Lights Out

Out-of-Band, Out of Mind, Out of Control

HD Moore

https://GitHub.com/runZeroInc/oobscan

runZero

DEFCON

Agenda

OOB & BMCs	What, why, and where
Wins & Losses	Maps, progress, and scoreboard
Bug Beastiary	Vuln classes, flavors, and smells
OOBscan & OSS	Scans, vuln checks, and exploits
VulnDev	Plans, processes, pipelines, payloads, and shells

Acknowledgements

Thank You!

Dan Farmer	Feedback and advice
Gadi Evron	OpenBMC audit collaboration
Fabien Perigaud & Sn0rkY	Green Hills Integrity tips, feedback, and advice
Solar Designer	JtR pull request review & feedback
NetRise	Analyzing many GiB of firmware into nice reports
Dragos Threat Intelligence	Feedback, tool testing, and detections
runZero Research	Review, detection work, and feedback

Prior Work: 2013 to 2019

Year	Researchers	Published work
2013+	Dan Farmer	Sold Down the River / Freight Train to Hell
2014	Wikholm and CARISIRT	Supermicro PSBlock password disclosure
2014	Openwall and Tenable Research	RAKP cracking support, PSBlock detection
2017	Malyutin, Ermolov and Goryachy	Intel SA-00075 and Intel SA-00086
2018	Perigaud, Gazet and Czarny	iLO4 teardown, ilo4_toolbox, RECON
2018	Waisman and Soler, Immunity	iLO2 and IPMI overflows, Black Hat USA
2019	Smith, Jeffery and Stanley	CVE-2019-6260, Pantsdown
2019	Nicholas Iooss	iDRACKAR, host RAM read from the BMC
2019	Czarny, Gazet and Guinet	Defeating NotPetya from an iLO4
2019	Eclypsium Research	Avocent MergePoint EMS, CRC32 updates
2019	Altherr and Eclypsium	USBAnywhere and CloudBorne
2019	Niewohner	smcbmc, Supermicro firmware decryption

Prior Work: 2020 to 2025

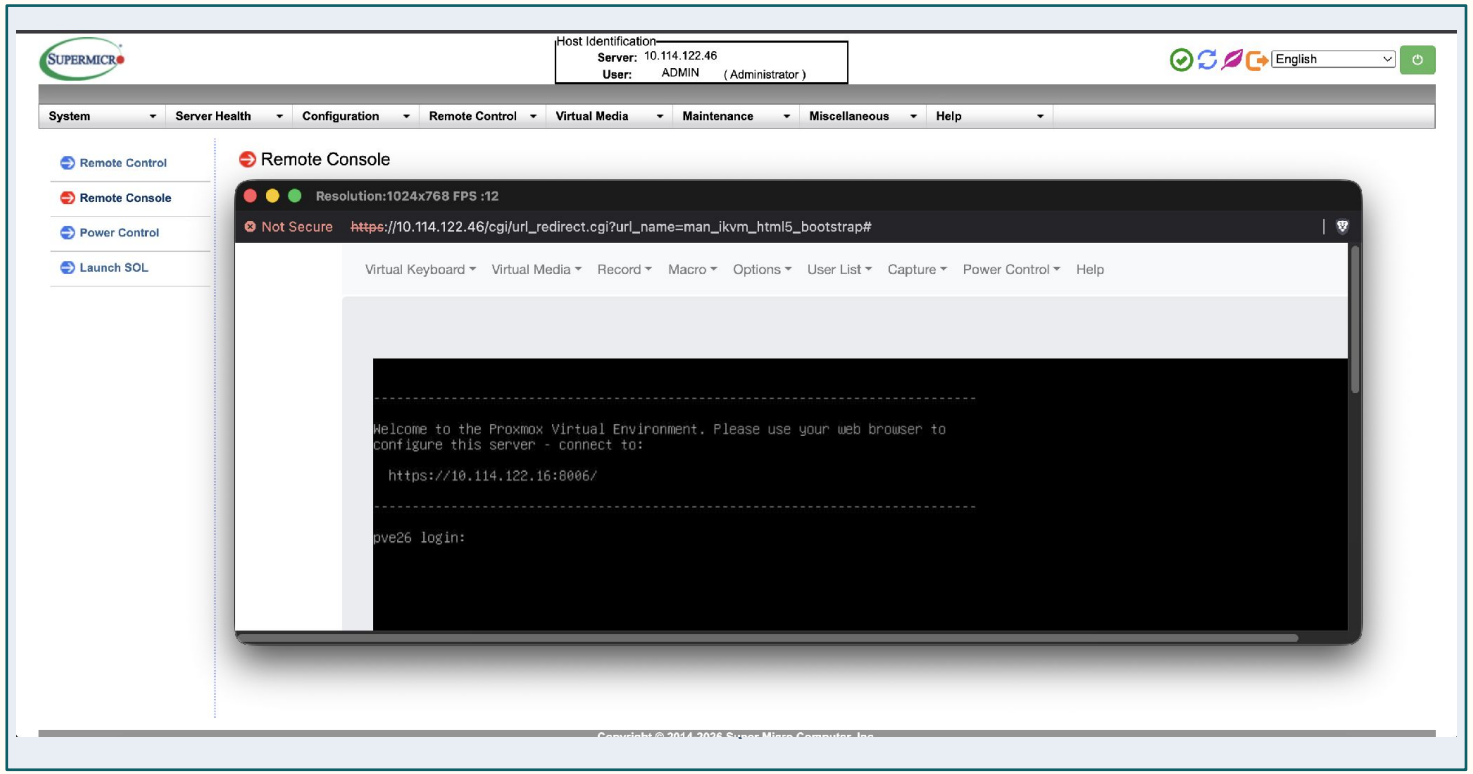
Year	Researchers	Published work
2020	Kiguradze and Ermolov	CVE-2020-5366, iDRAC9 path traversal
2021	Google security research	CVE-2021-39296, phosphor-net-ipmid
2021	Amnpardaz	iLOBleed, resident iLO4 implant
2021	Analysed by Eclypsium Research	AMI MegaRAC source leak, RansomExx breach
2022 to 2025	Babkin and Scheferman	BMC&C parts one to three, AMI MegaRAC
2023	Binarly	Old But Gold, decades-old BMC attacks
2023	Tereshkin and Zabrocki, NVIDIA	Breaking BMC, 18 bugs, DEF CON 31
2024	Tereshkin, analysed by Binarly	CVE-2024-36435, Supermicro pre-auth

OOB & BMCs

Out-of-Band Management

- The official backdoors for important systems
- Serial ports, power controls, KVMs
- Baseboard management controllers (BMCs)

Out-of-Band and In-the-Box



- Hardware & power control
- Keyboard, video, and mouse
- Boot media over virtual USB
- BIOS/firmware updates
- Host serial console
- Often 32-bit ARM
- NICs shareable

- Vendor flavors
  - HPE iLO
  - Dell iDRAC
  - Lenovo XCC
  - Supermicro IPMI
  - H3C HDM
  - Huawei BMC

Exposure Stats

Data sources

- **Private:** Anonymized Sample
- **Public:** Shodan + IPv4 IPMI

Trivial exploitation*

- **Internal:** 12% → 30%
- **External:** 8% → 21%

* Includes default/weak passwords, auth bypass to escalation chains, and remote code exec. Does **NOT** include all RAKP hash exposure (46% & 19%).

OUT-OF-BAND EXPOSURE

Internal vs External

prevalence
% of each source's own population

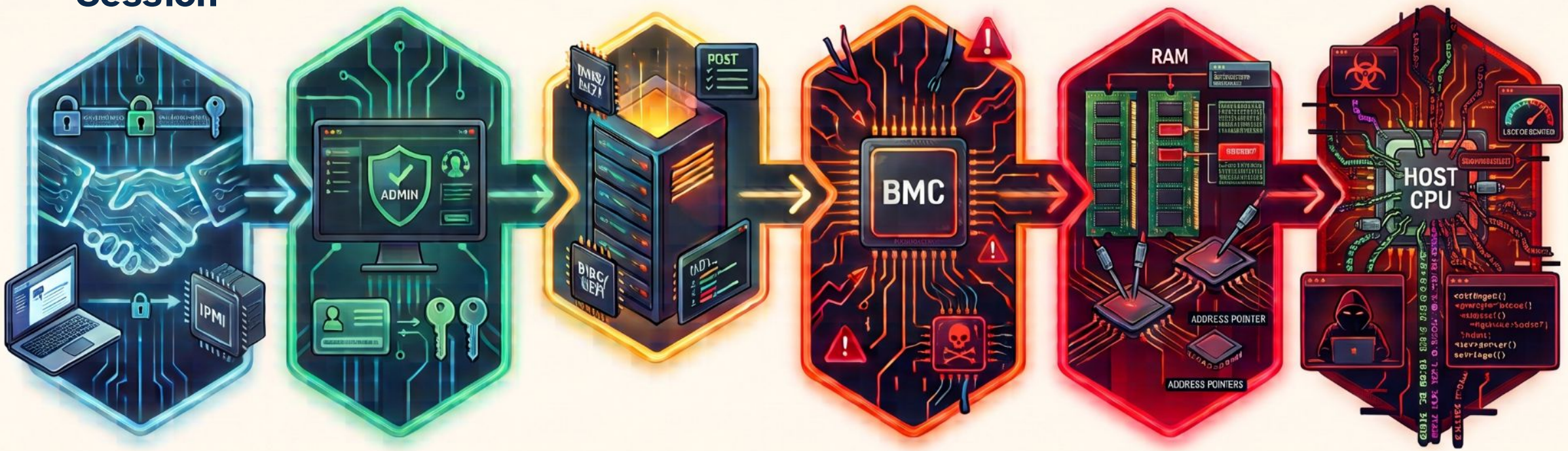
	INTERNAL	EXTERNAL
HOSTS SCANNED	**126,779** 100% · population	**125,430** 100% · population
Honeypot decoys (excluded) counted in scanned; findings withheld	**0** 0.0%	**28,905** 23.0%
IPMI reachable	**69,327** 54.7%	**50,926** 40.6%
RAKP hash disclosure	**58,148** 45.9%	**23,288** 18.6%
IPMI default credentials	**10,752** 8.5%	**2,023** 1.6%
Cipher-zero session	**1,538** 1.2%	**624** 0.5%
Weak credentials non-default, recovered by cracking	**16,378** 12.9%	**1,897** 1.5%
Trivial exploitation (before)	**15,442** 12.2%	**2,545** 2.0%
Extended with our findings	**38,441** 30.3%	**26,871** 21.4%

