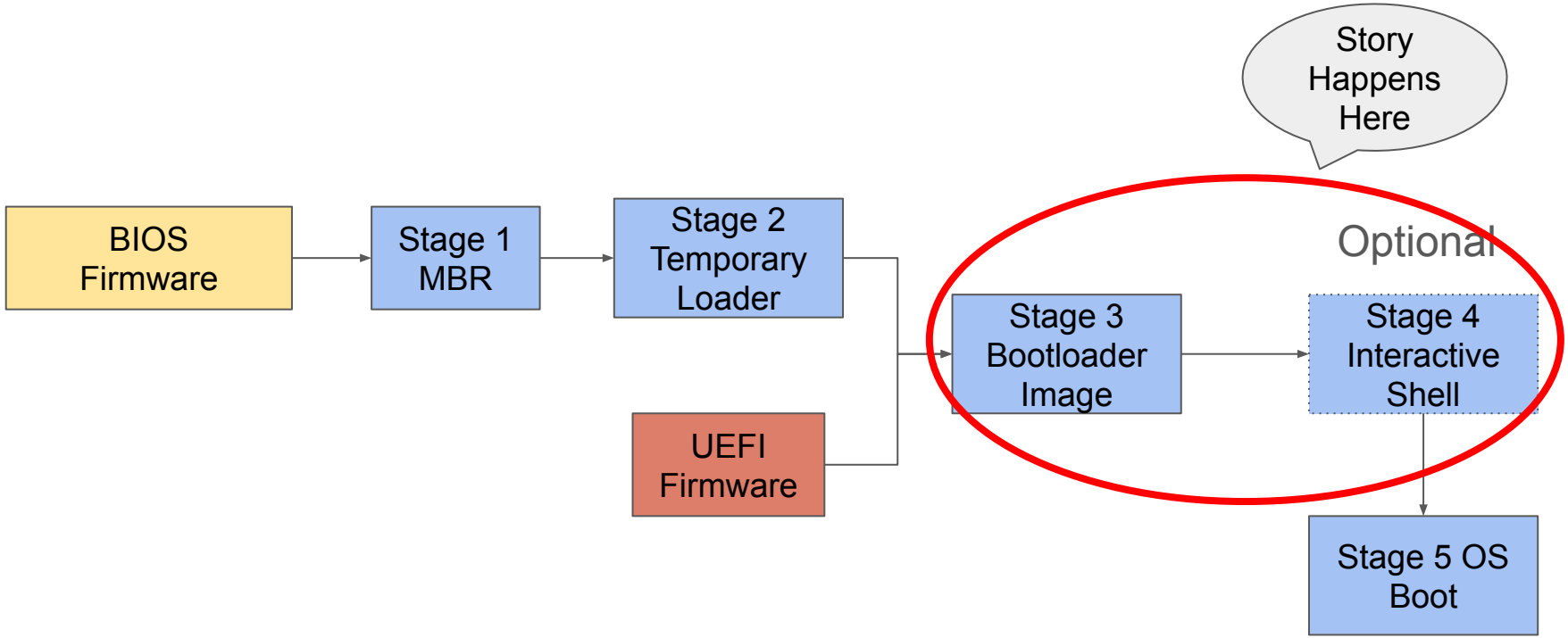
CISPA
HELMHOLTZ CENTER FOR INFORMATION SECURITY

A Comprehensive Memory Safety Analysis of Bootloaders

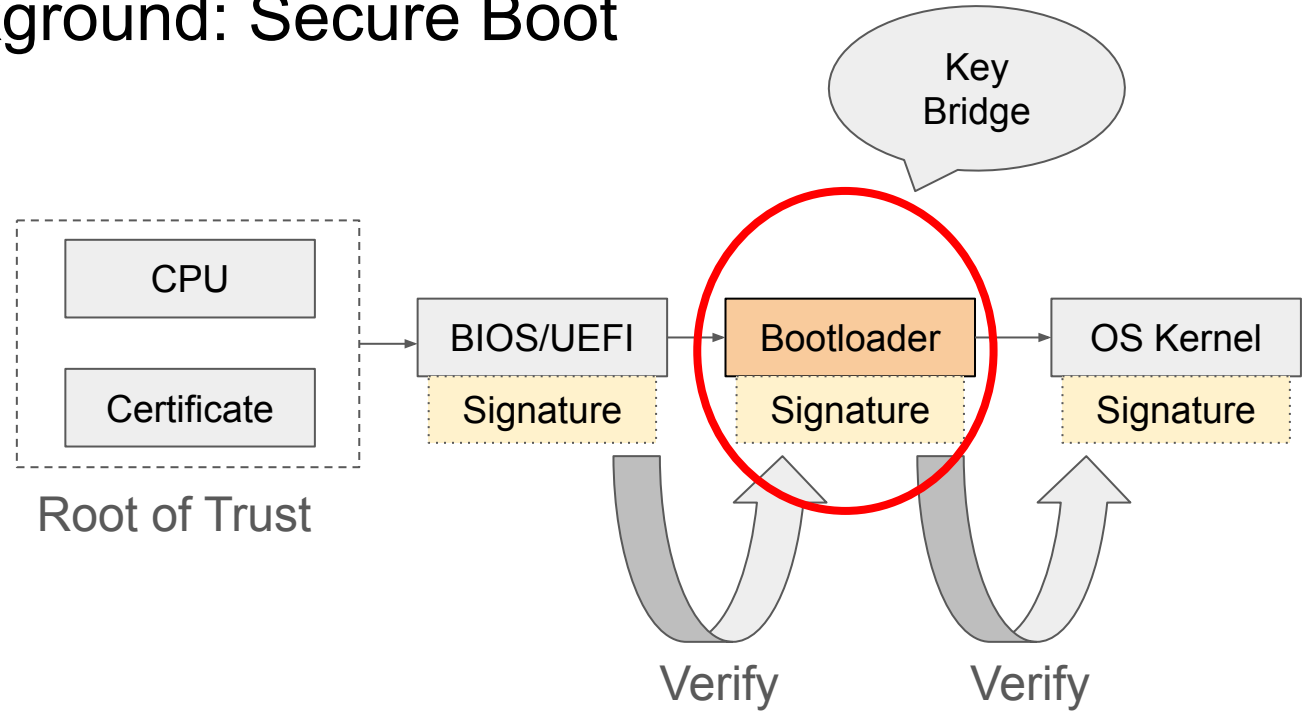
Authors: Jianqiang Wang, Meng Wang, Qinying Wang,
