Mind the Gap

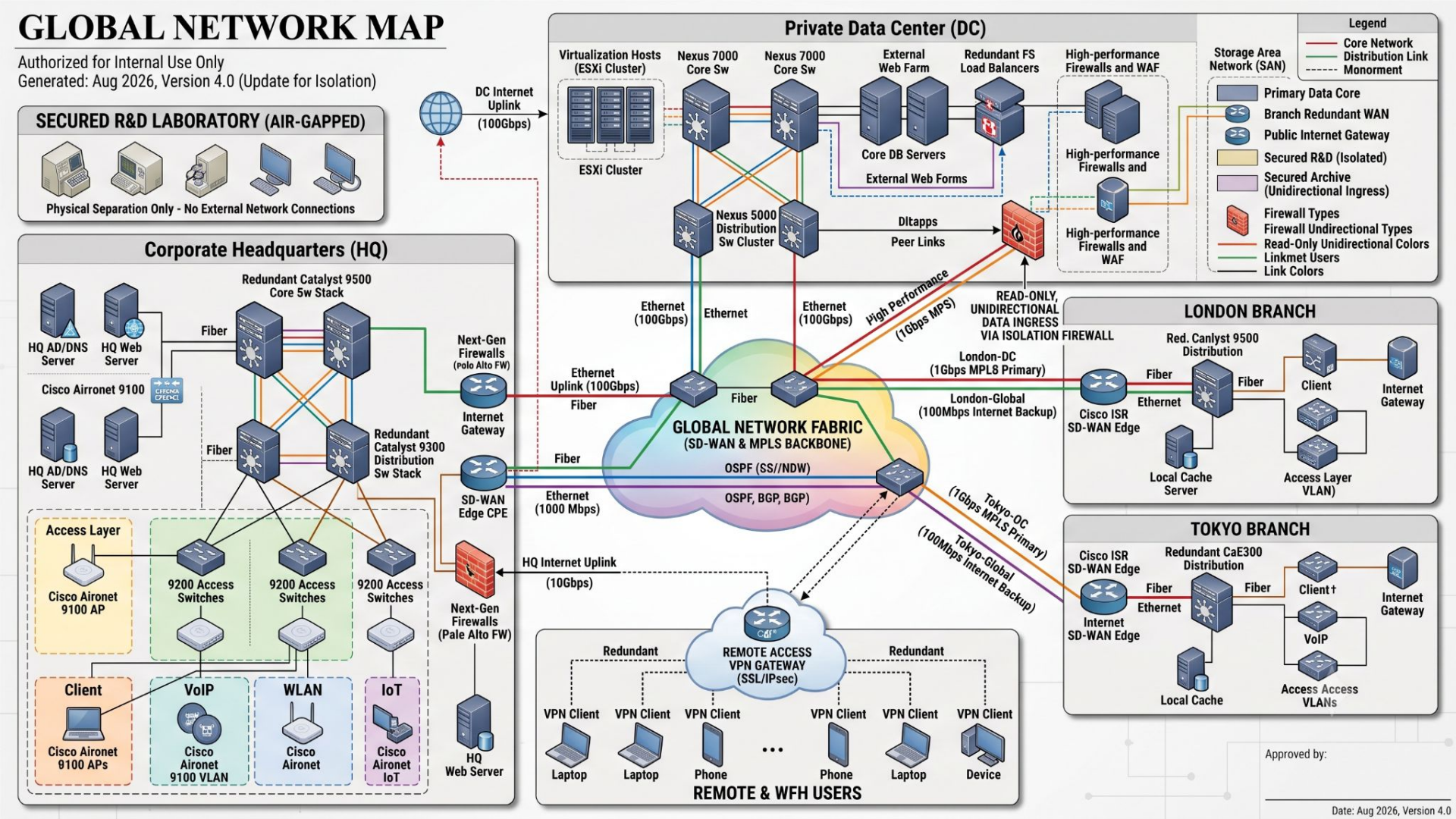
Bridges, Backplanes, and BloodHound

BSIDES LAS VEGAS 2026

HD Moore

GLOBAL NETWORK MAP

Authorized for Internal Use Only
Generated: Aug 2026, Version 4.0 (Update for Isolation)

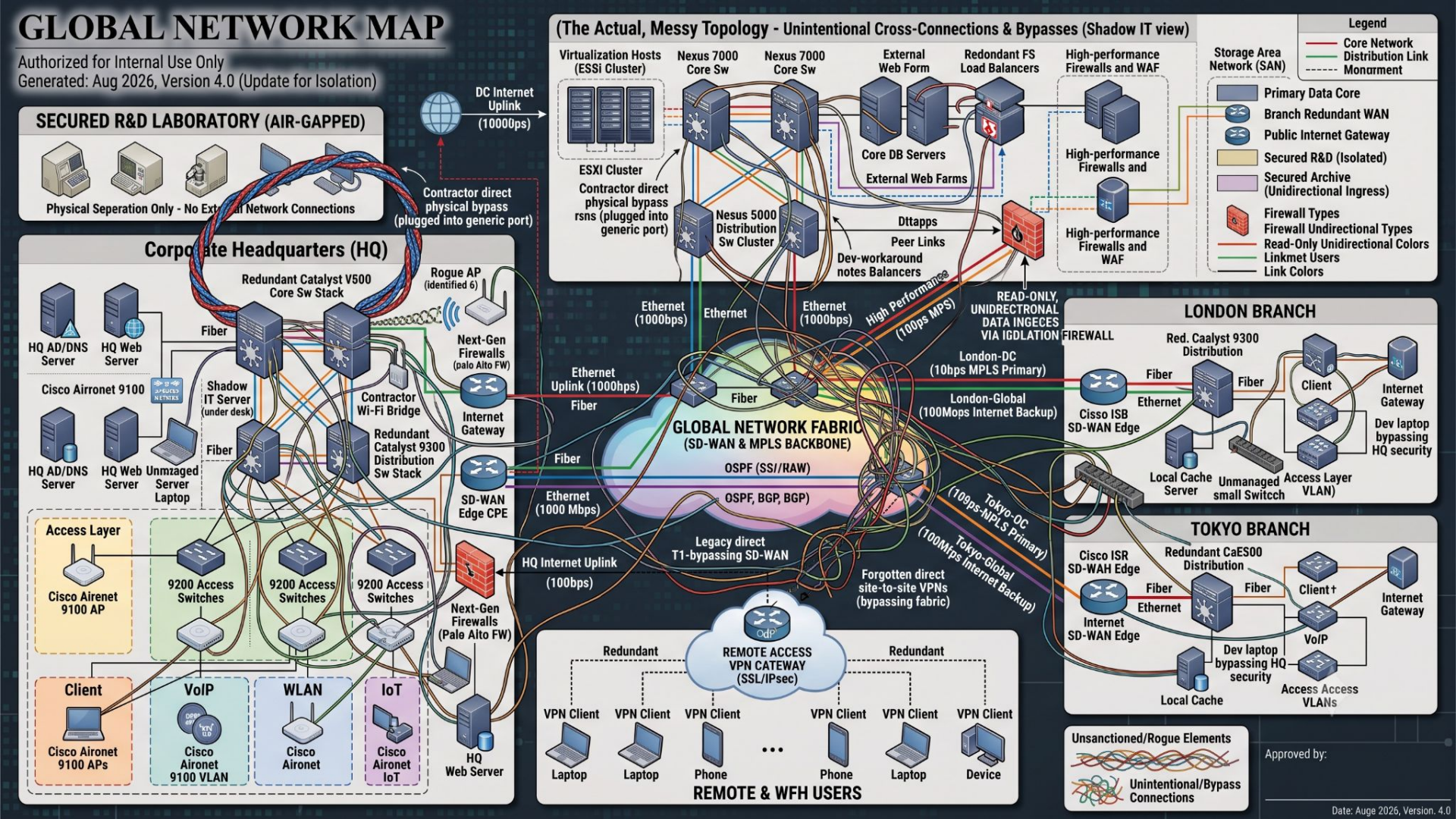


Approved by:

Date: Aug 2026, Version 4.0

GLOBAL NETWORK MAP

Authorized for Internal Use Only
Generated: Aug 2026, Version 4.0 (Update for Isolation)

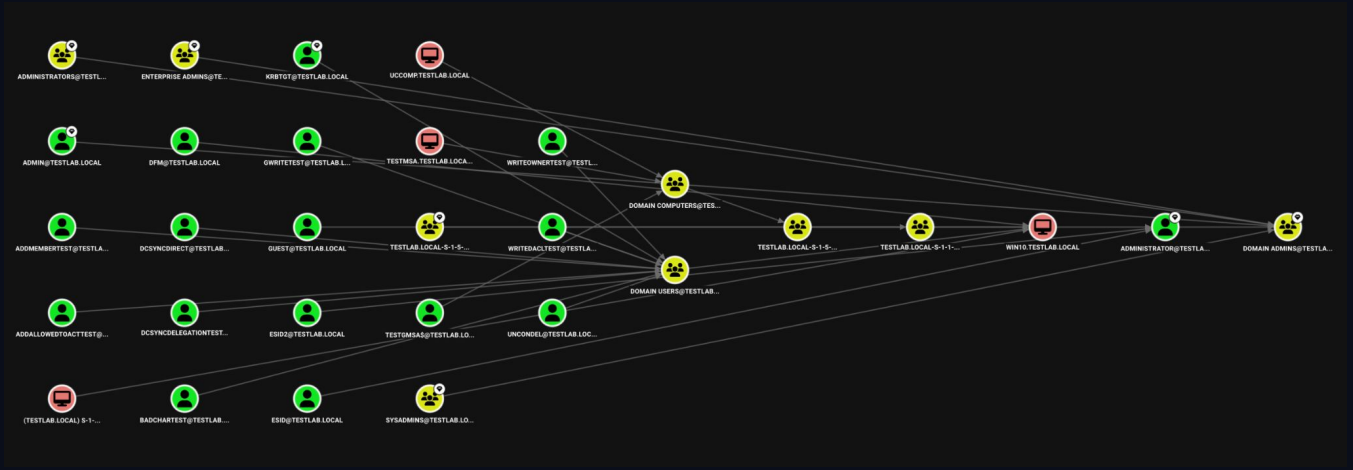


Mistakes are often subtle and dangerous

- Network maps are the intended state, almost never reality
- Even when generated from device configuration
- A crossed wire makes your intranet public
- Overlapping VLANs can live for years
- It’s worse when nobody notices

A similar challenge happens with identity

- BloodHound & OpenGraph continue to prove this point
- Mistakes compound with size & complexity



SEGMENTATION

The last control

- Can’t patch it?
- Can’t install an agent?
- Can’t reconfigure it?
- Segment it:
  - End-of-Life
  - Third-party
  - Machinery