Goal: Remote Unauthenticated *Host* Code Execution

IPMI Session

Host Boot

Host Memory Read

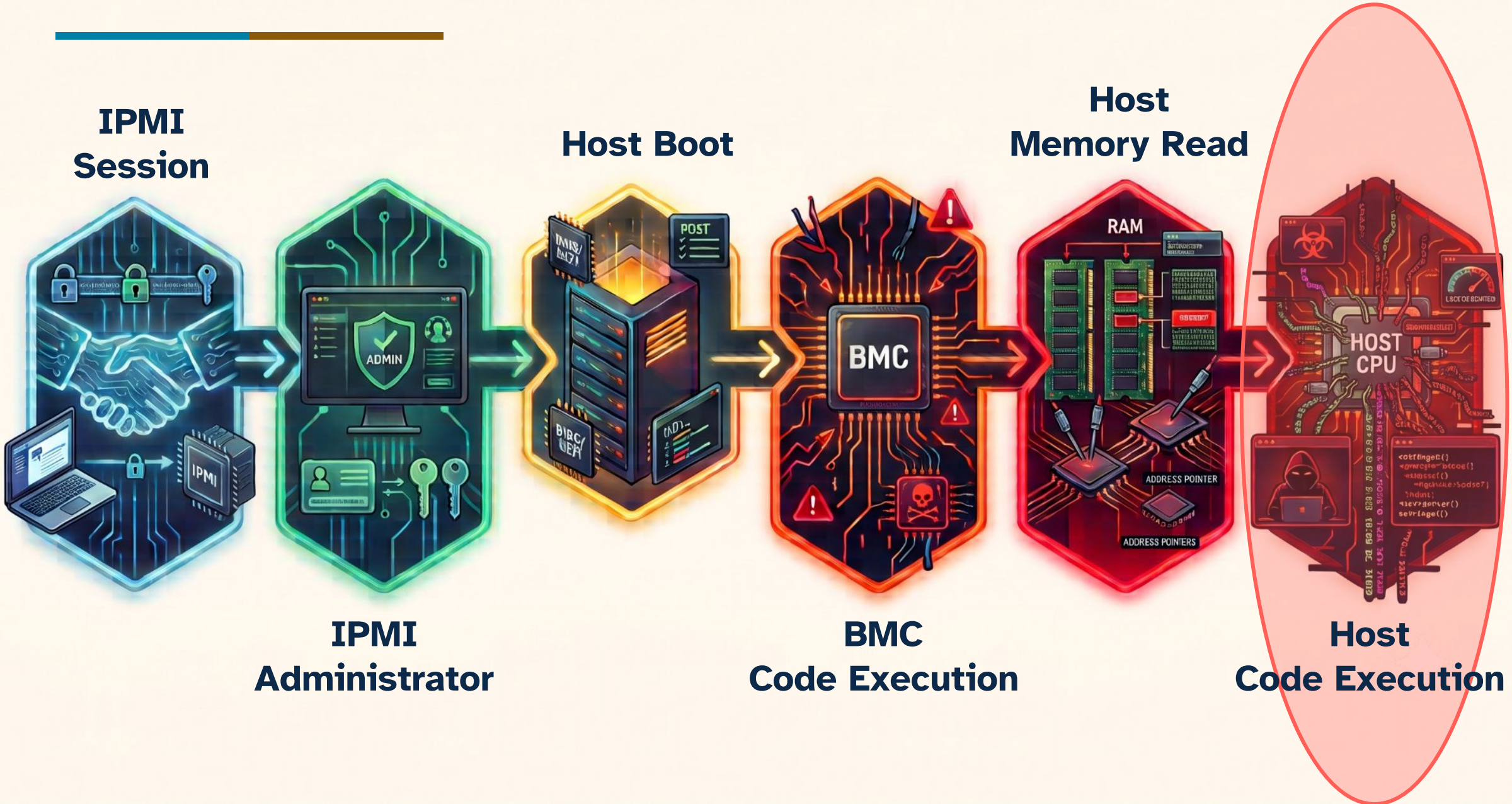


IPMI Administrator

BMC Code Execution

Host Code Execution

Goal: Remote Unauthenticated *Host* Code Execution



Wins & Losses

Highlighted Results

	Unauth to session	Low priv to admin	Admin to BMC exec	Admin to firmware install	Firmware to host exec
HPE iLO4	●	●		●	●
HPE iLO5/iLO6	●	●		●	●
Supermicro X10-H13/X13	●	●	●	●	○
Supermicro X14	●	●		○	○
OpenBMC IPMI	●	●		○	○
NVidia BlueField 3	●	●		○	○
Dell iDRAC 10				○	○

● demonstrated ○ conditional

Authentication Bypass

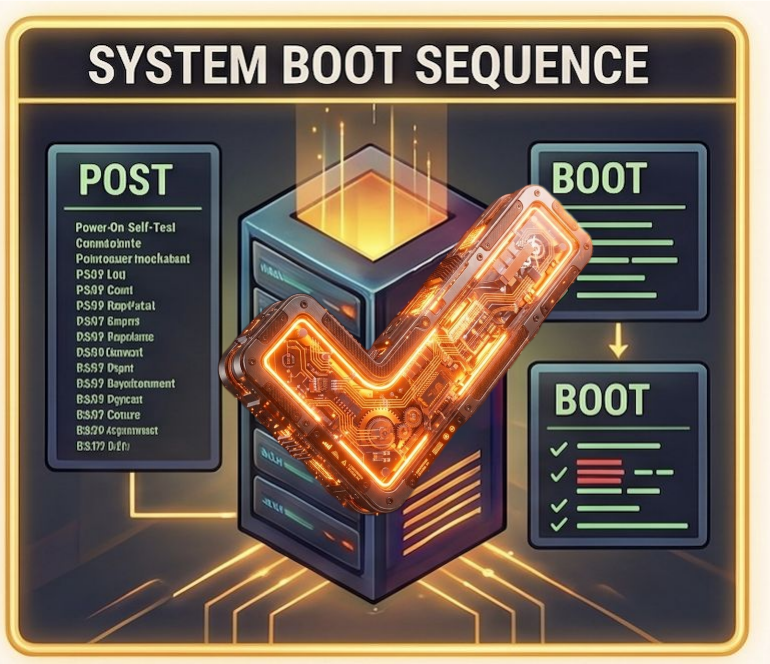
- IPMI 2.0 RAKP state transition bugs
- IPMI 1.5/2.0 type confusion
- Vendor-specific RAKP modes
- Code exec via memory corruption

Privilege Escalation

- IPMI 2.0 Set Privilege
- IPMI 2.0 Session Relabeling
- Depends on Integrity/Encryption bypass

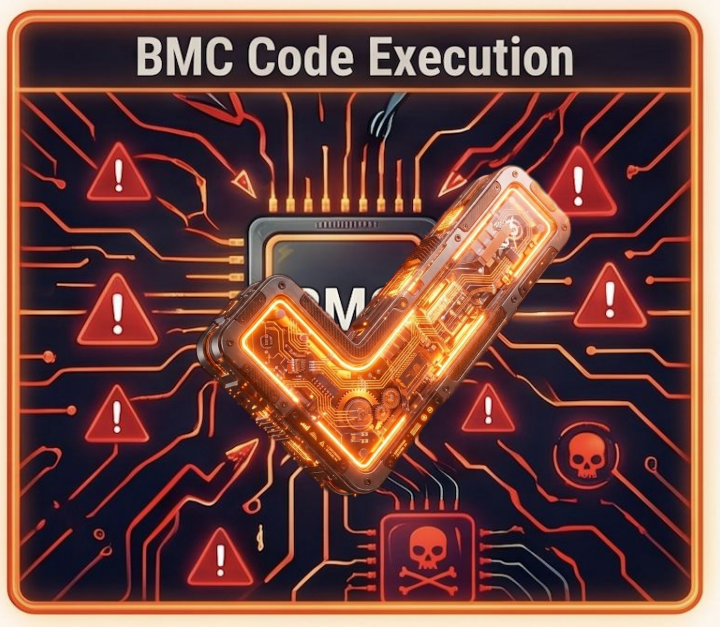
Host Boot Control

- Virtual Media & bootloader prompts
- Serial access via SOL for u-boot



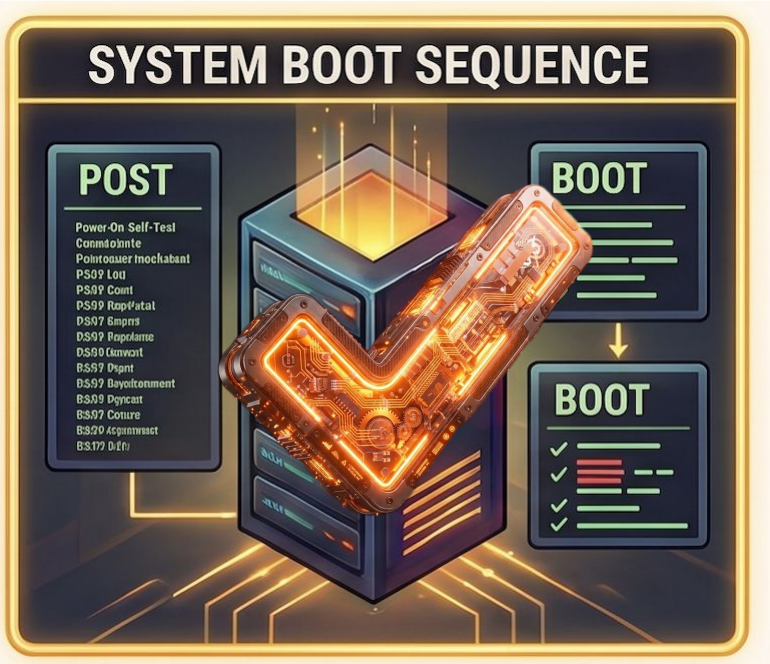
BMC Code Execution

- Backup & Restore backdoored config file
- Force firmware downgrade as needed
- Direct physical/virtual memory access
- Less command injection than before