Nils Langius, Li Sh, Ali Abbasi, Thorsten Holz

Presenter: Sahil Sihag

Background: System Boot Workflow



Background: Secure Boot



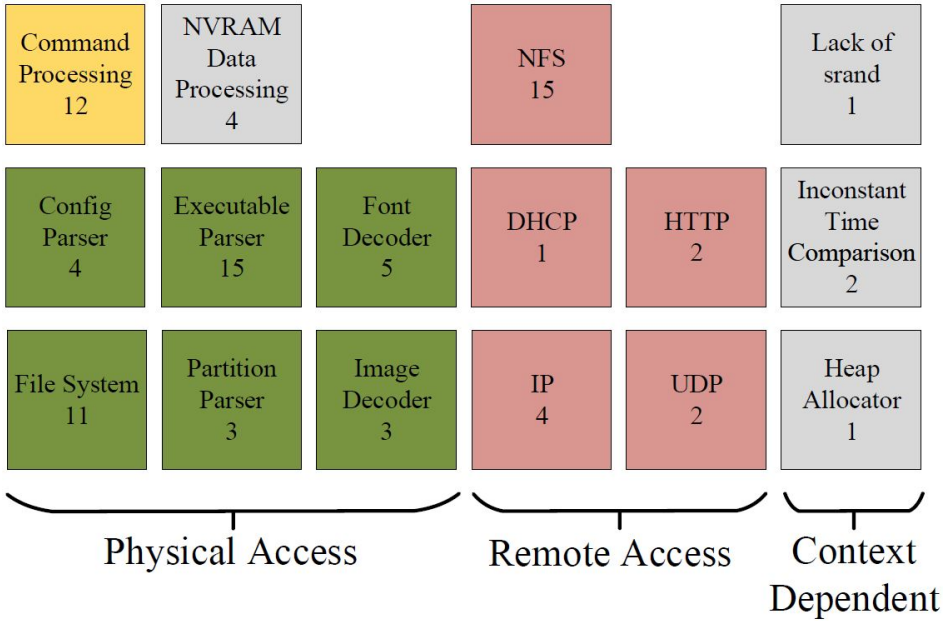
Secure boot functions as a chain of trust,
unmatched signature ⟹ reject loading