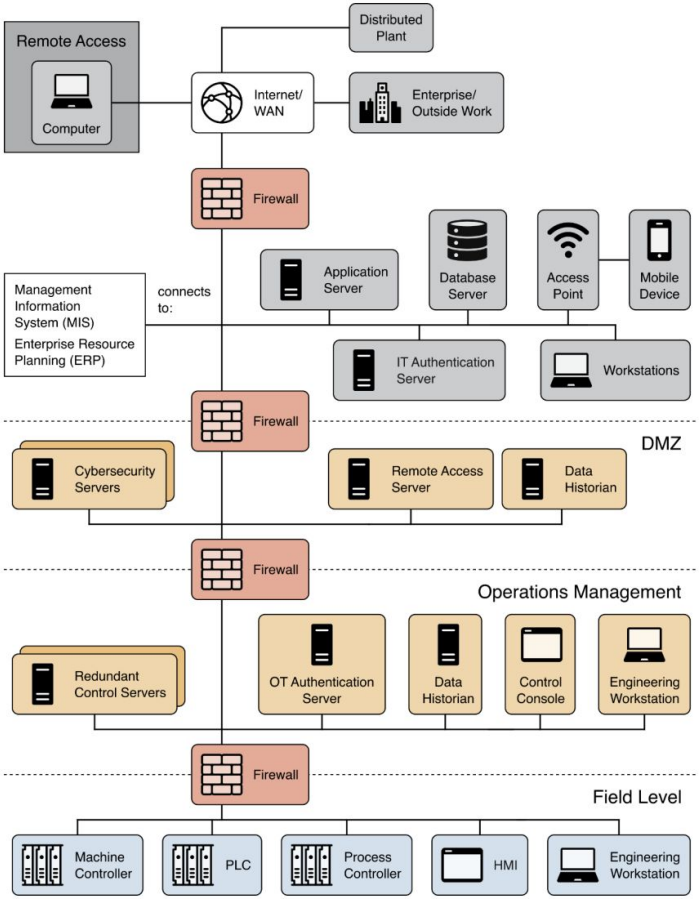
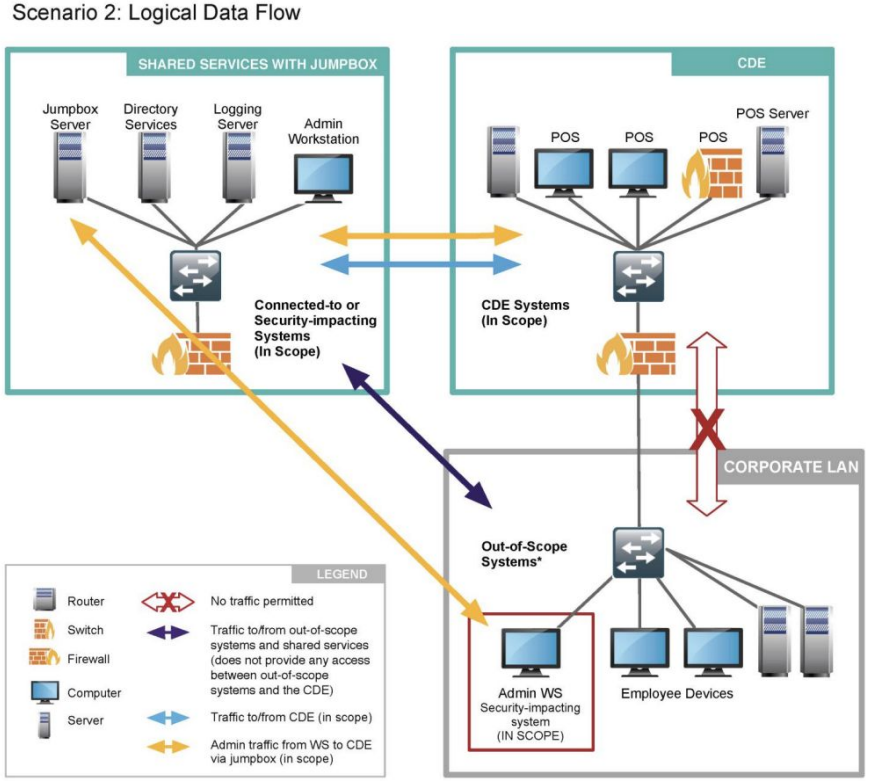


Fig. 18. A defense-in-depth security architecture example for a DCS system

Logical separation

- Security Impacting Systems
- Goes beyond cables & ACLs
- Can an attacker influence it?
- Can they talk to each other?

FIGURE 5 – Logical Data Flow: Administration of CDE Systems from a Security-Impacting System in the Corporate LAN



** Only if verified these systems meet all criteria for being out of scope, including there being no connectivity between these systems and the CDE. Controls must also be in place to prevent out-of-scope systems gaining access to the CDE via systems in the Shared Services network.*

Purdue Model for ICS Security: Layers and Zones of Defense

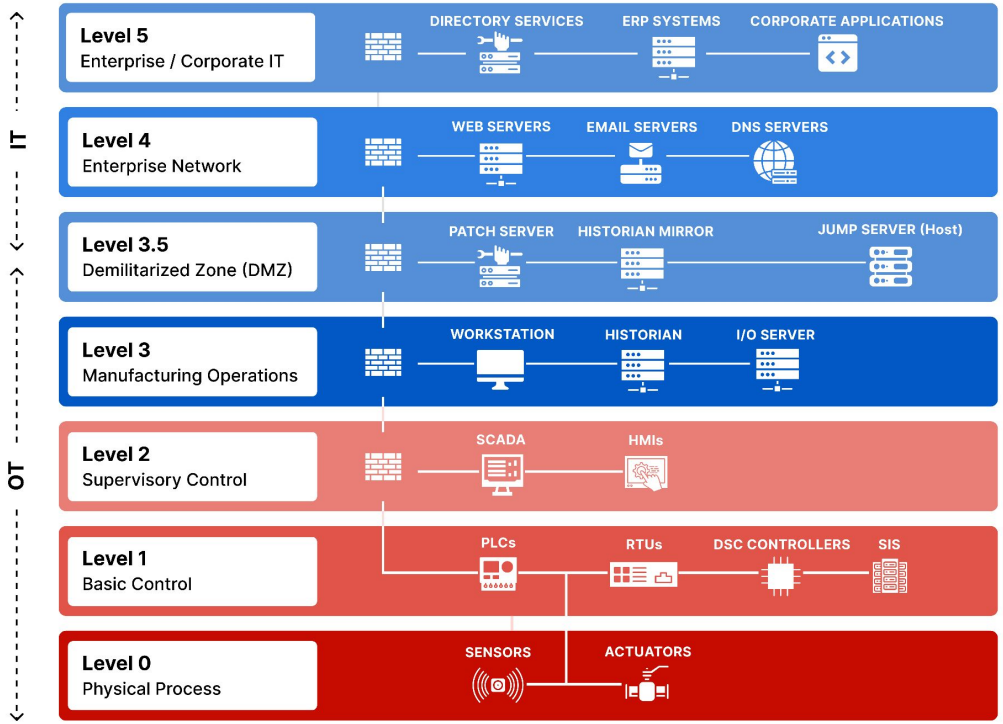


Diagram from Fortinet.com

"Firewalls are commonly used to support network isolation and employed as boundary protection devices to control connections and information flows between network segments."

*"Firewalls are very flexible isolation devices and typically constitute the **primary** mechanism for protecting OT devices"*