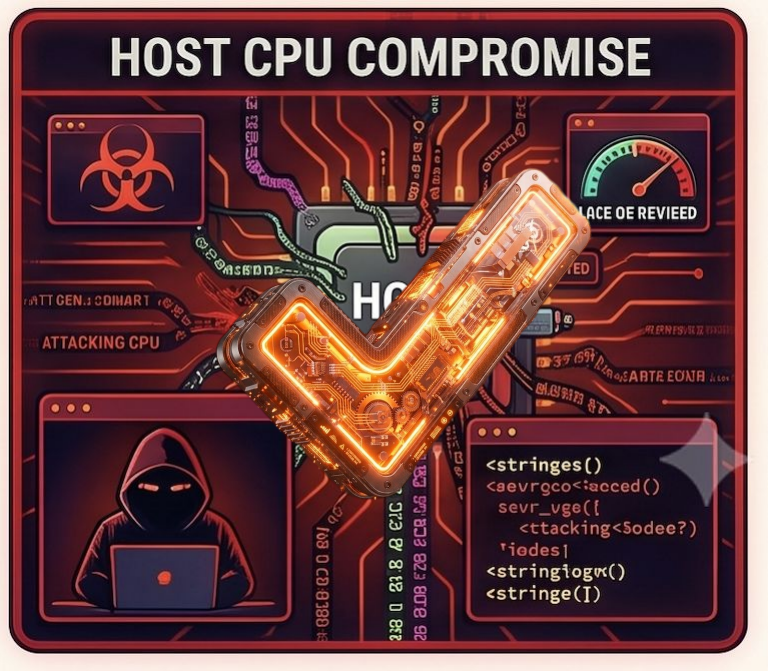
Host Memory Read

- HPE iLO: CHIF driver supports full read/write
- ASPEED: DMA/X-DMA is there, but tricky
  - AS2500: BME, 32-bit limit, few targets
  - AS2600: No BME, but 64-bit addresses
  - IOMMU & VT-D can actually matter



Host Code Execution

- HPE iLO: CHIF still makes this trivial
- Force firmware downgrade as needed
- DMA/X-DMA is very close on SMC/ASPEED
- AST2600: eSPI can **write**, not read



Goals: Mostly Complete (if spotty)

IPMI Session

Host Boot

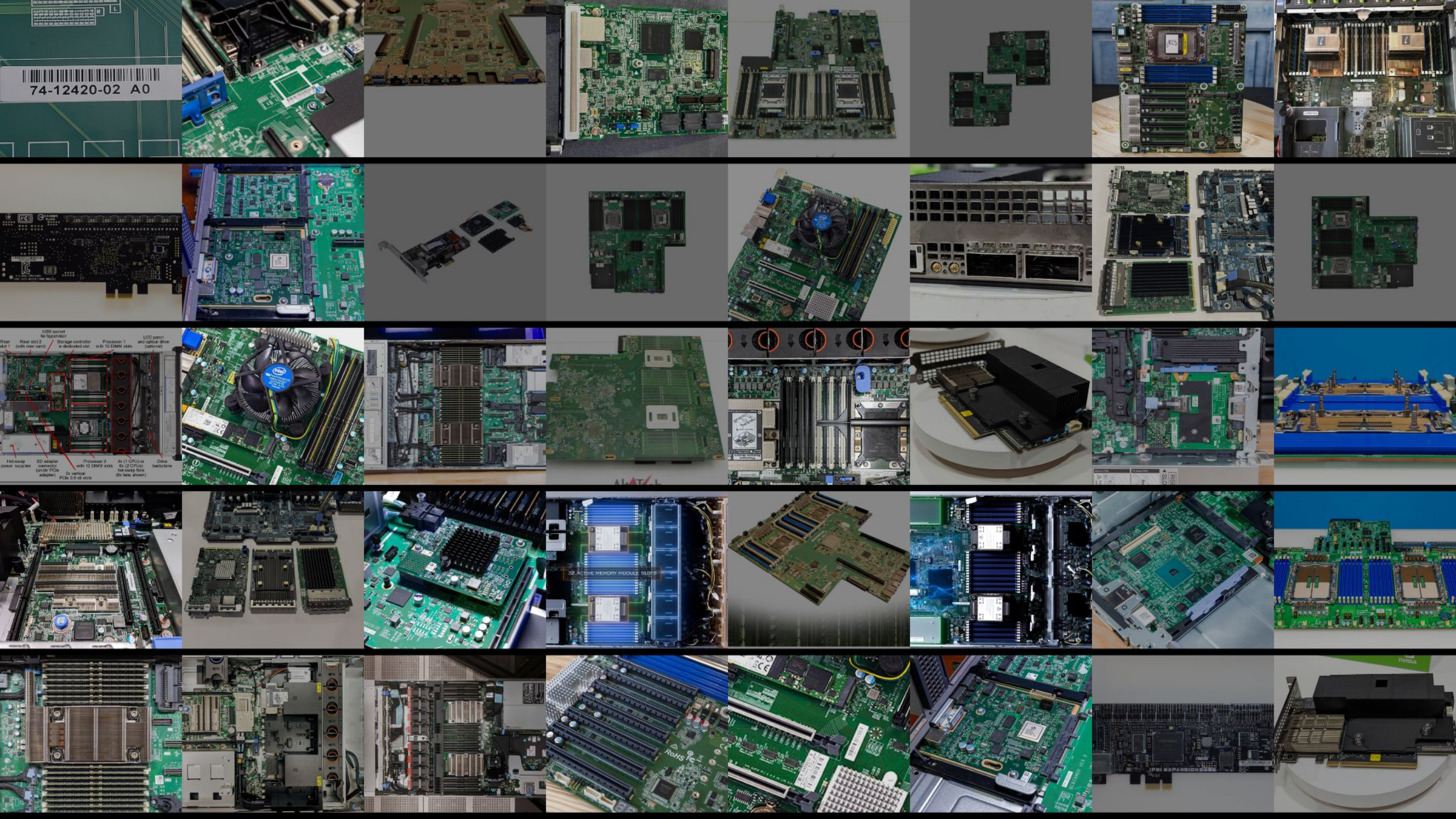
Host Memory Read

IPMI Administrator

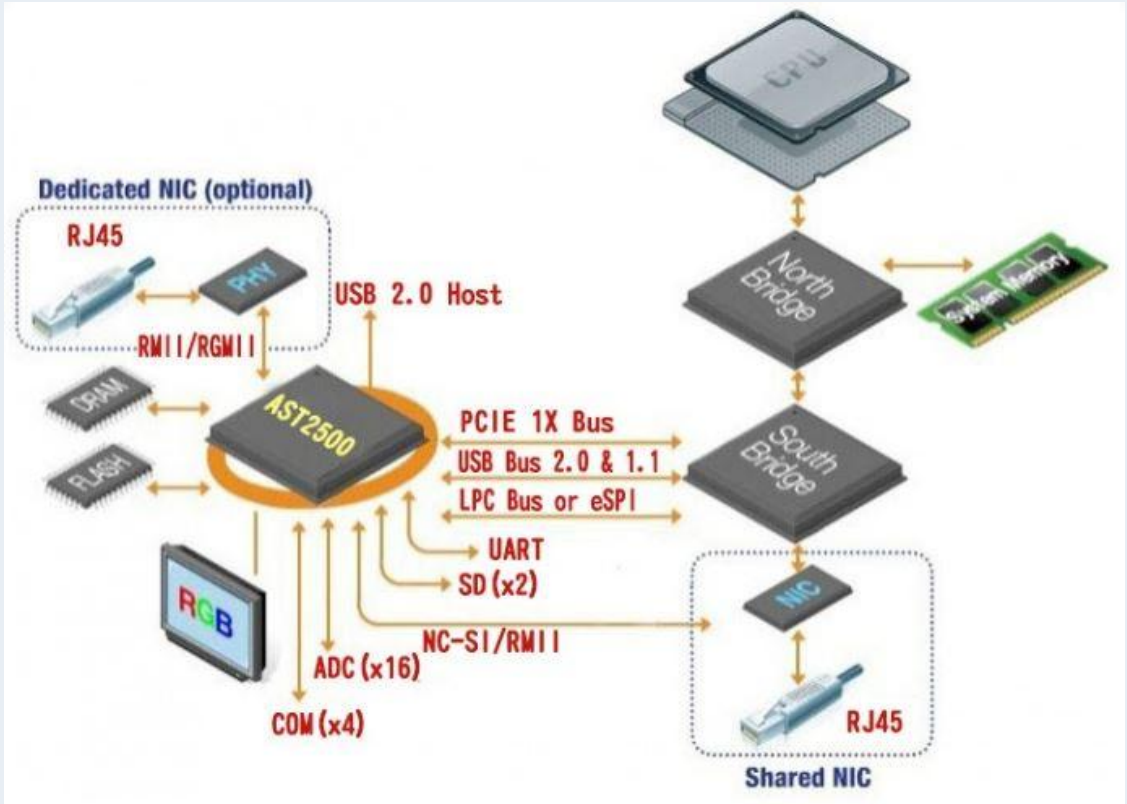
BMC Code Execution

Host Code Execution

Hardware


74-12420-02 A0
USB socket for hypervisor
Riser slot 1
Riser slot 2 (with riser card)
Storage controller in dedicated slot
Processor 1 with 12 DIMM slots
LCD panel and optical drive (optional)
Hot-swap power supplies
SD adapter connector (under PCIe adapter)
2x vertical PCIe 3.0 x8 slots
Processor 2 with 12 DIMM slots
4x (1 CPU) or 6x (2 CPUs) hot-swap fans (6x fans shown)
Drive backplane
32 ACTIVE MEMORY MODULE SLOTS
RoHS
PMI EXPANSION SLOT
NVIDIA

ASPEED AST2500 Block Diagram



ASPEED AST2500 vendor block diagram, reproduced unmodified

Two NIC paths

Dedicated NIC at upper left, marked optional. Shared NIC over NC-SI and RMII at lower right.