Survey: Bootloader CVEs

- 85 vulnerabilities collected
  - Categorized by attacker’s capability
    - physical access
    - remote access
    - Context-dependent
  - Categorized by source
    -
    -
    -

Survey: Bootloader CVEs

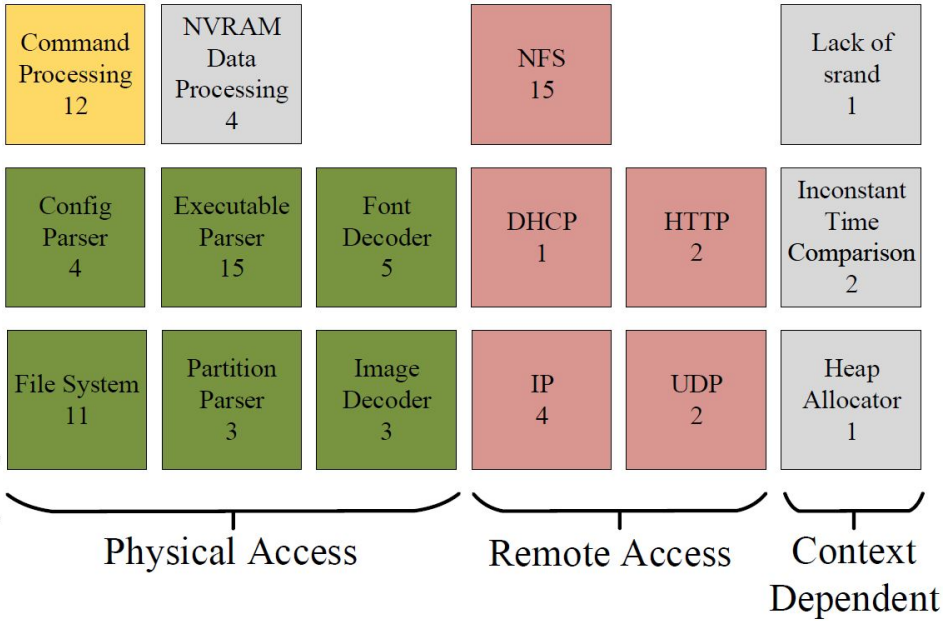
- 85 vulnerabilities collected
  - Categorized by attacker's capability
    - physical access
    - remote access
    - Context-dependent
  - Categorized by source
    - Storage
    - Network
    - Console



Survey: Bootloader CVEs

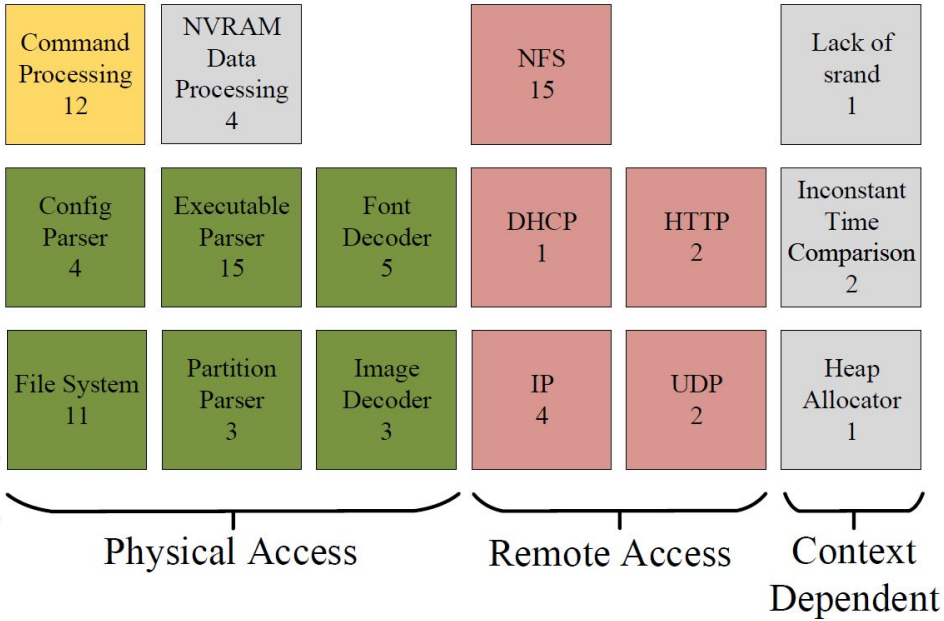
- 85 vulnerabilities collected
  - Categorized by attacker's capability
    - physical access
    - remote access
    - Context-dependent
  - Categorized by source
    - Storage
    - Network
    - Console