NIST 800-82 Rev 3

Purdue Model for ICS Security: Layers and Zones of Defense

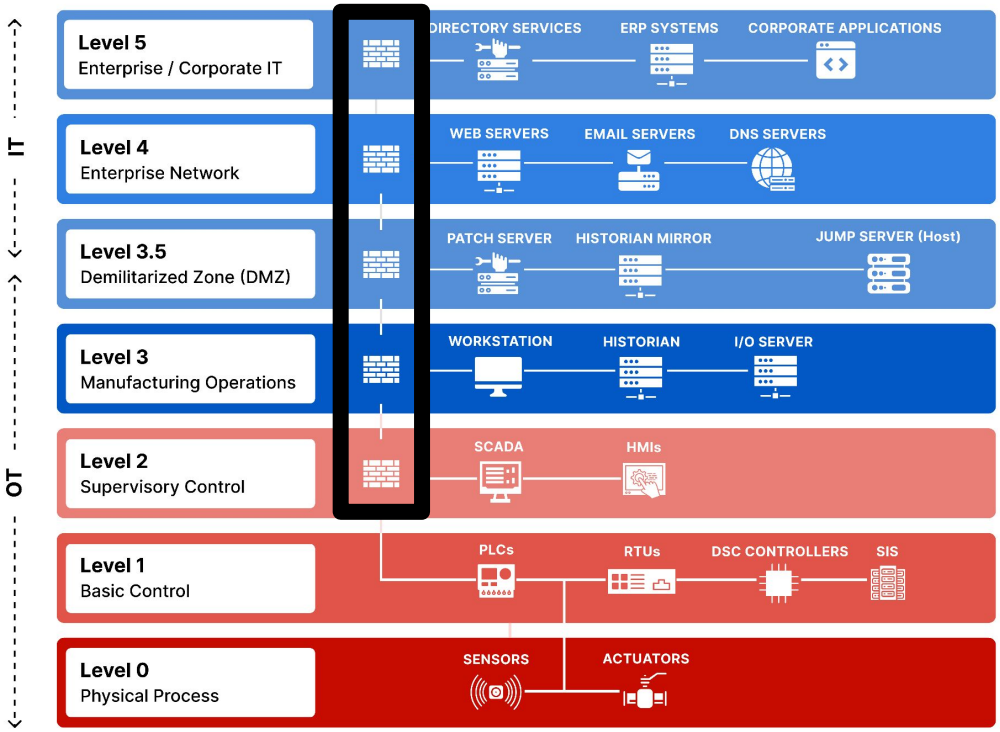


Diagram from Fortinet.com

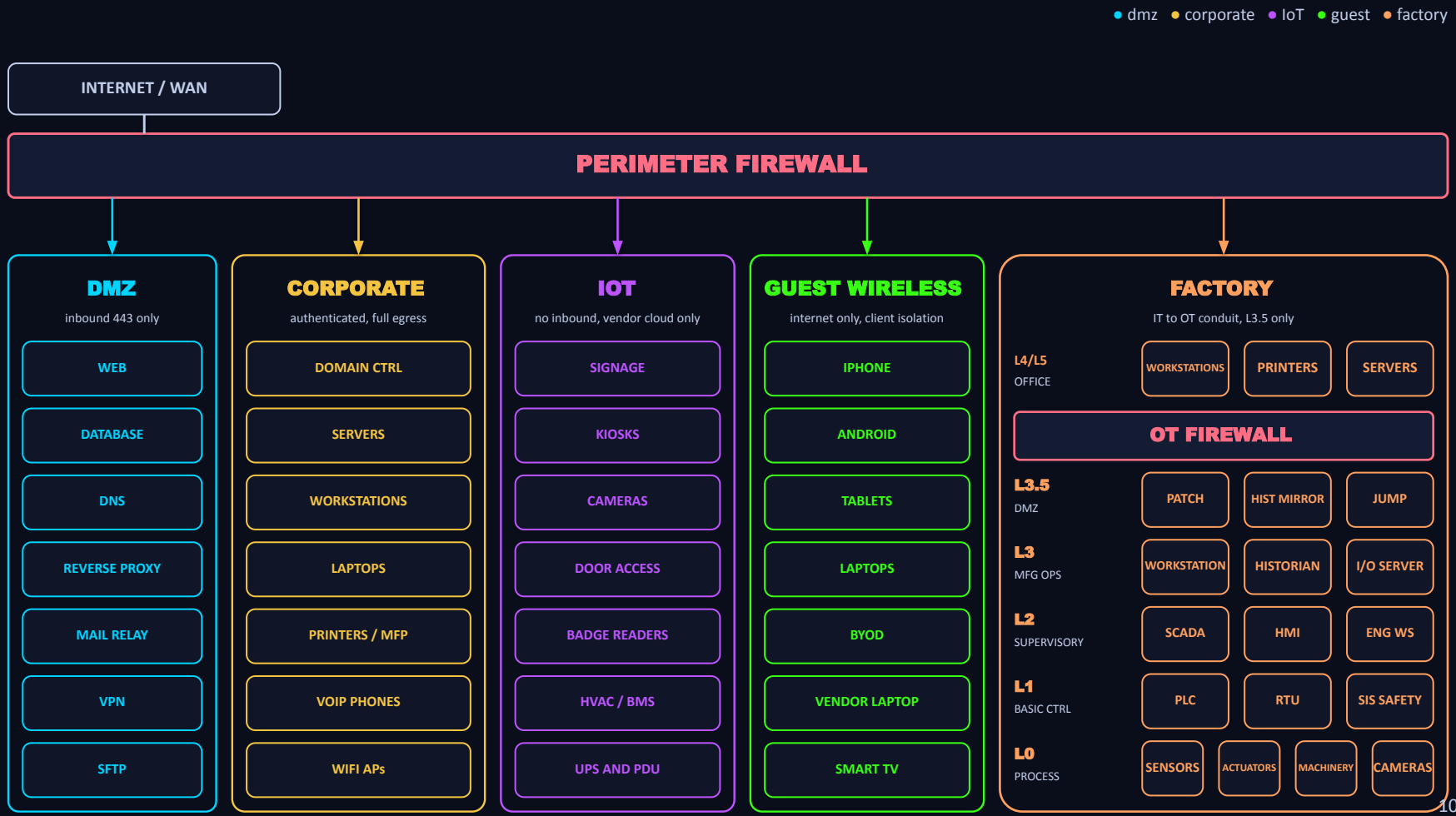
ALERT

Fortinet Releases Guidance to Address Ongoing Exploitation of Authentication Bypass Vulnerability CVE-2026-24858

Release Date: January 28, 2026

Newly disclosed vulnerability Common Vulnerabilities and Exposures (CVE)-2026-24858 [Common Weakness Enumeration (CWE)-288: Authentication Bypass Using an Alternate Path or Channel] allows malicious actors with a FortiCloud account and a registered device to log in to separate devices registered to other users in FortiOS, FortiManager, FortiWeb, FortiProxy, and FortiAnalyzer, if FortiCloud single sign on (SSO) is enabled on devices.[1]

Users are vulnerable to CVE-2026-24858 even if they updated Fortinet devices to address previously disclosed FortiCloud SSO bypass vulnerabilities CVE-2025-59718 and CVE-2025-59719 [CWE-347: Improper Verification of Cryptographic


dmz
corporate
IoT
guest
factory
INTERNET / WAN
PERIMETER FIREWALL
DMZ
inbound 443 only
WEB
DATABASE
DNS
REVERSE PROXY
MAIL RELAY
VPN
SFTP
CORPORATE
authenticated, full egress
DOMAIN CTRL
SERVERS
WORKSTATIONS
LAPTOPS
PRINTERS / MFP
VOIP PHONES
WIFI APs
IOT
no inbound, vendor cloud only
SIGNAGE
KIOSKS
CAMERAS
DOOR ACCESS
BADGE READERS
HVAC / BMS
UPS AND PDU
GUEST WIRELESS
internet only, client isolation
IPHONE
ANDROID
TABLETS
LAPTOPS
BYOD
VENDOR LAPTOP
SMART TV
FACTORY
IT to OT conduit, L3.5 only
L4/L5
OFFICE
WORKSTATIONS
PRINTERS
SERVERS
OT FIREWALL
L3.5
DMZ
PATCH
HIST MIRROR
JUMP
L3
MFG OPS
WORKSTATION
HISTORIAN
I/O SERVER
L2
SUPERVISORY
SCADA
HMI
ENG WS
L1
BASIC CTRL
PLC
RTU
SIS SAFETY
L0
PROCESS
SENSORS
ACTUATORS
MACHINERY
CAMERAS

Hunting for Leaks

Focus is on three techniques:

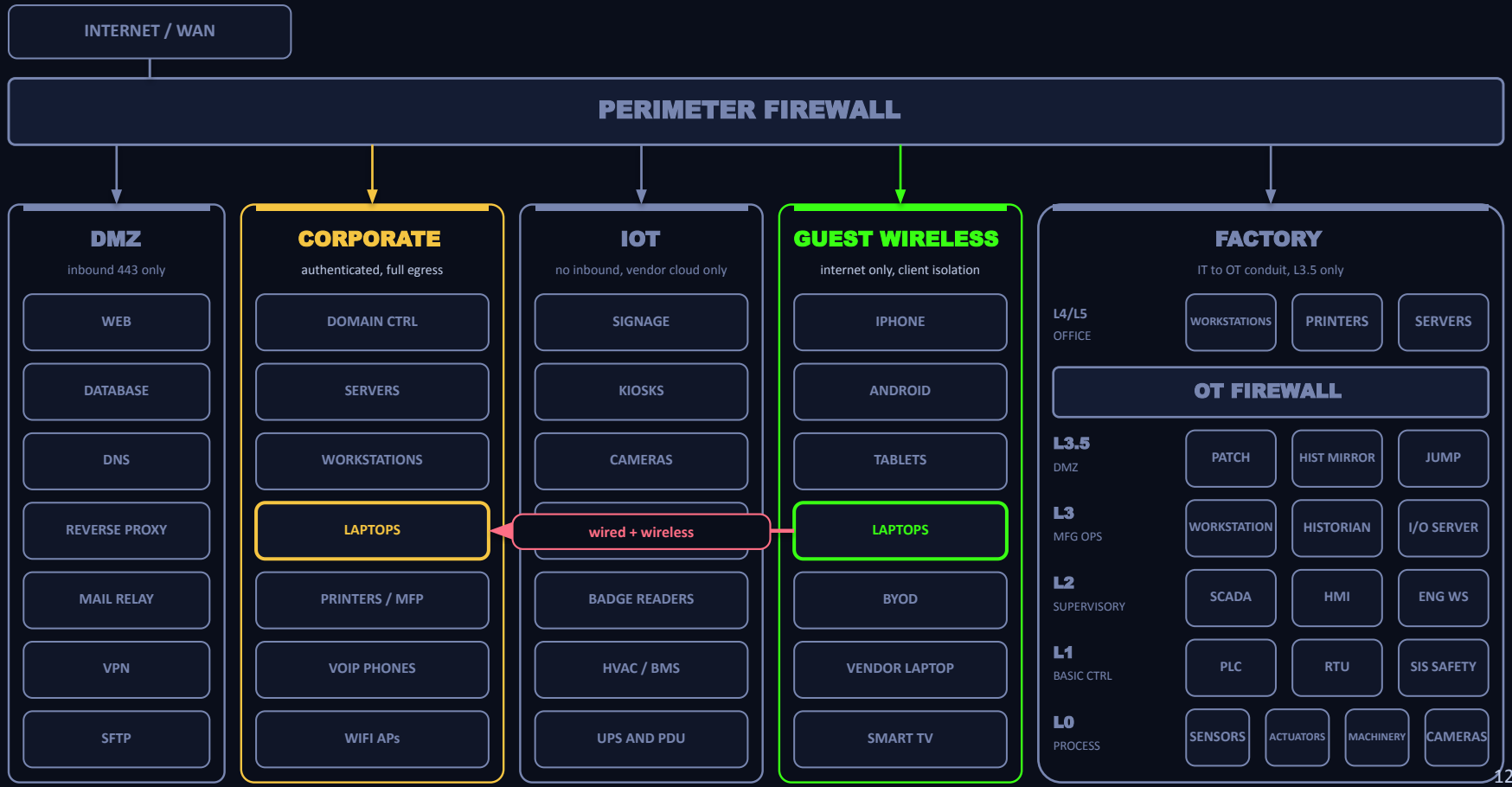
- Identify network bridges (pivot points)
- Explore cross-protocol gateways
- Test out-of-band management

Bridges

Any system with more than one network interface can become a pivot point.

WHERE IT BREAKS

One laptop, wired and wireless



Multi-homed assets are the standard

- Printers, NAS, phones, TVs, cameras
- Building automation, industrial controls
- Point of sale, access controls
- Every laptop with a container runtime
- 28 interfaces on a stock MacBook
- IPv6 auto-addressing

```
~ $ ifconfig | grep fe
        inet6 fe80::1%lo0 prefixlen 64 scopeid 0x1
        inet6 fe80::8303:bc79:ab87:1385%utun0 prefixlen 64 scopeid 0x12
        inet6 fe80::10c3:108c:424b:1a73%en0 prefixlen 64 secured scopeid 0xe
        inet6 fe80::2436:f4ff:fe5b:d936%awdl0 prefixlen 64 scopeid 0x13
        inet6 fe80::2436:f4ff:fe5b:d936%llw0 prefixlen 64 scopeid 0x14
        inet6 fe80::473:9a81:8402:ee19%utun1 prefixlen 64 scopeid 0x15
        inet6 fe80::2692:7b70:9a8:26eb%utun3 prefixlen 64 scopeid 0x17
        inet6 fe80::ce81:b1c:bd2c:69e%utun4 prefixlen 64 scopeid 0x18
        inet6 fe80::99eb:5f9:29b1:3ec8%utun5 prefixlen 64 scopeid 0x19
        inet6 fe80::d0ba:55bb:a283:ebaa%utun6 prefixlen 64 scopeid 0x1a
        inet6 fe80::f468:1527:d512:60a0%utun7 prefixlen 64 scopeid 0x1b
        inet6 fe80::e73e:d177:4661:7dfc%utun8 prefixlen 64 scopeid 0x1c
        inet6 fe80::fc:4298:be02:a52a%en8 prefixlen 64 secured scopeid 0x10
        inet6 fe80::f0c8:77ff:fe9f:62a%en12 prefixlen 64 scopeid 0x11
```

Sources of unintended multi-homing

Migration servers

A second interface added for a cutover, left cabled and configured after the project closed