LPC or eSPI to the south bridge

The IPMI system interface, the KCS channel to the host. It carries no authentication from the host side.

PCIe, USB, video, DRAM, flash

Virtual media and KVM traverse these buses. The DRAM and the flash belong to the controller, on the auxiliary rail.

AST2050 / AST1100

TSMC 0.13um, 19x19mm 355-pin TFBGA, datasheet V1.05 (2010)

2nd generation

CPU AND MEMORY

- **ARM926EJ-S** — 200 MHz, 16K+16K cache
- not present
- **SMC** — SPI NOR
- **DDR / DDR2 ctrl** — 16-bit, 200 MHz, max 128 MB
- not present
- **AHBC** — AHB controller

M-Bus

AHB Bus

ENGINES

- **USB2.0 virtual hub** — 21 endpoints
- not present
- **10/100 MAC** — x2
- not present
- **MDMA engine** — memory to memory
- **Hash & Crypto**
- not present
- not present
- **MIC engine** — memory integrity check
- **Video engine** — AST2050 only, RC4 stream
- not present
- not present
- not present
- **SCU** — PLL x3

APB Bus

OUT OF BAND: THE SIDE THAT TOUCHES THE BOARD

- **I2C / SMBus** — x7
- not present
- **UART** — x2
- **Virtual UART** — x1
- **GPIO** — 46 pins
- not present
- **PWM** — x4
- **Fan Tach** — x16
- not present
- **PECI**
- **Timer** — x3
- **WDT**
- **RTC**
- not present
- not present
- not present

IN BAND: THE SIDE THAT TOUCHES THE HOST

- **P2A bridge** — PCI to AHB
- not present
- not present
- **LPC to AHB bridge** — registers only, not in figure
- **VGA controller** — 1600x1200
- **2D engine**
- not present
- not present

P-Bus

HOST AND FLASH ATTACH

- not present
- **SPI** — VGA BIOS
- **LPC master / slave**
- not present
- **PCI 2.3** — 32-bit 33 MHz slave

Bridges the vendor figure never draws: LPC-to-AHB (ENL2H) is in the register map of every generation from AST2050 on, and so is P2A.

Redrawn from ASPEED datasheet section 1.2 (chip architecture) and section 1.3 (feature set). Same grid for every generation, so the holes line up.

AST2050 / AST1100 to AST2400 / AST1250

2nd to 5th generation: AST2150 (3rd) and AST2300 (4th) are skipped because no public block diagram exists for either

2nd to 5th

14 added, 1 gone, 20 changed

CPU AND MEMORY

- **ARM926EJ-S** — 400 MHz, 16K+16K cache — was: 200 MHz, 16K+16K cache
- **ColdFire V1** — 2 KB cache, AST2400 only
- **SMC** — NOR / NAND / SPI — was: SPI NOR
- **DDR2 / DDR3 ctrl** — 16-bit, 400 MHz (DDR3-800) — was: DDR / DDR2 ctrl
- **SRAM 32 KB**
- **AHB-MBus bridge** — was: AHBC

M-Bus

AHB Bus

ENGINES

- **USB2.0 hub + 1.1 HID** — was: USB2.0 virtual hub
- **USB2.0 / 1.1 host**
- **1 Gbps MAC** x2 — was: 10/100 MAC
- **SD / SDIO** x2
- **MDMA engine** — memory to memory
- **Hash & Crypto**
- not present
- not present
- **MIC engine** — was: memory integrity check
- **Video engine** — compression — was: AST2050 only, RC4 stream
- not present
- not present
- **Graphics CRT**
- **SCU** — PLL x4 — was: PLL x3

APB Bus

OUT OF BAND: THE SIDE THAT TOUCHES THE BOARD

- **I2C / SMBus** x14 — was: x7
- not present
- **UART** x5 — was: x2
- **Virtual UART** x1
- **GPIO** 216 pins — was: 46 pins
- **SGPIO** x80
- **PWM** x8 — was: x4
- **Fan Tach** x16
- **ADC** 16 ch
- **PECI**
- **Timer** x8 — was: x3
- **WDT** x2
- **RTC**
- **MailBox**
- **SuperIO ACPI**
- **JTAG master**

IN BAND: THE SIDE THAT TOUCHES THE HOST

- **P2A bridge** — PCIe to AHB — was: PCI to AHB
- **DMA**
- **MSI & MCTP**
- **LPC to AHB bridge** — registers only, not in figure
- **VGA controller** — was: 1600x1200
- **2D engine** — 64-bit
- **E2M bridge**
- not present

new in this generation · gone · changed · carried over

P-Bus

HOST AND FLASH ATTACH

- **SPI slave** — pass-through + AHB bridge
- **SPI master** — was: SPI
- **LPC master / slave**
- not present
- **PCIe 1x** — on-chip PCIe PHY — was: PCI 2.3

Diff computed slot by slot from the two datasheets. Carried-over blocks are dimmed so only the delta reads.

AST2400 / AST1250 to AST2500

19x19mm 408-pin LFBGA, datasheet V1.4 (2015) then 19x19mm 456-pin TFBGA, datasheet V1.6 (2017)

5th to 6th

2 added, 0 gone, 13 changed

CPU AND MEMORY

- **ARM1176JZS** — 800 MHz, 16K+16K cache — was: ARM926EJ-S
- **Coprocessor** — 200 MHz — was: ColdFire V1
- **FMC + SPI master** — SPI x3 — was: SMC
- **DDR3L / DDR4 ctrl** — 16-bit, 800 MHz (DDR4-1600) — was: DDR2 / DDR3 ctrl
- **SRAM 32 KB** — + battery backed
- **AHB-MBus bridge**

M-Bus

AHB Bus

ENGINES

- USB2.0 hub + 1.1 HID
- **USB2.0 host x2 / 1.1** — was: USB2.0 / 1.1 host
- 1 Gbps MAC — x2
- **SD / SDIO / eMMC** — x2 — was: SD / SDIO
- not present
- Hash & Crypto
- not present
- not present
- MIC engine
- Video engine — compression
- not present
- not present
- **SOC display** — was: Graphics CRT
- SCU — PLL x4

APB Bus

OUT OF BAND: THE SIDE THAT TOUCHES THE BOARD

- I2C / SMBus — x14
- not present
- UART — x5
- Virtual UART — x1
- **GPIO** — 228 pins — was: 216 pins
- SGPIO — x80
- PWM — x8
- Fan Tach — x16
- ADC — 16 ch
- PECI
- Timer — x8
- **WDT** — x3 — was: x2
- RTC
- MailBox
- SuperIO ACPI
- JTAG master

IN BAND: THE SIDE THAT TOUCHES THE HOST

- P2A bridge — PCIe to AHB
- **X-DMA** — was: DMA
- MSI & MCTP
- LPC to AHB bridge — registers only, not in figure
- VGA controller
- 2D engine — 64-bit
- E2M bridge
- **BMC device** — PCIe function

new in this generation · gone · changed · carried over

P-Bus

HOST AND FLASH ATTACH

- SPI slave — pass-through + AHB bridge
- **SPI1 master** — muxed — was: SPI master
- LPC master / slave
- **eSPI**
- **PCIe Gen2 1x** — PCIe-to-PCI bridge — was: PCIe 1x

Diff computed slot by slot from the two datasheets. Carried-over blocks are dimmed so only the delta reads.

AST2500 to AST2600

19x19mm 456-pin TFBGA, datasheet V1.6 (2017) then 28nm, 21x21mm 624-pin TFBGA, datasheet V1.2 (2022)

6th to 7th

5 added, 1 gone, 26 changed

CPU AND MEMORY

- **Dual-core Cortex-A7** — 1.2 GHz, L1 32K+32K, L2 256K — was: ARM1176JZS
- **Cortex-M3** — 200 MHz, 4 KB cache — was: Coprocessor
- **FMC + SPI master** — SPI x3, NAND — was: SPI x3
- **DDR4 ctrl + ECC** — 16-bit, 800 MHz (DDR4-1600) — was: DDR3L / DDR4 ctrl
- **SRAM 64 KB** — + battery backed — was: SRAM 32 KB
- **AXI + AHB-MBus bridge** — was: AHB-MBus bridge

M-Bus

AHB Bus

ENGINES

- USB2.0 hub + 1.1 HID
- **USB host / hub + EHCI** — device mode too — was: USB2.0 host x2 / 1.1
- **1 Gb MAC** — x4, DMA x2 — was: 1 Gbps MAC
- **SD card / eMMC** — x3 — was: SD / SDIO / eMMC
- not present
- **Hash / Crypto engine** — was: Hash & Crypto
- **RSA / ECC engine**
- **Secure Boot MPU + OTP** — 64 Kbit OTP, RSA to 4096
- ~~MIC engine~~
- **Video encoder** — was: Video engine
- **JPEG encoder**
- **RVAS encoder** — remote video and sensor
- **Display ctrl + DP** — was: SOC display
- **SCU** — PLL x5 — was: PLL x4