	Storage	Network	Console	Others
GRUB	18	2	10	3
barebox	0	3	0	2
shim	8	3	0	2
Das U-Boot	15	16	2	1
	41 (48%)	24 (28%)	12 (14%)	8 (9%)



Survey: Bootloader CVEs

- 85 vulnerabilities collected
  - Categorized by attacker's capability
    - physical access
    - remote access
    - Context-dependent
  - Categorized by source
    - Storage
    - Network
    - Console



	Storage	Network	Console	Others
GRUB	18	2	10	3
barebox	0	3	0	2
shim	8	3	0	2
Das U-Boot	15	16	2	1
	41 (48%)	24 (28%)	12 (14%)	8 (9%)

91% are from storage, network, and console.

Memory Safety Analysis: Threat Model

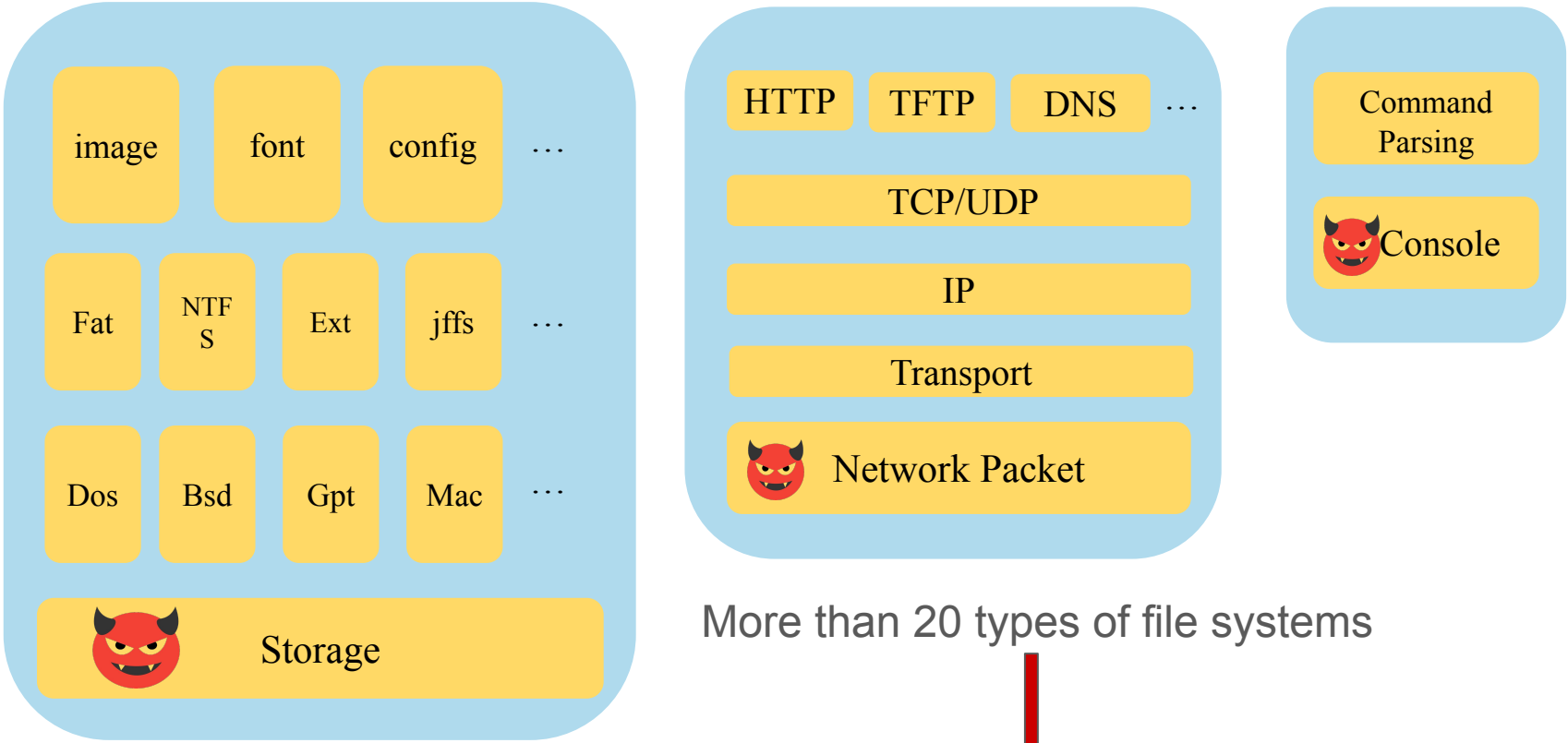
- Attacker’s goal: compromise the system by exploiting memory corruption vulnerabilities
- Attacker’s capability
  - Cannot be directly modified
    - Firmware
    - CPU & memory access
    - Persistent storage access
    - Tampering with bootloader image
  - Limited Access
    - Storage data: can plug in extra storage devices
    - Network data: can send arbitrary network packets
    - Console data: can access the keyboard or any remote terminal