Hypervisors

A management interface plus a virtual switch uplink on every guest VLAN the host serves

Multifunction printers

Wired and wireless interfaces active simultaneously, frequently on different segments

Vendor-managed appliances

An outbound support tunnel and an inbound management interface, both persistent

Engineering workstations

Corporate wireless, a tethered handset, and a cable to the equipment under test

Cellular gateways

Terminate inside the site network and present a management plane on a carrier address

IPv4 forwarding defaults

- Default is 0 on mainline Linux
- Container runtimes set it to 1
- So does hypervisor NAT networking
- Tailscale subnet routers require it
- WireGuard PostUp can set it to 1
- No log entry, no event

BEFORE AND AFTER A CONTAINER RUNTIME INSTALL

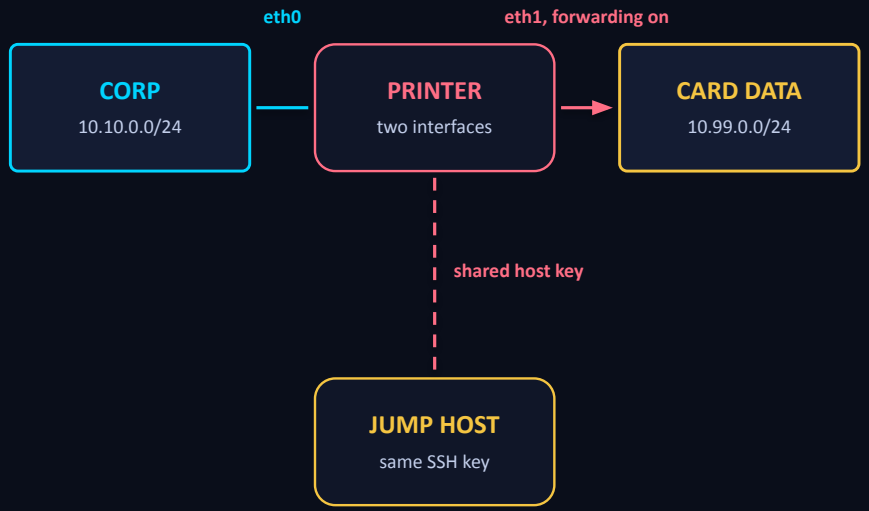
```
$ sysctl net.ipv4.ip_forward
net.ipv4.ip_forward = 0

$ sudo apt-get install -y docker-ce
  ...
$ sysctl net.ipv4.ip_forward
net.ipv4.ip_forward = 1

$ iptables -S FORWARD | head -1
-P FORWARD DROP
$ iptables -S DOCKER-USER
-A DOCKER-USER -j RETURN
```

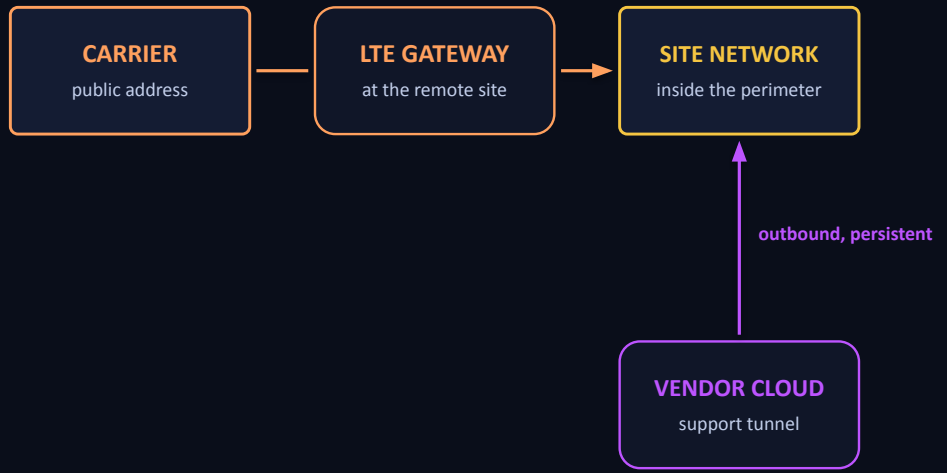
A printer on two segments

- Wired interface, corporate segment
- Second interface for a finance scanner
- IPv4 forwarding enabled in the web UI
- SSH host key shared with the jump host
- Firmware predates the current contract



Vendor tunnels and cellular gateways

- Support tunnels dial out and stay up
- Gateways terminate in the site network
- Both in a contract or support agreement
- Neither is in the firewall rule base
- Neither can be removed on its own



Removing the bridge removes a service under contract.

Finding Bridges

Three specific methods:

- Ask the system to leak it’s IP addresses
- Correlate unique identity across IPs
- Forcibly route and check errors*

NetBIOS: The lesser known get address

- Packet one: node status, NBSTAT 0x21
- Reply: name table and adapter MAC
- Packet two: name query for that name
- Reply: every address registered to it
- Includes addresses on other segments

```
$ nbtscan -v 10.10.0.41
FILESRV01    <00> UNIQUE  workstation
CORP         <00> GROUP   domain
MAC          00:1b:21:3c:2f:14

$ nbname query FILESRV01 10.10.0.41
FILESRV01 -> 10.10.0.41
             10.99.0.7
```

Still often enabled on Windows and answered by network appliances.

SNMP: Direct and via peer ARP caches

- ipAddrTable: every IPv4 address
- ifTable: the name behind each ifIndex
- ipNetToMediaTable: the ARP cache
- ipRouteTable: the networks it routes to
- A read-only community is enough

```
$ snmpwalk -v2c -c public 10.10.0.41 ipAddrTable
ipAdEntAddr.10.10.0.41  = IpAddress: 10.10.0.41
ipAdEntAddr.10.99.0.7   = IpAddress: 10.99.0.7
ipAdEntIfIndex.10.99.0.7 = INTEGER: 3
ipAdEntNetMask.10.99.0.7 = IpAddress: 255.255.255.0
```

The ARP cache of a printer can be used to leak IP/MACs of neighbors

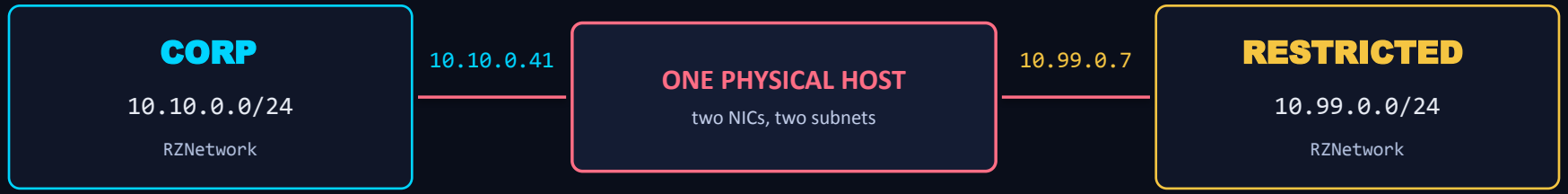
DCE/RPC: The OXID2Resolver

- IObjectExporter, opnum 5, ServerAlive2
- Unauthenticated bind, no NTLM required
- Returns the server's string bindings
- Returns name, IPv4 and **IPv6** addresses
- Effective at finding VPN users too

```
$ python3 oxidresolver.py 10.10.0.41
[*] ServerAlive2 string bindings:
    FILESRV01
    10.10.0.41
    10.99.0.7
    172.16.4.9
    fe80::21b:21ff:fe3c:9f04
```

TCP port 135, no credential, and no limit on number or type of addresses

Identifying an asset across networks



ONE HOST KEY ON TWO ADDRESSES

```
10.10.0.41      SSH-2.0 host key      SHA256:9Zk4xO...q7Td
10.99.0.7       SSH-2.0 host key      SHA256:9Zk4xO...q7Td
```

TLS: Certificates & SANs

- Self-signed CN is the management name
- subjectAltName carries the IP address
- Redirects point at an internal hostname
- Bonus: Unique correlation via SPKI

```
$ openssl s_client -connect 10.10.0.41:443 </dev/null 2>/dev/null \
    | openssl x509 -noout -subject -ext subjectAltName
subject=CN = ilo-4823.mgmt.corp.local
X509v3 Subject Alternative Name:
    DNS:ilo-4823.mgmt.corp.local,
    IP Address:10.99.0.7
```

SNMPv3: Pre-auth identity leak

- No credential and widely supported
- Report reply: msgAuthoritativeEngineID
- Some encode a raw MAC or a serial
- Stable across reboots on most stacks
- Same engine ID means the same device

```
$ snmpget -v3 -u nosuchuser -l noAuthNoPriv 10.10.0.41 sysDescr.0
snmpget: Unknown user name
  msgAuthoritativeEngineID: 80001f88803c9f041b21

$ snmpget -v3 -u nosuchuser -l noAuthNoPriv 10.99.0.7 sysDescr.0
snmpget: Unknown user name
  msgAuthoritativeEngineID: 80001f88803c9f041b21
```

Corroborate with leaked counters to confirm unique identity and monitor activity

NTLMSSP: Hostname, DNS name, version

- SMB2, RDP, LDAP, SMTP, HTTP offer NTLM
- CHALLENGE carries AV_PAIR target info
- NetBIOS, DNS computer name, DNS domain
- Returned before any credential is sent
- Same computer name on two addresses
- RDP also returns a TLS certificate

```
$ ntlm-info smb 10.10.0.41
  NetBIOS computer : FILESRV01
  DNS computer     : filesrv01.corp.local
  DNS domain       : corp.local

$ ntlm-info rdp 10.99.0.7
  NetBIOS computer : FILESRV01
  DNS computer     : filesrv01.corp.local
```

SSH: Unique hostkeys

- Offered during key exchange, no login
- Fingerprint is stable across reboots
- Duplicates? That also useful to know
- Check against https://badkeys.info

```
$ ssh-keyscan -t ed25519 10.10.0.41 10.99.0.7
10.10.0.41 ssh-ed25519 AAAAC3NzaC1lZDI1...q7Td
10.99.0.7  ssh-ed25519 AAAAC3NzaC1lZDI1...q7Td

# same key on 40 hosts is a golden image
# same key on 2 addresses is one machine
```