APB Bus

OUT OF BAND: THE SIDE THAT TOUCHES THE BOARD

- **I2C / SMBus** — x16 — was: x14
- **I3C** — x6
- **UART** — x13 — was: x5
- **Virtual UART** — x2 — was: x1
- **GPIO** — 244 pins — was: 228 pins
- SGPIO — x80
- **PWM** — x16 — was: x8
- Fan Tach — x16
- ADC — 16 ch
- PECI
- Timer — x8
- WDT — x3
- RTC
- MailBox
- SuperIO ACPI
- **JTAG master** — x2

IN BAND: THE SIDE THAT TOUCHES THE HOST

- **PCI-AHB bridge** — still P2A in the registers — was: P2A bridge
- X-DMA
- MSI & MCTP
- LPC to AHB bridge — registers only, not in figure
- **VGA controller + device** — was: VGA controller
- 2D engine — 64-bit
- **E2M bridge** — registers only, not in figure
- BMC device — PCIe function

new in this generation · gone · changed · carried over

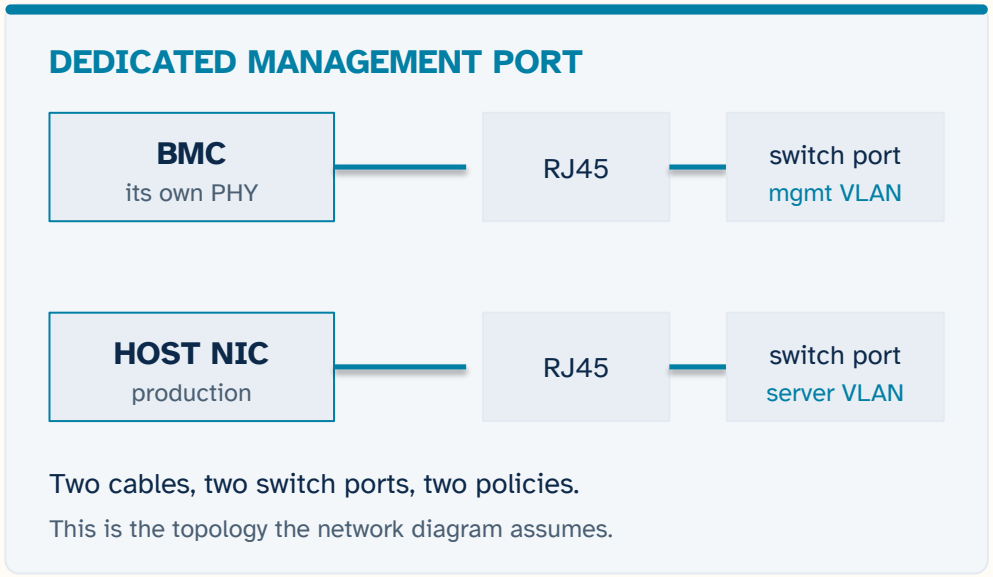
P-Bus — dropped: replaced by the sub-block fabric

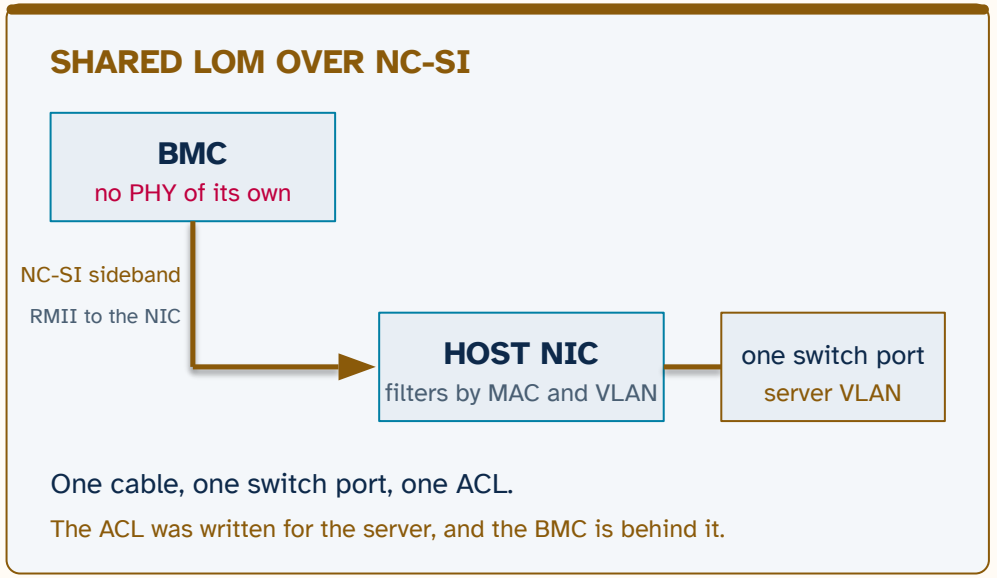
HOST AND FLASH ATTACH

- **SPI slave to AHB** — registers only, not in figure — was: SPI slave
- **SPI master** — x3 — was: SPI1 master
- **LPC bus controller** — was: LPC master / slave
- eSPI
- **PCIe Gen2** — device + root complex — was: PCIe Gen2 1x

Diff computed slot by slot from the two datasheets. Carried-over blocks are dimmed so only the delta reads.

NC-SI Sideband and Shared LOM





The controller drives the host NIC over an RMII sideband, filtered to it by MAC address and VLAN.

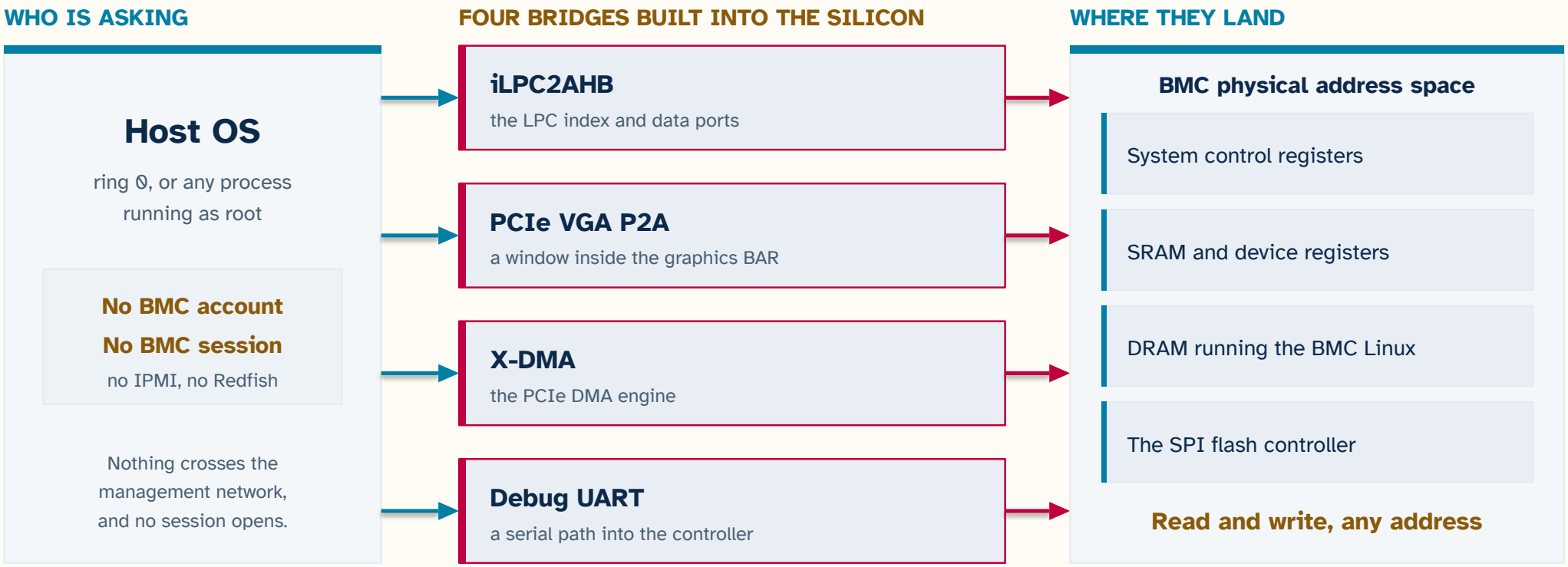
Bus Capabilities Between Host and Controller

BUS	CAPABILITIES	DIRECTION	EVIDENCE
KCS and eSPI the in-band system interface	Add a BMC account, set a password, flash sessionless channel, effective privilege Admin	→ host to BMC	OpenBMC in-band daemon session-less channels run at Admin
CHIF loopback the HPE host mailbox	Full Redfish admin with no session token the reply names it: AutoLogin	→ host to BMC	CHID is specific to HPE iLO HPE ships it as a feature
AHB bridges iLPC2AHB, P2A, X-DMA, UART	All of BMC physical memory, read and write	→ host to BMC	AST2400, AST2500, AST2600 public variants, new methods here
USB and UEFI volumes keyboard, boot media, pre-OS disks	Type at the console, boot virtual media every one of these is a product feature	← BMC to host	Dell and Supermicro vendor surface documents
PCIe and DMA Varies by part and difficulty	Read and write host memory, no agent, no login	← BMC to host	iLO4 proven, ASPEED WIP Airbus Seclab, X-DMA/DMA here

Three lanes up, two lanes down, and neither endpoint authenticates the other.

Five buses between the host and the controller, and the direction each capability travels.