Memory Safety Analysis: Selected Targets

- 9 bootloaders are selected
  - Open-source bootloaders
  - Actively maintained over the past two years
  - Latest version is selected

	Version	Supported Targets	Firmware
GRUB [22]	v2.02-beta2	Linux, GNU/Hurd, macOS, BSD Solaris/illumos (x86 port), Windows	BIOS,UEFI
Limine [32]	v7.x	Linux	BIOS
Das U-Boot [16]	v2024.04-rc3	Linux, NetBSD, VxWorks, QNX RTEMS, INTEGRITY	BIOS
barebox [5]	v2024.01	RTOSes	UEFI
CloverBootloader [11]	v2-5158	macOS	UEFI
Easyboot [9]	v1.0.0	Linux, Windows, OpenBSD, FreeBSD, FreeDOS ReactOS, MenuetOS, KolibriOS, SerenityOS, Haiku	UEFI
rEFInd [47]	v0.14.3	Linux, Windows, macOS, TrueOS	UEFI
systemd-boot [53]	v256	Linux	UEFI
shim [44]	v15.8	GRUB	UEFI

Memory Safety Analysis: Grub as Example



More than 20 types of file systems

Large attack surfaces, vulnerable bootloader

Memory Safety Analysis: Storage

File	config, jpeg, png, font	config, png, bmp, tga	fdt, slre,
File System	zfs, affs, bfs, f2fs	fatfs, iso9660	btrfs, cbfs, ext4fs
Partition	amiga, gpt	gtp, ms-dos	amiga, gpt, ms-dos
	GRUB	Limine	Das U-Boot