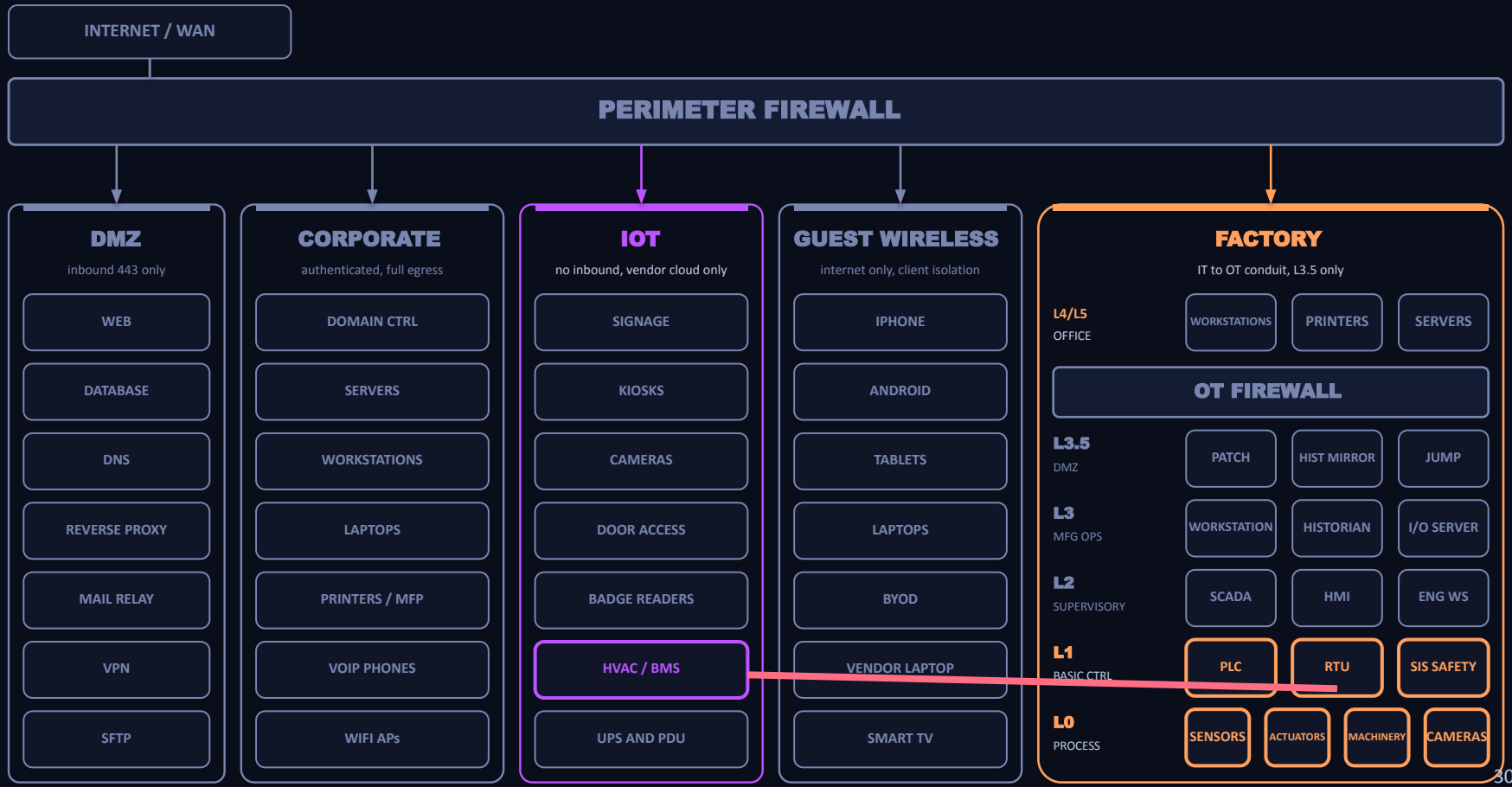
Protocol gateways

Three specific methods:

- Scan for protocol gateways to enumerate peers
- Subscribe and listen to broadcasts
- Relay connections through peers

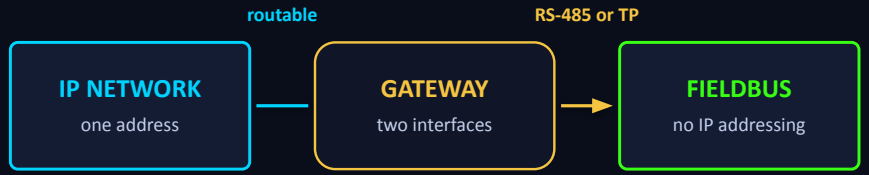
WHERE IT BREAKS

- one address, the bus behind it



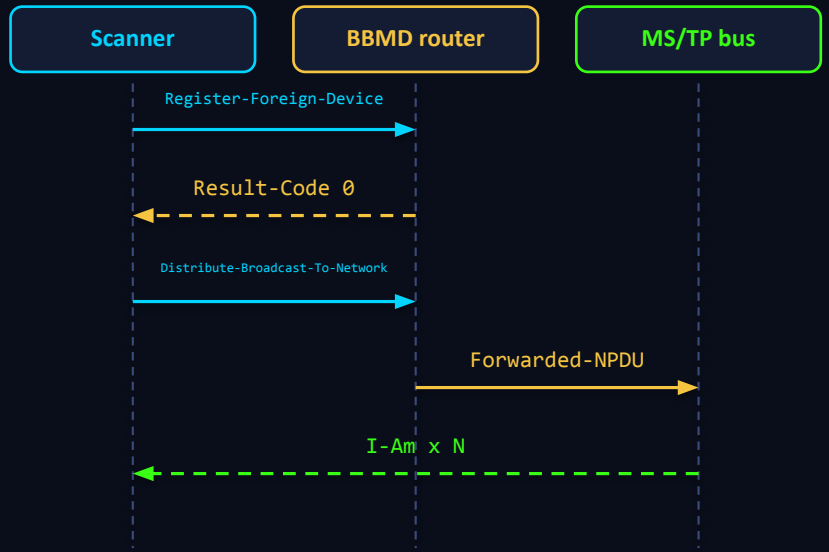
TCP/IP to serial & non-IP links

- A single IP may expose hundreds of backplane devices
- Every protocol has its specific quirks
- Local broadcast vs unicast vs subscribe



BACnet/IP: Register as a foreign router

- UDP 47808, no authentication*
- BBMD forwards broadcasts across subnets
- Register-Foreign-Device adds the sender
- BVLC 9 wraps a Who-Is broadcast
- BBMD re-emits it as Forwarded-NPDU



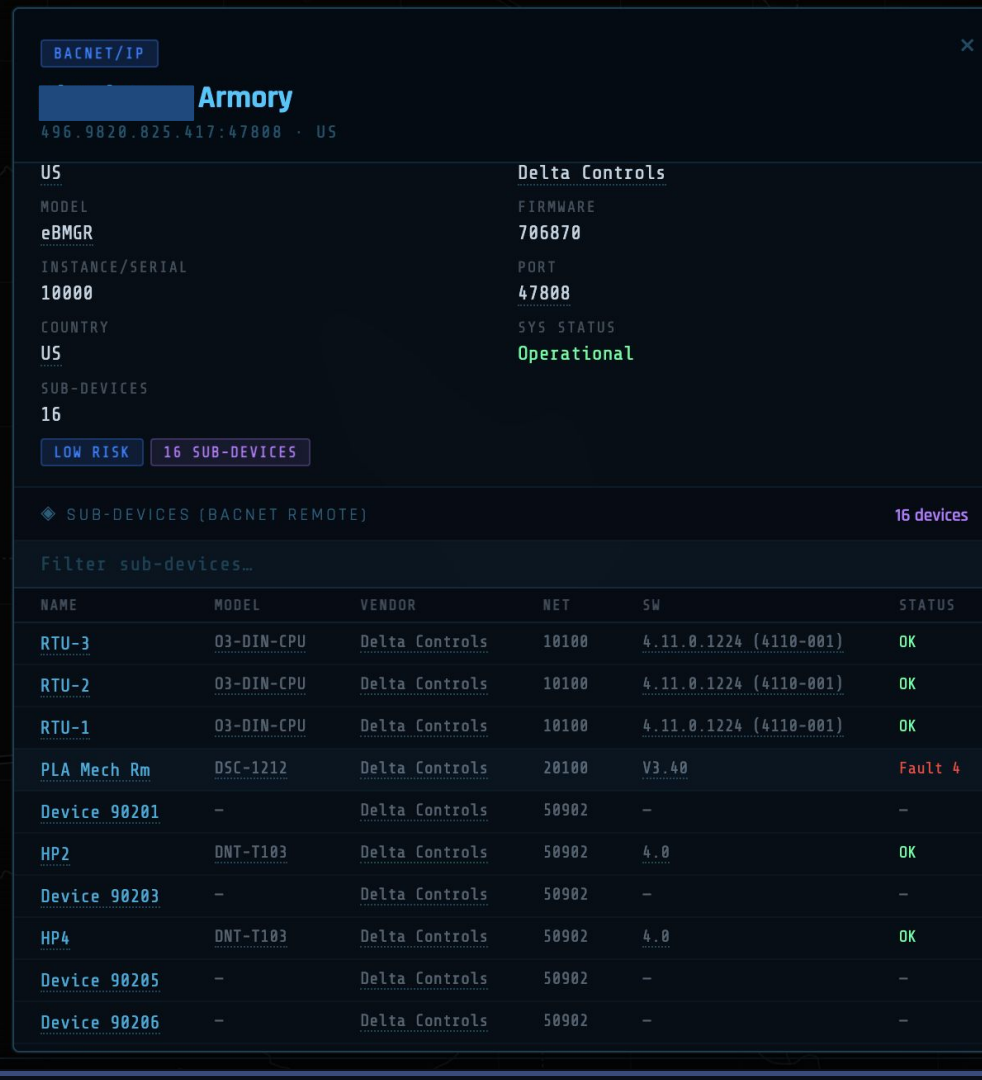
four packets before the first I-Am, no credential

Each I-Am carries a device instance, max APDU length, segmentation support and vendor identifier.