Host-to-BMC: ASPEED AHB Bridges

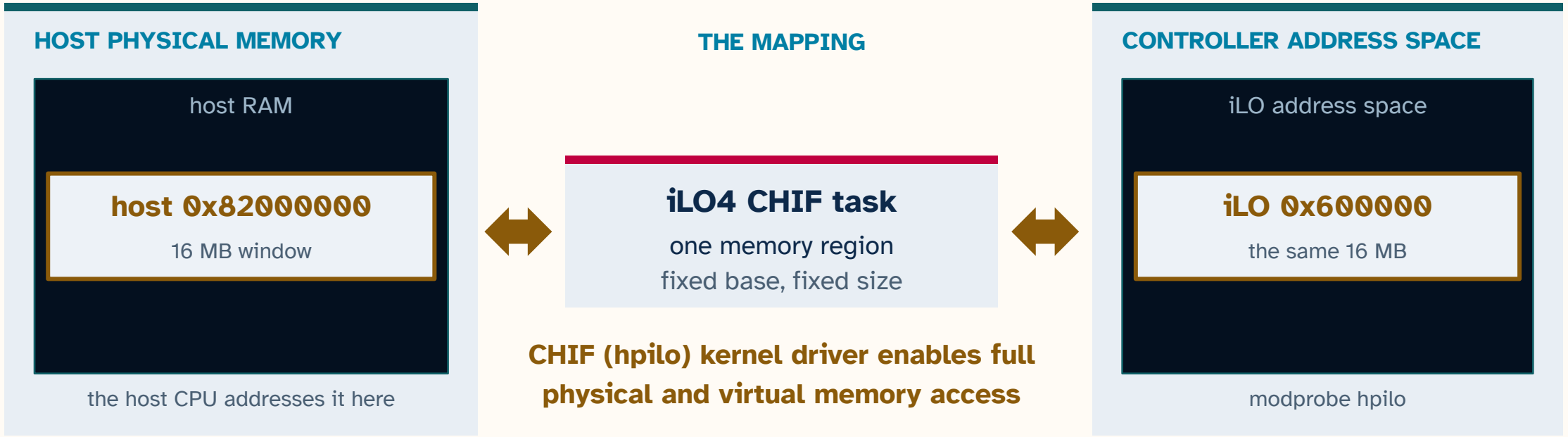


AST2400, AST2500, AST2600 ship the bridges enabled. Firmware can switch each one off. BMC security bits can be CLEARED later

Pantsdown, CVE-2019-6260, public since January 2019. Affected parts shipped for years after.

Four bridges into the controller address space: iLPC2AHB, PCIe VGA P2A, X-DMA, and the debug UART.

BMC-to-Host: Host Memory Access (HPE iLO)



iLO4, 2019	NotPetya keys read out of a locked host
Tooling	packaged as a DMA backend for PCIleech (uses CHIF driver for host agent)

ASPEED DMA Attacks

ASPEED AST2400 / AST2500

- BMC can force itself to be bus master (BME)
- DMA can reach under 4GiB of physical
- IOMMU and VT-d still apply
- IOMMU disable via AHCI write in POST
- IOMMU/VT-d disabled via BIOS patch

ASPEED AST2600

- Difficult to BME (fake NVMe/Firewire card)
- eSPI give 64-bit address writes (no READ!)
- AMD/H13 makes racing POST for AHCI hard

Patching the SMC BIOS config

1. In-memory replace of Public Key
2. Backup and rewrite the /nv config file
3. Force host boot w/o IOMMU & VT-d
4. Force host to boot under <4GiB memory

Easy mode remote DRAM writes

1. Hypervisors with IOMMU groups enabled
2. eSPI write avoids need for BME

Stuck POST recovery is fun...

Bug Beastiary

Hello! This session has been temporarily omitted while the vendors catch up on disclosure.

OOBscan & Open Source Software

Open Source

OOBscan

Auth bypass, memory read, persistent payloads, host DMA, and offline password cracking.

v1 Live!

Nuclei

An IPMI protocol for the nuclei JavaScript bindings, and templates that speak it.

PR Soon!

Nmap

NSE scripts for the pre-authentication IPMI checks.

PR soon!

John the Ripper

RAKP MD5 and SHA256 formats, on CPU and GPU.

https://github.com/openwall/john/pull/6005

OOBscan Coverage and Build

116
registered checks

95
products fingerprinted

61
vendors

0
destructive checks
(on by default)

One static Go binary, no runtime dependencies. 12 top-level commands, 63 default ports: 9 UDP and 54 TCP.

Scan Pipeline for One Target

One target

63 ports
4s per probe
4m budget

1 Identify

18 protocol collectors
148 fingerprint rules
System GUID taxonomy
TLS and SSH key match

class, vendor, product

2 Scoped checks

only the checks whose applicability names this vendor or device class

no vendor probe fires at the wrong device

3 Cross-vendor

RAKP hash capture
cipher-suite walk
TLS, SSH, SNMP, SSDP
known-CVE matching

116 checks registered

4 Credentials

crack the leaked hash offline first, 97 rows
capped online fallback
only where nothing leaks

no lockout from a crack

Decoy verdict short-circuits the whole battery
a honeypot runs the identity reporter and nothing else

One NDJSON record per host
identity, findings with proof, services, ports, RAKP material

Off by default, because they cost time or touch an account

Full IPMI walk
every NetFn and command, OEM reachability, IPMB satellite discovery

Measure null privilege
an advertised null account may be clamped to nothing, so open a session and ask

OpenBMC key sweep
serial heap-pointer search, 5 to 15 candidates a second, an hour or more per host

Native Protocol Implementations and Ports

Protocol	Port	Use in the scan
RMCP and IPMI 1.5	623/udp	presence ping, auth caps, username oracle
IPMI 2.0 RMCP+	623/udp	RAKP-2 hash per username at one suite
Redfish and HTTP	443, 9080	fingerprints, OEM actions, auth bypasses
SNMP, SSDP, SLP	161, 1900, 427	sysDescr, libupnp CVEs, service replies
mDNS, WS-Discovery	5353, 3702	device advertisements on the segment
Lantronix, Moxa, Digi	30718, 4800, 2362	vendor setup and discovery protocols
Intel AMT, WS-Man	16992, 16993	identify plus CVE-2017-5689

Identity from One Pre-Session Field

Get System GUID, sessionless, sixteen bytes

Supermicro smc-board-serial

board code 4101MS names the X11 generation

`34 31 30 31 4d 53` `3c ec ef 00 11 22` `00 00 00 00`

printable model prefix | embedded NIC MAC | zero trailer

Dell dell-service-tag

service tag 7X2K9Q1, one marker bit stripped

`44 45 4c 4c` `58 10 4d 32 92 4b b7 c3 4f 39 51 31`

the ASCII marker DELL | seven tag characters at scattered offsets

HPE hpe-printable

part P53933, plant CZJ, serial 7A00042

`50 35 33 39 33 33` `43 5a 4a` `37 41 30 30 30 34 32`

part id | plant | unit serial

Corroborated before the verdict is published

148 signature rules, the IANA manufacturer id, 24 shared-key fingerprints

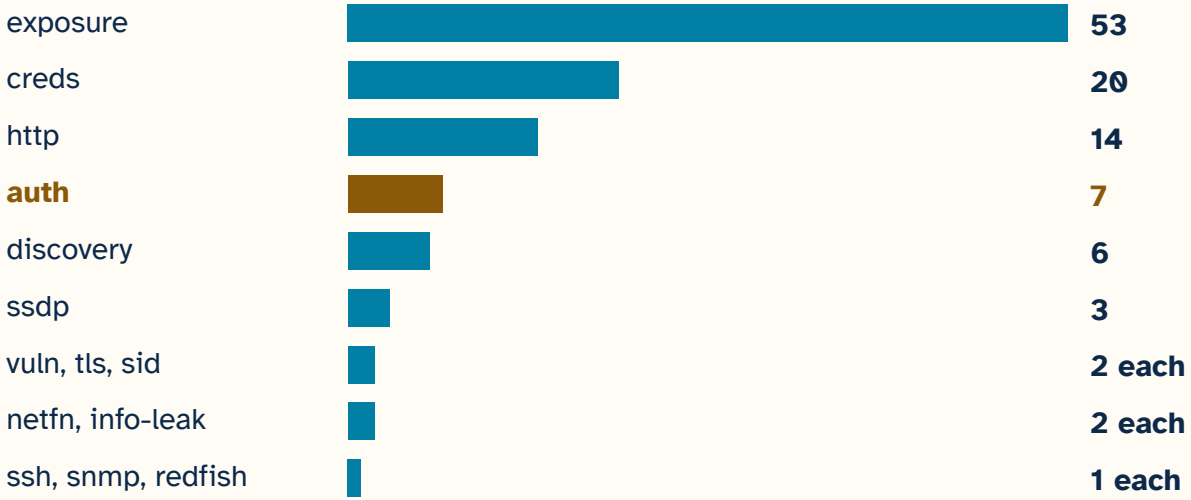
An all-zero GUID names no unit

it carries no vendor bits, so no vendor is claimed

Check Registry by Category

116 checks, 14 categories

each one declares the vendors and device classes it applies to



Six are off by default and three need a credential

Nothing in the registry is marked destructive, and the flag that would enable such a check has no check bound to it.

A check that never fires across a whole corpus is reported as dark by the quality audit.