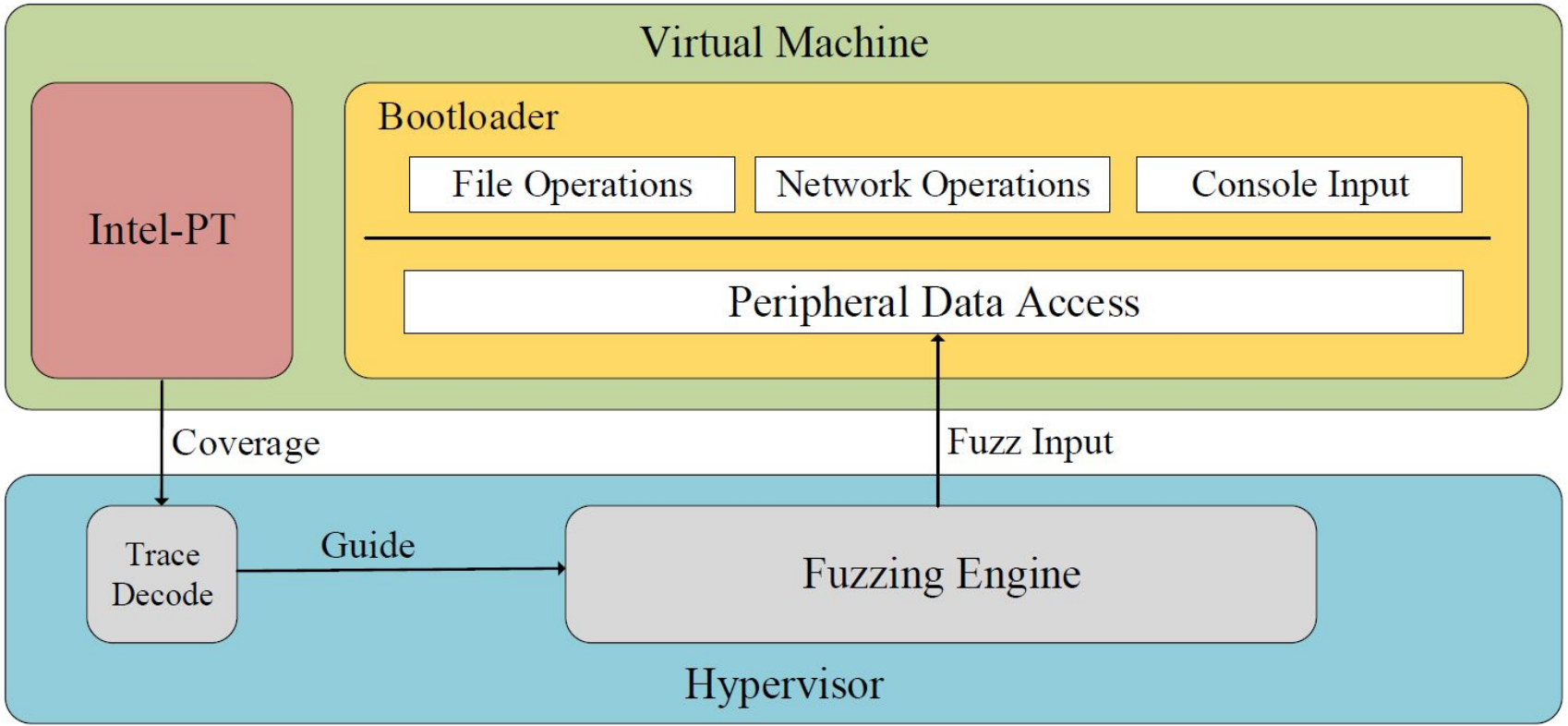
Memory Safety Analysis: Network

Application	HTTP, DNS, TFTP	TFTP wrapper	HTTP, TFTP, NFS
Transport	TCP, UDP	-	TCP, UDP
Network	IP, ICMP, ICMP64	-	IP, ICMP, NDP
Data Link	ETH, ARP	-	ETH, ARP, CDP, RARP
	GRUB	Limine	Das U-Boot