POPULATION

BACnet/IP Scope

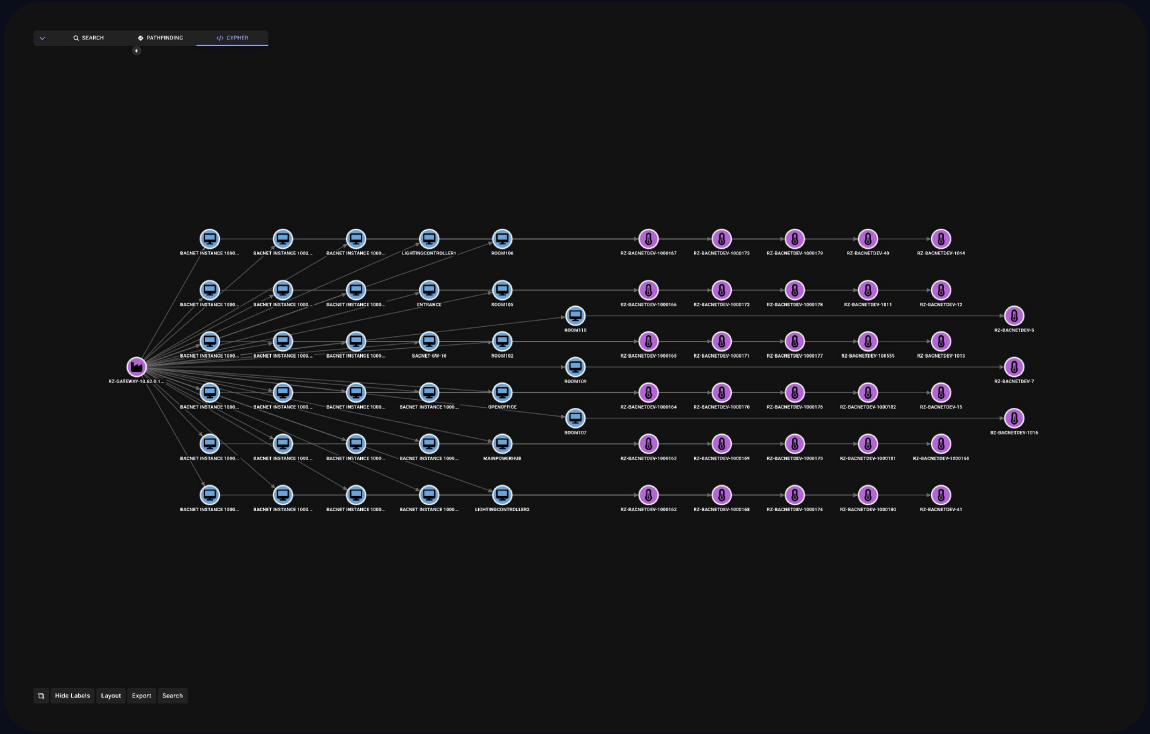
- The gateway device itself might be low-risk
- It can expose hundreds of other systems
- Boilers, chillers, valves,doors
- Fire suppression and safety



IN THE GRAPH

BACnet: Correlating though BloodHound

ONE BACNET/IP ADDRESS, EXPANDED INTO WHAT ANSWERS BEHIND IT



Each node on the right is one device object on the MS/TP side. The gateway is the single node on the left that a firewall rule can see.

The bus names what it controls. Entrance, Room102, LightingController1 and MainPowerHub are all in there, and no credential was supplied to learn any of it.

BloodHound Community Edition, RZGateway and RZBACnetDevice.

EtherNet/IP: CIP relays and routing

- 44818: UDP ListIdentity, TCP messaging
- Requests carry an explicit source route
- Each segment: port and link address
- Port 1 is the backplane, link the slot
- The path travels inside UnconnectedSend

CIP CONNECTION PATH, PORT AND LINK ADDRESS PER SEGMENT

0x01	0x02	0x12	0x0C	192.168.1.10	0x01	0x00
port: backplane	slot 2	port: ethernet	length 12	link address, ASCII	port: backplane	slot 0

Backplane slot 2, out to 192.168.1.10, then slot 0.

Iterating the link address enumerates a chassis from one TCP session.

CIP: Forward IP > Serial > IP > Serial > IP

- IP-capable port takes an IP as the link
- The gateway originates that hop itself
- The client supplies it, unauthenticated
- Reply or timeout gives the outcome
- The client never touches that segment
- SSRF but for Ethernet/IP!

the gateway opens the connection on your behalf

Modbus: Unit identifiers

- TCP 502, no authentication
- MBAP carries a one-octet unit id (RS-485 serial)
- Each value maps to one RS-485 device
- Read/write registers and FCs*
- Register write challenges…

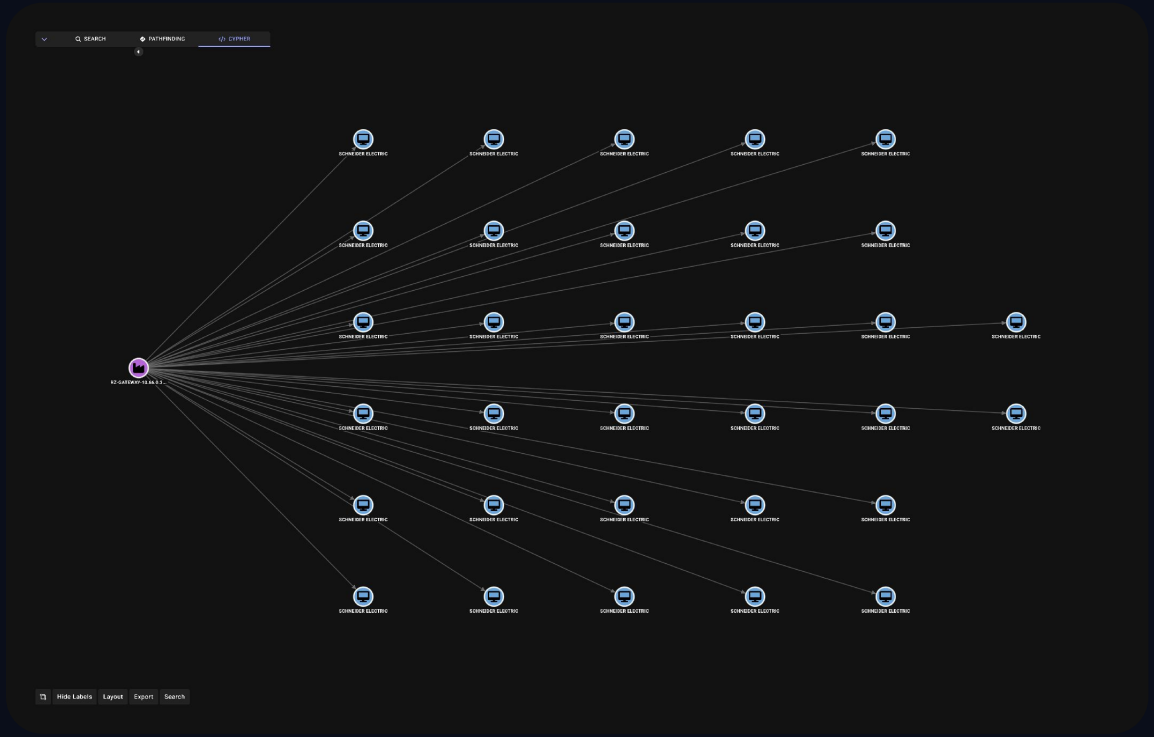
Four header fields, then the one addressing byte.

Specified before control networks were assumed hostile. Vendors special purpose conflicting registers.

IN THE GRAPH

Modbus: Correlating via BloodHound

THE UNIT ID SWEEP, DRAWN AS EDGES



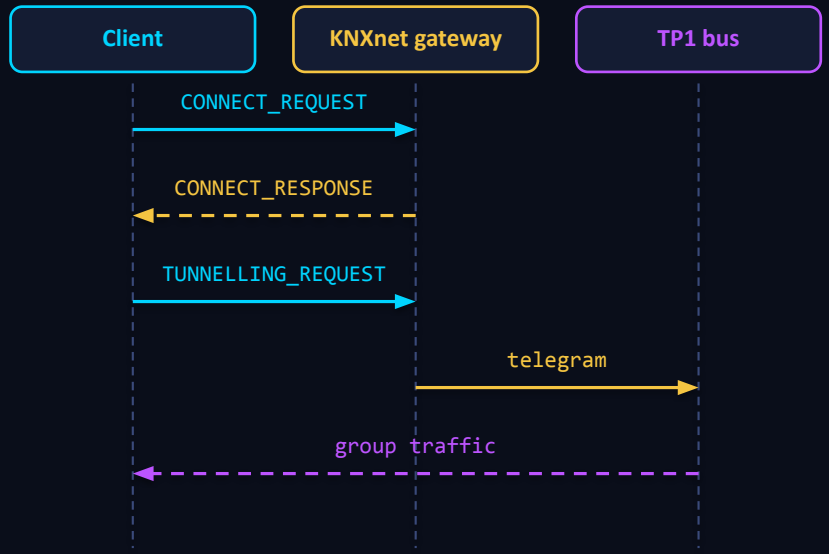
One TCP 502 socket on the left. Every node on the right answered a read on a different unit id, and each one is a separate RTU on the RS-485 line.

The population behind the address is 32 times what a port scan reports, and nothing in the exchange authenticated.

BloodHound Community Edition, RZGateway fan-out.

KNXnet/IP: Tunnel connections

- UDP 3671, no authentication
- Gateway allocates a bus address
- The client is a device on TP1
- Group telegrams observable
- Individual addresses enumerable



lighting, HVAC, blinds and access control share the segment

KNX Secure is a later, opt-in addition.

Internet exposure

6,611

protocol routers

5,006 BACnet, 1,605 EtherNet/IP, across 100 countries

101,933

devices behind them

enumerated through the routers, none reachable directly from the scanner

15x

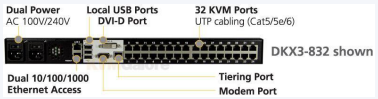
sub-devices

average devices per router across the 6,611

https://hdm.io/data/ics.html

Out-of-band access

BMC
on the motherboard



IP KVM
rack appliance

SERIAL
console aggregation

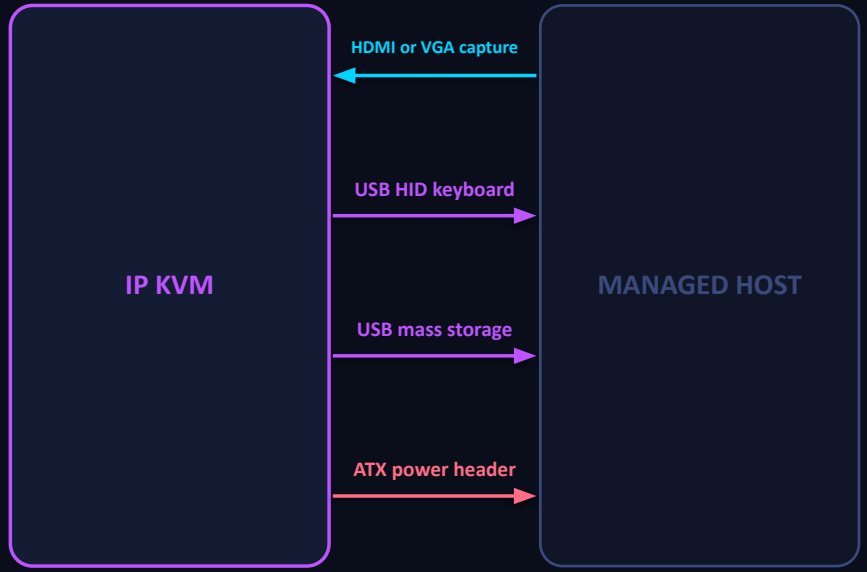
PDU
outlet-level power control

WHERE IT BREAKS

● attached to these, never inventoried

IP KVM peripheral emulation

- Captures video at the connector
- Presents a USB HID keyboard
- Presents USB mass storage
- Some present a USB NIC
- All below the host OS boundary



No host OS credential is involved. The session reaches BIOS, boot order and media.