The seven in the auth category

Payload-type confusion
HPE-0005, an auth-algorithm record typed 0x02

Stale default user key
OBMC-0005, bounded patch-status handshake

Sessionless relabel
OBMC-0004, RAKP 1 retargets a live session

Unordered RAKP
SMC-0014, message 3 with no message 1

Record-0 key forge
SMC-0014 follow-on, one candidate per run

Reserved role nibble
SMC-0015, low privilege to Administrator

No RAKP rate limit
CVE-2013-4786, still unthrottled in 2026

The IPMI Subcommand

`mc  chassis  lan  fru`

the standard ipmitool command surface

inventory and configuration

`user  channel`

account list, enable, disable, privilege

the whole account table

`sol  isol`

Serial-over-LAN on 2.0 and on 1.5

the host console

`i2c  spd  mem  xdp`

platform buses, DIMM contents, memory

controller and host silicon

`smcbackup`
`smcrestore-shell`

OEM config export and the restore path

persistence across a reboot

`smc0014  smc0015`

the two RAKP findings, with controls

negative controls per run

Semi-automatic exploitation of auth bypass, integrity/encryption omission, and priv escalation

Reading BMC Memory over IPMI

mem subcommand, Supermicro X11 profile

```
$ oobscan ipmi -u ADMIN -P **** 192.0.2.10 mem info
profile smc-x11   Supermicro X11, ATEN, AST2500   confidence live
read  0x30/0x70 subfn 0xC0    write 0x30/0x70 subfn 0xC1
spaces  physical virtual flash scu gpio i2c spd fru sdr mii peci
        cpu nvram
hazard  unrestricted physical write is BMC code execution

$ oobscan ipmi ... mem read 0x21FC0000 2k   # the U-Boot environment
reply order MSB-first on this generation, reversed to rebuild the image
```

Offline Cracking Without Authentication

1 On the wire

one RAKP message 1 per
candidate account name
the reply carries the salt
and the keyed HMAC

message 3 is never sent

2 In the record

account, algorithm number,
salt bytes, HMAC bytes
one row per account that
answered

SHA1, MD5 and SHA256

3 Crack in place

every core on the scanning
host, all three algorithms
recovered values land in a
potfile under the cache

no wordlist leaves the host

4 Hand it to a GPU

export as two hashcat modes
or three John formats
fold the finished potfile
back into the corpus

the GPU never sees a target

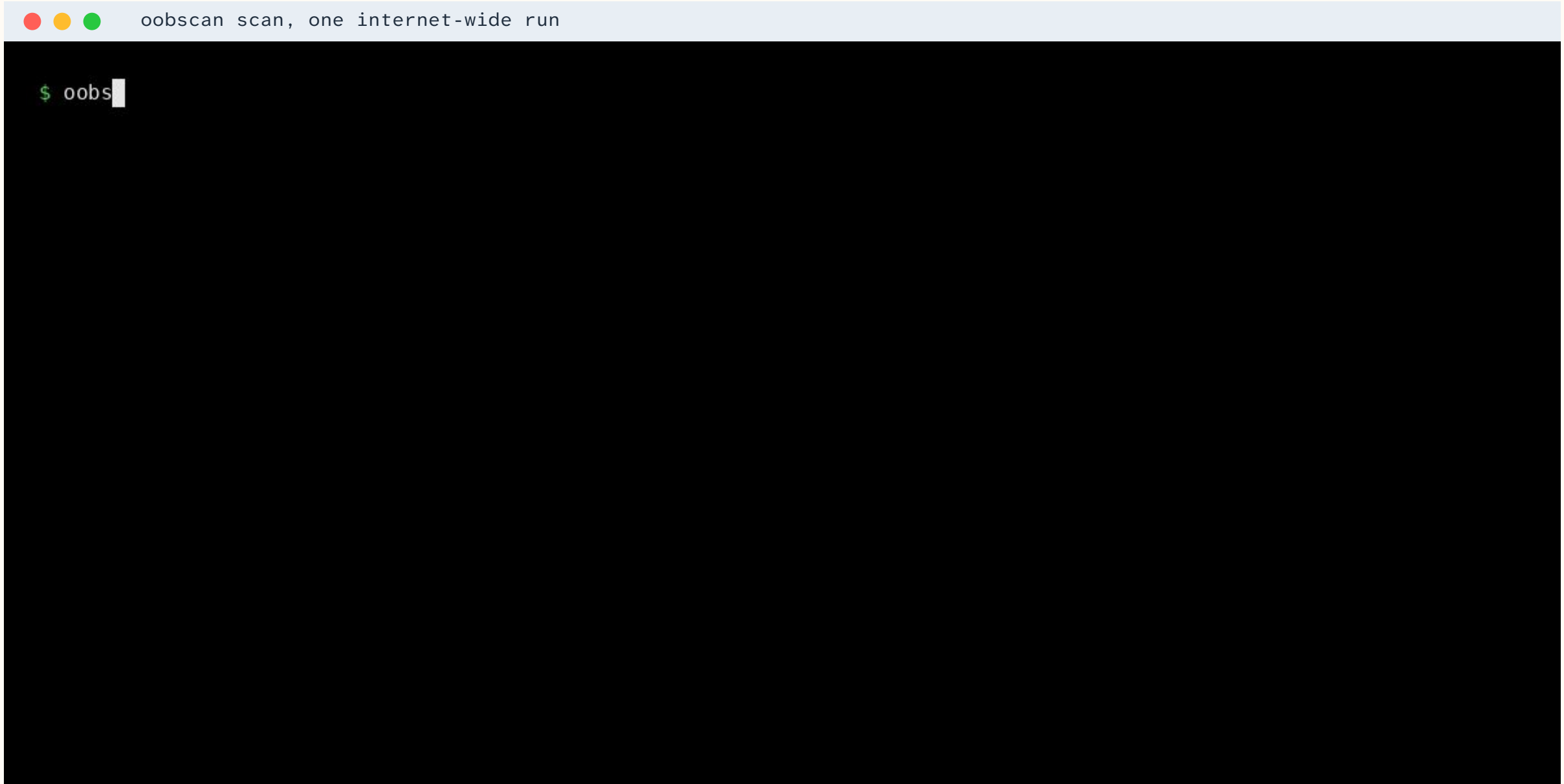
potfile import, one shared store

The crack path authenticates nothing
no session, no failed login, no account lockout

Where a hash never leaked
six online guesses at most, and an accepted session is closed at once

The BMC hands over the hash before it has seen any proof, so the guessing happens where no account can lock.

OOBscan at Scale: One Internet-Wide Pass



Safety Properties of the Scanner

Read First

- Probes read without writing
- Cracking runs offline
- Writes stage until apply
- No destructive check exists

Scoped

- Checks declare their scope
- Generation gating
- A decoy cancels the battery
- Panic isolation per host

Opt Out

- One flag disables logins
- Aggressive lanes stay off
- Deadlines at both levels
- Undecidable runs report inconclusive

Common Sense Defense

	Operator action	Reason
1	Disable IPMI and RMCP+ where the platform allows it	Stick with Redfish only, try to IP ACL that too
2	Disable the in-band KCS host interface channel. Blocklist ipmi/ilo/chif modules	No auth - host to BMC reflash/reconfigure
3	Set a unique username AND password on every device	Avoid RAKP leak with random username, make GPUs work!
4	Use Redfish only, with IPMI off	Redfish-only avoids the majority of attack surface
5	Give each device its own VLAN where the topology allows it	Pre-authentication defects stay reachable from any host on the same segment. Keep BMC compromises from chaining
6	Use a dedicated management NIC; avoid shared LOM and NC-SI sideband	A sideband path puts the controller on the host switch port and its access list. Avoid surprise public IPMI

VulnDev

Research Setup and the Repeating Loop

Set up once

Pull every firmware
40 vendors, 282 trees

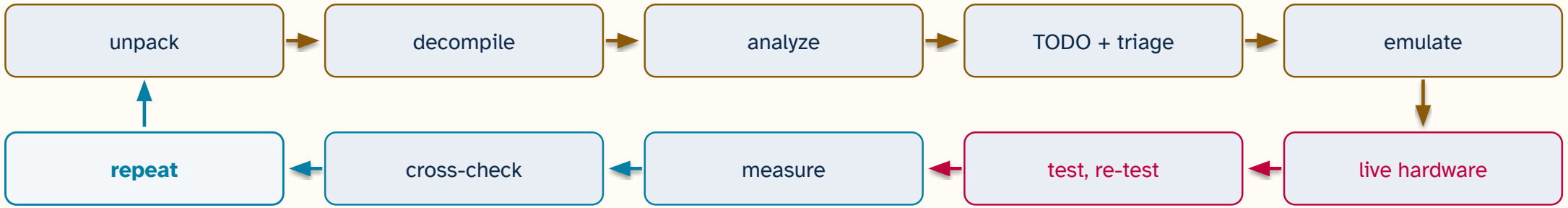
→

Install the toolchain
unpackers, Ghidra, QEMU

→

A repeatable test harness
so the passes run unattended

Then repeat, for weeks



What each pass does

Each pass records a lead and attempts to refute it. Most leads are refuted.

Leads that survive are emulated, fired on real hardware, then measured against the internet.

Internal and external measurement runs in parallel so a lab finding gets a field number the same week.

Model Token Spend Over One Research Month

38 billion

model tokens, one research month

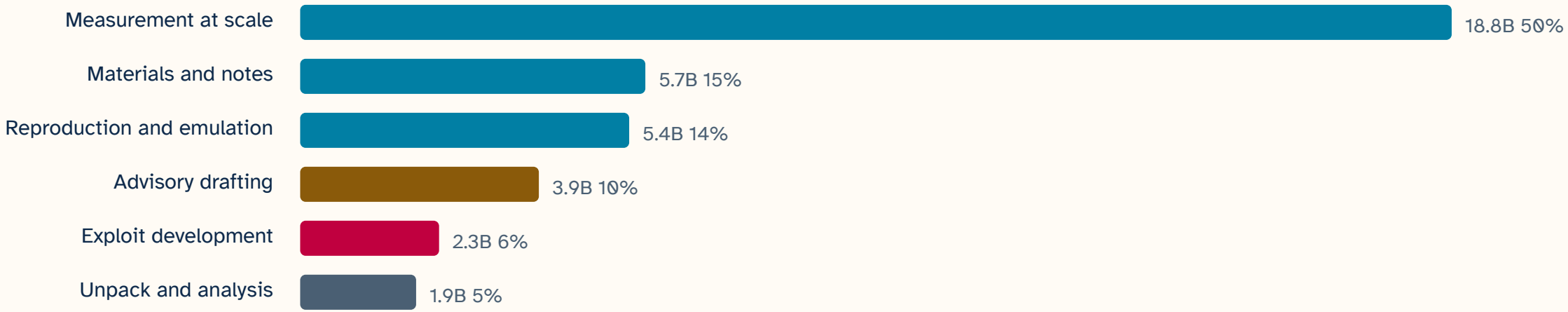
206 sessions

1 Jul to 2 Aug

161M generated

the rest is cached context re-read

Where the tokens went, by stage



Measurement is the largest stage

Exploit development was six percent of the total. Measurement at scale was fifty percent.