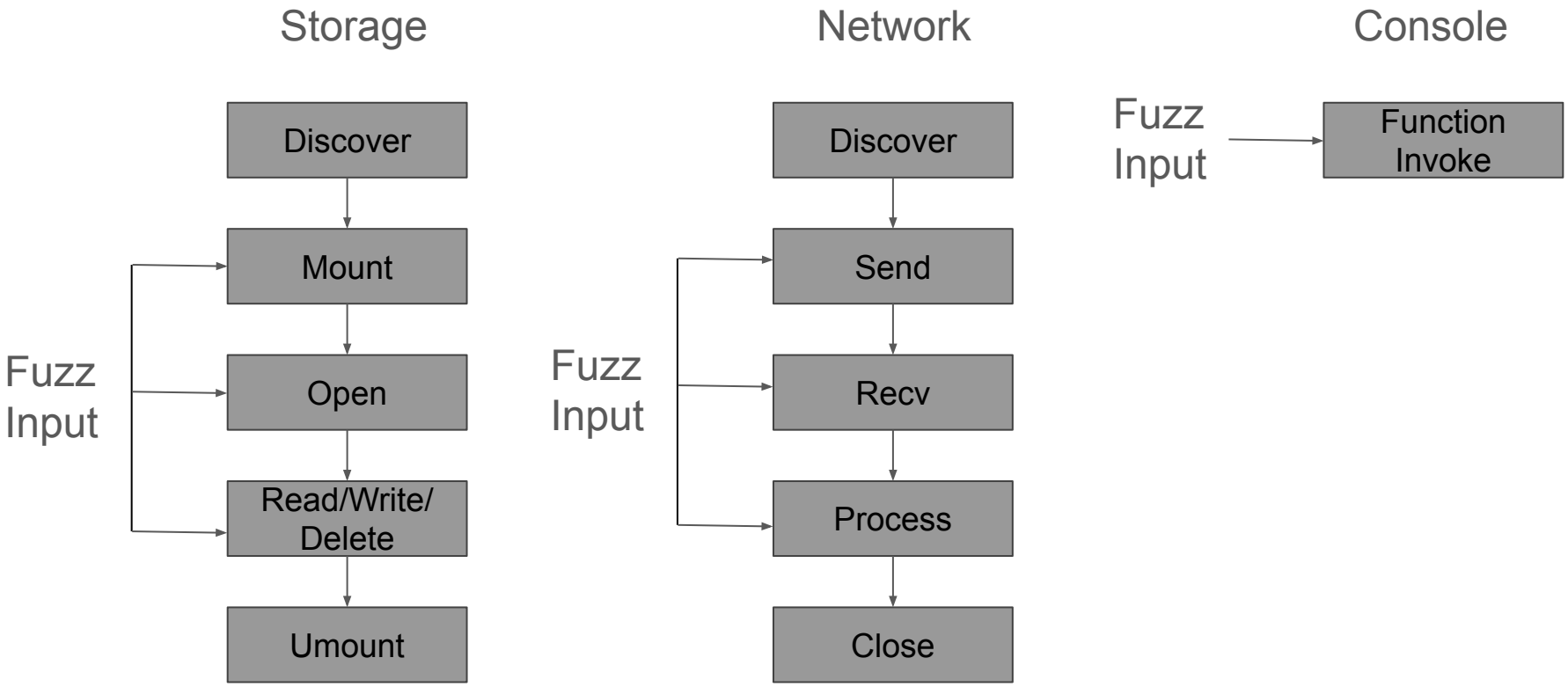
Memory Safety Analysis: Console

Console: Accepts user input as a string and parses it into several options.

Bootloader Fuzzing Framework: Overview



Bootloader Fuzzing Framework: Harness



Bootloader Fuzzing Framework: Crash Detection

- Paging
  - Customized paging for BIOS bootloader
  - UEFI bootloader natively support paging
- Customized Interrupt Handling
  - Customized interrupt service routine hook
- Panic Hook
  - Customized panic function hook
- Heap Sanitizer
  - Customized heap management function hook

Evaluation: Finding New Vulnerabilities

	Bootloader	Category	Type	Status
1	GRUB	Storage, file parser	Logic bug, heap overflow	Confirmed
2	GRUB	Storage, file parser	Integer overflow, heap overflow	Confirmed
3	GRUB	Storage, file parser	Integer overflow, heap overflow	Confirmed
4	GRUB	Storage, file parser	Integer overflow, heap overflow	Confirmed
5	GRUB	Storage, file parser	Logic bug, use of uninitialized data	Confirmed
6	GRUB	Storage, file system	Lack of boundary check, heap overflow	Confirmed
7	GRUB	Storage, file system	Infinite loop, stack overflow	Confirmed
8	GRUB	Storage, file system	Integer overflow, heap overflow	Confirmed
9	GRUB	Storage, file system	Off-by-one access, heap overflow	Confirmed
10	GRUB	Storage, file system	Integer overflow, heap overflow	Confirmed
11	GRUB	Console, command parsing	Unlimited recursion, stack overflow	Confirmed
12	GRUB	Console, command parsing	Missing sanity check, null-pointer dereference	Confirmed
13	GRUB	Console, command parsing	Infinite loop, stack overflow	Confirmed
14	GRUB	Storage, file parser	Off-by-one access, heap overflow	Confirmed
15	Limine	Storage, file parser	Missing sanity check, null-pointer dereference	Patched
16	Limine	Storage, file parser	Logic bug, heap overflow	1-day

39 found,

29 confirmed or patched.

Evaluation: Comparison with Static Analysis

- No ready-to-use bootloader fuzzing tools are available
- Static analysis tools
  - CodeQL
  - Clang Static Analyzer

	CodeQL		CSA		Fuzz	
	TP	Reported	TP	Reported	TP	Reported
GRUB	1	18	1	88	14	14
Limine	0	0	0	2	4	4
Das U-Boot	2	34	0	25	3	4
barebox	0	6	3	19	5	6
CloverBootloader	0	40	0	0	3	3
Easyboot	0	0	0	1	3	5
rEFInd	0	0	0	10	7	7
systemd-boot	0	0	0	6	0	0
shim	0	7	0	0	0	0
	3	105	4	151	39	43

Higher TP, less FP

Conclusion

- Identified the three main attack surfaces (storage, network, and console) in bootloaders based on a survey of 85 CVEs
- Designed and developed a fuzzing framework to test bootloaders
- Discovered 39 new vulnerabilities
  - 29 confirmed
  - 5 CVEs assigned
- Highlight critical areas of concern in bootloader security

Questions

Please send email to jianqiang.wang@cispa.de