Open-hardware IP KVM devices

PiKVM
Raspberry Pi based

NanoKVM
Sipeed, RISC-V

TinyPilot
Voyager

JetKVM
single board

BliKVM
PCIe and standalone

Equivalent capability to the rack appliance, below most procurement thresholds.

vendor product images, see assets/SOURCES.md

Serial device servers

Moxa
NPort series

Digi
PortServer TS

Lantronix
device servers

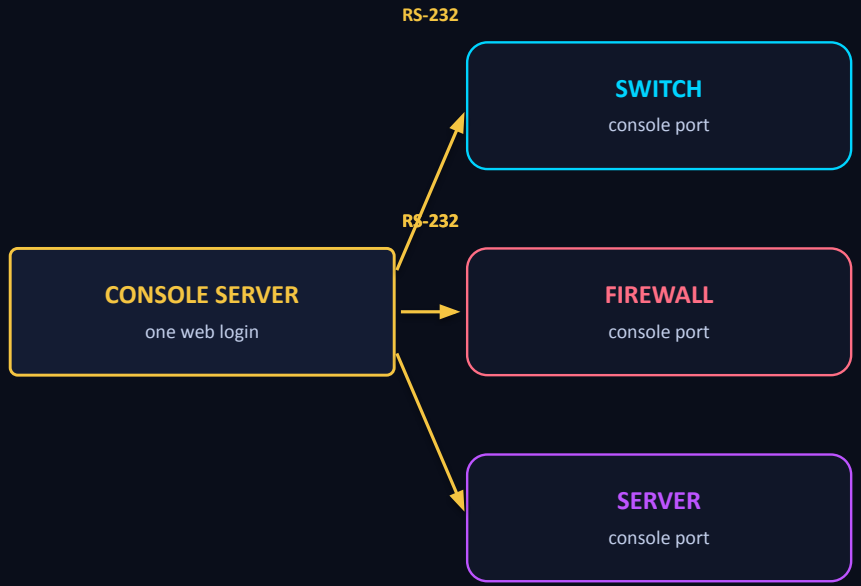
Perle
IOLAN

Pre-auth UDP discovery: Lantronix 30718, Moxa 4800, Digi ADDP 2362.

vendor product images, see assets/SOURCES.md

Console servers and pre-boot access

- Serial line below every network ACL
- Break at boot reaches ROM monitor
- Boot loader responds pre-OS
- One appliance owns the whole cabinet
- Cellular backup links are common



32 and 48 line densities are common

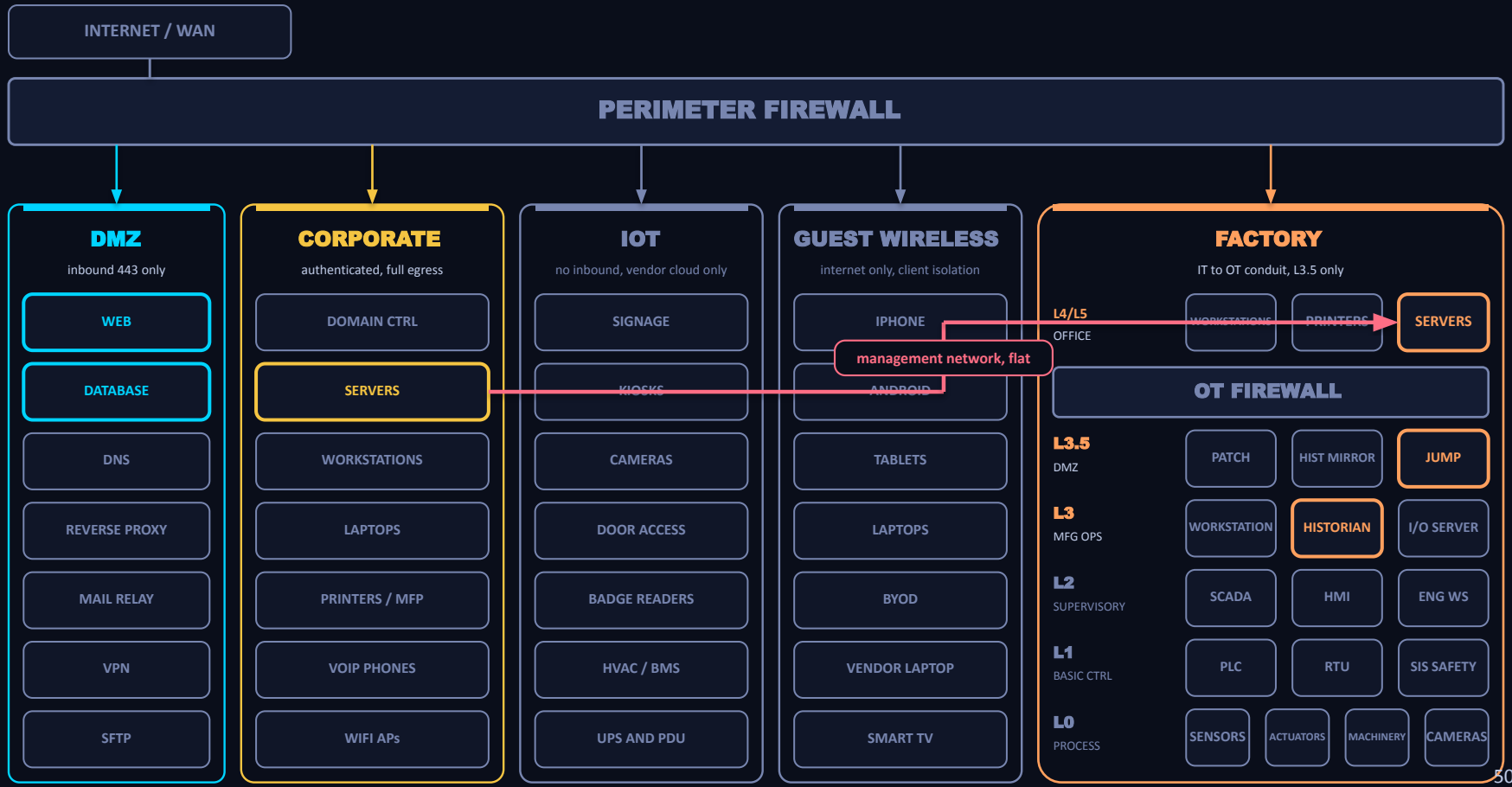
One login yields physical-equivalent access to every device in the cabinet.

Baseboard management controllers

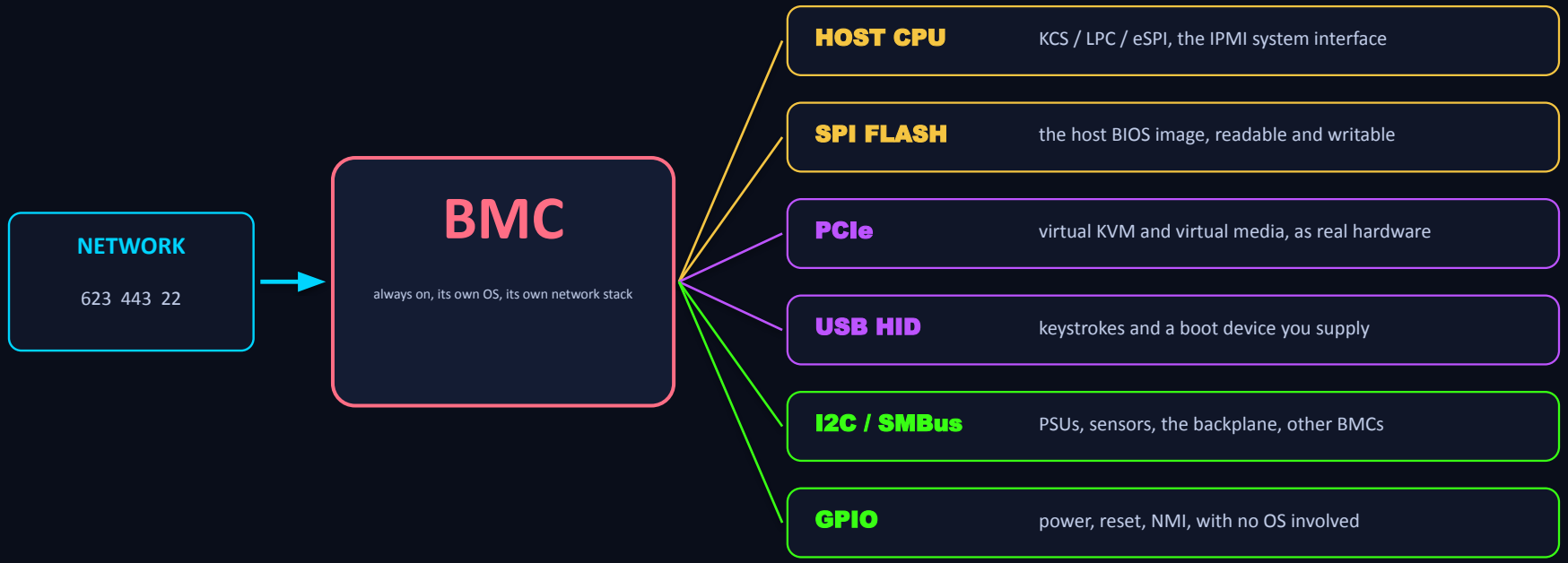
An independent processor on the mainboard with its own operating system, network stack and power domain, and a bidirectional trust relationship with the host. This is the control edge, and it points both ways.

WHERE IT BREAKS

A second computer inside every server

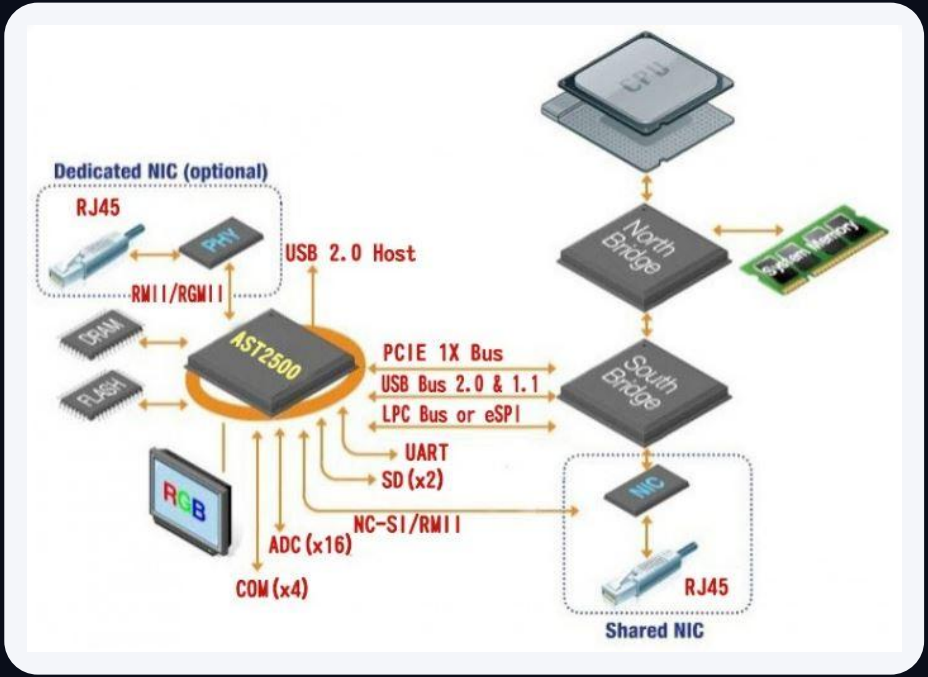


BMC hardware interfaces



Network access to the BMC is equivalent to physical access.