Stage split is a per-session classification. The 38 billion total and 161 million generated are measured.

Unpack, Decompile, Emulate, Prove

01 UNPACK

282

firmware versions
across 40 vendors

374 payload files, 61 GB

168 open to a full rootfs

33 of 40 vendors give one

Stock tools, no keys

02 DECOMPILE

1,897

decompiled C modules
in 83 output trees

Headless, and in parallel

One real-time-OS image needed its own loader

Several models, one file

Agreement filters noise

03 EMULATE

6

shipped images boot
on one AST2600 board

Four vendors, one model

69 trees run their own binaries under QEMU

11 boot a real kernel

Two blind bypasses here

04 PROVE

5

boards on metal,
two vendors

15 execution proofs

Live code execution confirmed on a device

Read-only probes only

Anything unproven says so

OUTCOME

123 confirmed findings, 35 drafted advisories across 8 vendors

396 experiment write-ups behind them

Firmware Images That Do Not Extract

The images that yield no Linux root filesystem fall into three classes.

No Linux Userland

- RTOS and bare-metal MCU
- APC, Eaton, Tripp Lite
- ServerTech, Moxa, xPico
- Nothing to chroot into

Signed Monolith

- HPE iLO4, iLO5, iLO6
- One ELF, no root filesystem
- 626 program headers
- Read the image itself

Sealed On The Part

- Huawei legacy iBMC
- RSA-wrapped AES payload
- Key on the secure element
- No offline path at all

Two of the three classes are read directly from the image.

Four Layers to a Readable Configuration

supermicro x10, firmware 3.53

```
$ file config.bin
config.bin: data

layer 1  3DES, key from the web binary
layer 2  CRC32 header, computed then discarded
layer 3  openssl salted, password in the same binary
layer 4  tar

$ ./unwrap config.bin | tar t | head -3
etc/passwd
etc/shadow
nv/PSBlock
```

> **Four layers, and every key needed to walk them ships in the image that reads them.**

Rebuilding the Memory Map for Decompilation

AS DELIVERED

One 24 MB ARM image

Green Hills INTEGRITY, no filesystem

program headers	626
claim zero bytes on disk	310
PROGBITS sections	316

The section map carries a true address.

STOCK ELF LOADER

Segment headers taken as written

Maps every segment uninitialized

About 672 bogus functions

Zero string cross-references

The flow repair never converges

Hundreds of thousands of warnings

RAW IMPORT, THEN REBUILD

Rebuild from the sections

Raw import, analysis held off

One initialized block per section

Placed at its true address

Analysis converges, zero repairs

Decompiled one module at a time

The section table carries the true addresses.

THE DECOMPILER

`RECYCLE_EVERY = 1500`

Cached state exhausts the heap across tens of thousands of functions.

THE HEAP

`GHS_HEAP = 48g`

The older build died in constant propagation at 24 gigabytes.

Load Parameters and Analyzer Run Outcomes

ilo4 elf.bin, load parameters

```
program headers      626
headers with filesz 0  626   # nothing to map from the file

# a stock ELF load produces an empty program, so the loader
# is section aware and lays segments down by hand
RECYCLE_EVERY = 1500         # analyser restarts, or it dies
peak resident   48 GB        # for one 2.30 image

runs 24    completed 24    OutOfMemory during 14
```

Fourteen runs threw OutOfMemory and still reported success, so the run log is not the check.

Address Translation Between Decompile and Runtime

ELF virtual addresses what the decompile prints		SSH task runtime map what the CPU actually addresses	
.ssh.elf.text base	0x0390d000	.ssh.elf.text base	0x00010000
rdpkt epilogue	0x0391b1a0	rdpkt epilogue	0x0001e1a0
self-loop gadget	0x0390d054	self-loop gadget	0x00010054

```
runtime = elf_vaddr - 0x38FD000
```

INTEGRITY maps each task's modules into a per-task address space

FIRED AT THE ELF ADDRESS

PC = 0x0390d054

Unmapped in this task. Contained fault,
no wire, no reset, no log line.

The unmapped page yields no observable.

SAME PAYLOAD, TRANSLATED

PC = 0x00010054

Mapped. The gadget runs, and burns CPU.
A positive, repeatable observable.

Directed pre-auth execution, confirmed live.

Eight Models over One Module

Twelve tracked entries in one iLO4 IPMI module, each re-checked line by line against the decompile

		opus48	sonnet5	gpt55	deepseek	qwen37	sakana	sol56	nemo3	found by
P1	Integrity never enforced per packet	●	●	●	●	●	●	●	●	**8/8**
P2	No inbound replay window	●	●	●	●	●	●	●	■	**7/8**
P3	Decrypt runs before any MAC check	○	◎	○	◎	●	●	●	○	**5/8**
P4	RAKP2 hands back a password verifier	●	●	●	●	●	●	●	●	**8/8**
P5	Account enumeration via status codes	●	●	●	●	●	●	●	●	**8/8**
P6	Session keys dumped at log level 4	●	●	●	●	●	●	●	●	**8/8**
P7	Non-constant-time MAC compare	●	●	●	●	◎	●	●	◎	**8/8**
P8	Session ID and nonce RNG unverified	●	●	●	●	●	●	●	●	**8/8**
P9	Short payload wraps a copy length	○	○	○	●	○	◎	●	○	**3/8**
P10	Session never bound to a peer address	○	○	○	○	○	○	●	○	**1/8**
P11	Integrity-none suite still negotiable	●	●	●	●	●	○	●	●	**7/8**
P12	Secondary set of 5, session-kill shown	○	○	○	○	○	○	●	○	**1/8**

● reported ○ missed ◎ reported, but overstated ■ asserted a check that is absent

Three Tiers of the Lab Environment

Simulators	Protocol-accurate IPMI servers, cheapest to run and the first place a check is regression tested
Shims and qemu-user	The real vendor web daemon out of an extracted rootfs, for sealed devices that will not boot
Full firmware VMs	Real vendor images booted end to end, serving live Redfish, IPMI and SSH

Thirty-three environments, every service on its own port, so any combination runs at once.

Emulation Results by Firmware Stack

Stack	Under emulation	What it proved
OpenBMC Phosphor	Boots, full userland	Session handling, live
AMI MegaRAC	Boots on Debian builds	Command dispatch and OEM routing
ATEN	Userland only, under a shim	Web login and config paths
Huawei openUBMC	Boots, heavy host	Declarative command tables
Green Hills INTEGRITY	Does not boot	Static work and metal only

A stack that will not boot is not a dead end, it moves the work onto real hardware.

Running the Shipped Web Daemon under qemu-user

qemu-user against an extracted rootfs

```
$ qemu-arm -L ./rootfs ./rootfs/usr/sbin/httpd -f httpd.conf

# the vendor's own binary, its own configuration parser,
# its own authentication path, on a host that is not the device
$ curl -sk https://127.0.0.1:19080/cgi/login.cgi \
    -d 'name=ADMIN&pwd=ADMIN' -D-
HTTP/1.1 200 OK
Set-Cookie: SID=hK3m9QpZ2rLd8Xv
```

Running the shipped binary answers questions a rewritten emulator would only guess at.

Detection Logic under Regression

replay, one corpus, two builds

```
$ oobscan replay corpus/ --against previous

check                       was     now   delta
ipmi.rakp.hashDisclosure   23,288 23,288       0
ipmi.cipherZero               697    697       0
ipmi.defaultCreds           1,817  1,904     +87
snmp.defaultCreds          11,761  2,614   -9147   <-- regression

1 check changed beyond tolerance
```

Replaying a stored corpus catches a detection regression that no unit test was going to find.

Four Ledgers from Lead to Advisory

LEDGER 1

Raw follow-ups

620

leads, one per tracking id

Every follow-up raised during the per-vendor firmware audits.

LEDGER 2

Removed pre-triage

210

answered classes

Already promoted	53
Default credentials	64
Shared TLS, SSH keys	48
Known-CVE confirms	45

LEDGER 3

Tracked, triaged

313

items, evidence attached

Unblocked this pass	40
Still blocked	273
Critical and High	43
Medium, Low, Info	270

LEDGER 4

Confirmed findings

123

rows, risk-rated per vendor

Critical and High	23
Medium, Low, Info	100

OUTCOME

35 drafted advisories

8 vendors, 35 CVE IDs reserved

FOUR INDEPENDENT COUNTS

620 and 210 were counted in the 2026-07-11 pass, 313 in the 2026-07-13 recalculation. The columns carry different dates and do not subtract.

Refutation is the other exit from triage. The vendor lead ledgers hold 396 experiment write-ups.

Negative Results: Fuzzing and Bundled Components

Pre-auth fuzzing produced no crashes, and reading native code produced no bundled-component findings.

396

failed experiment write-ups behind the confirmed findings

4

hand-written fuzz batteries, zero crashes

0

n-days in bundled components found by native code review

9

candidates an outside toolchain found, 7 of that class

Wild Goose Chases

Serial console	the session token rides in a cookie, no URL leak
SMTP injection	an escape filter neutralizes it at the sink
Web session IDs	the generator reads the ASPEED hardware RNG
Chunked overflow	the guard only wraps to a negative read length
Shared key file	the file only holds a challenge dictionary

Method Notes

Read The Image

- Decompile the whole image
- The relocations are done
- Eight files per vendor
- One schema, one rulebook

Triage At Scale

- Several models, one file
- Count the agreement
- Rate severity by hand
- Fix the rulebook, rerun

Prove It Live

- Emulate early
- Emulate the stock image
- Translate the address first
- Confirm the base address

Questions

x [at] hdm.io