ASPEED AST2500 block diagram



Upper left: Dedicated NIC, annotated optional. Lower right: Shared NIC over NC-SI or RMII. The same part supports both; the board designer selects one.

LPC or eSPI to the south bridge. This is the IPMI system interface, and it carries no authentication from the host side.

PCIe, USB 2.0 host controller and the video path. Virtual media and KVM are endpoints on real buses.

Vendor documentation, reproduced unmodified. ServeTheHome, Explaining the BMC in Servers

Trust in both directions

Host to BMC

- No credential required from the host side
- KCS carries no authentication by design
- PCIe and LPC bridges expose BMC memory to the host
- Persists across operating system reinstallation
- Produces no packet and no authentication event

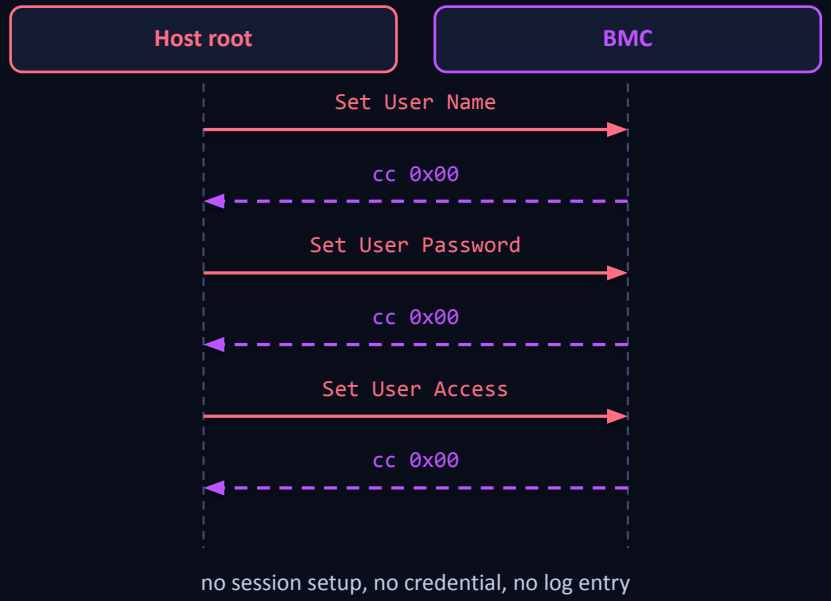
BMC to host

- Requires BMC access as a precondition
- Every subsequent path is a shipped feature
- Boot media, input injection, host memory*
- Yields host control before any OS loads
- Leaves no host log and no network record

Compromise of either side yields the other. The host-to-BMC direction requires no vulnerability, only root on a machine already inside the estate.

The KCS system interface

- The in-band IPMI transport
- Linux: /dev/ipmi0, root owned
- The spec defines the host as trusted
- No authentication on the channel
- Account creation and flash reachable



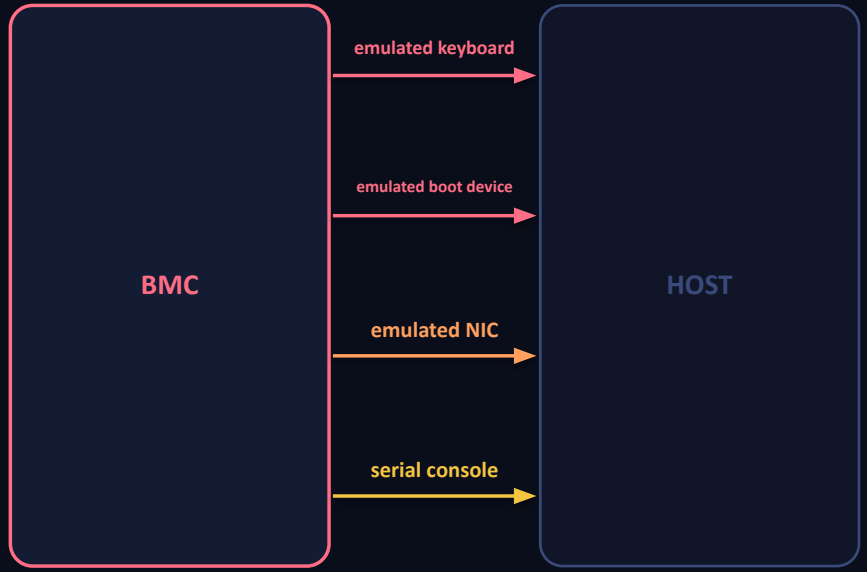
Host root can set a new BMC password or flash firmware over KCS

In-band channels from host to BMC

KCS and BT	The previous slide. BT is the same trust model on a block-transfer FIFO, present on every stack examined
HPE CHIF	Proprietary iLO host channel. Redfish accepts requests on it with no session token, marked AutoLogin
PCIe P2A and DMA	CVE-2019-6260. ASPEED AST2400 and AST2500 leave AHB bridges enabled, giving the host the BMC address space
I2C and MCTP	The management bus to every component. The trust model assumes every device on the bus is genuine
Serial and debug UART	Console access, and on some parts an additional route onto the internal bus
Host-side private ports	HPE binds IPMI and RIBCL on TCP 49623 for host-resident tooling

Control paths from BMC to host

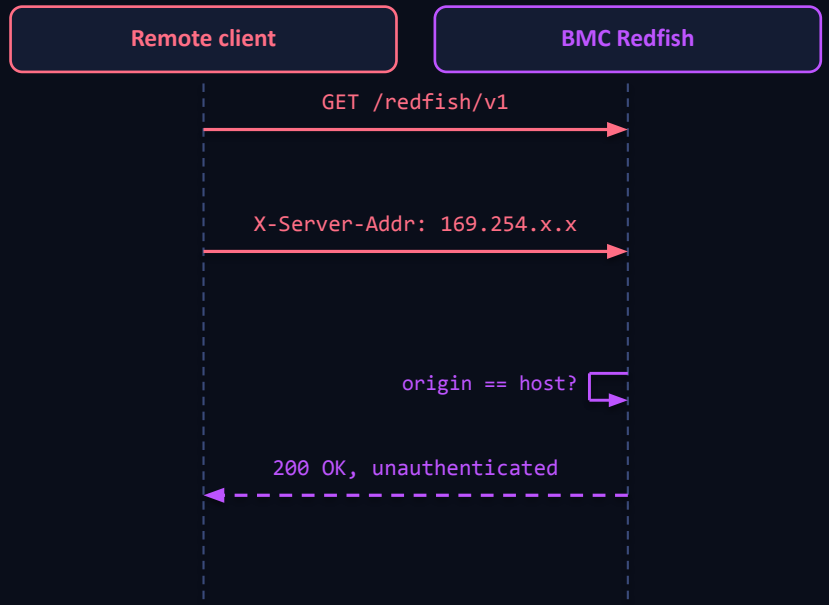
- USB HID keyboard emulation
- Virtual media as bootable USB, Dell MASTER
- USB CDC ethernet into the host
- Serial over LAN
- Direct memory access (XDMA*)



A BMC compromise is a host compromise conducted entirely through supported functionality.

Redfish host interface trust

- BMC presents USB ethernet
- That interface is treated as local
- Local origin skips authentication
- Origin read from X-Server-Addr
- The first BMC entry in CISA KEV



A trust boundary implemented as a string comparison against a header the requester controls.

IPMI & RAKP is still a trainwreck

```
oobscan scan 10.114.122.48

CRITICAL 10.114.122.48   ipmi.v20.rakpHash        623/udp ipmi username=ADMIN hash=fbb98eff2fac03008a907f300
d739055207b25a7a944ce35ce67dd4240be14b9ee290ca6c9182923000000000000000000000000000000140541444d494e:045cd3
3c07fc8b9a23dae164a7919a9fc47d54aa hashAlgo=hmac-sha1 cracked=ADMIN crackedFrom=defaults defaultCredential=t
rue [CVE-2013-4786]
CRITICAL 10.114.122.48   ipmi.v20.defaultCreds    623/udp ipmi username=ADMIN password=ADMIN source=default
cracked=defaults proof="openSession=ok bmcSession=0x0003ac2f rakp2Status=0x00 rakp2Hmac=verified"
CRITICAL 10.114.122.48   ipmi.v15.defaultCreds    623/udp ipmi username=ADMIN password=ADMIN source=md5 priv
=admin proof="activateSession=ok setPrivCc=0x00 grantedPriv=admin readCc=0x00"
CRITICAL 10.114.122.48   ssh.defaultCreds         22/tcp ssh username=ADMIN password=ADMIN response="Insyde
SMASH-CLP System Management Shell, versions" access=shell
CRITICAL 10.114.122.48   http.supermicro.defaultCreds 443/tcp https vendor=supermicro username=ADMIN passwor
d=ADMIN loginVia=httpForm formEncoding=plaintext source=ipmi.rakpCracked evidenceEndpoint=/cgi/login.cgi pro
of="authenticated session via /cgi/url_redirect.cgi?url_name=mainmenu"
INFO     10.114.122.48   id.final                 vendor=supermicro product=X11 version=01.74.14 type=bmc co
nfidence=high defaultCert=low defaultCreds=critical kvm=info rakpHash=critical
```

The hash comes out before authentication, cracks to the vendor default, and the credential then works over IPMI 2.0, IPMI 1.5 and SSH.

The account table

```
oobscan ipmi -u ADMIN -P ADMIN 10.114.122.48 user list
```

ID	Name	Privilege	Enabled	Call In	Link Auth	IPMI Auth
1		0x00	yes	yes	no	no
2	ADMIN	admin	yes	no	no	yes
3	LOWPRIV	user	yes	yes	yes	yes
4	OPPRIV	operator	no	yes	yes	yes
5		0x00	no	yes	no	no
6		0x00	no	yes	no	no
7		0x00	no	yes	no	no
8		0x00	no	yes	no	no
9		0x00	no	yes	no	no
10		0x00	no	yes	no	no

System inventory

```
oobscan ipmi -u ADMIN -P ADMIN 10.114.122.48 fru print

+--------+----------------------+----------------------+
| Device | Field                | Value                |
+--------+----------------------+----------------------+
| 0      | Chassis Type         | Other                |
| 0      | Chassis Part Number  | CSE-217HQ+-R2K20BP3  |
| 0      | Chassis Serial       | C2170AH35A30021      |
| 0      | Board Mfg            | Supermicro           |
| 0      | Board Product        | NONE                 |
| 0      | Board Serial         | OM187S000139         |
| 0      | Board Part Number    | X11DPT-PS            |
| 0      | Product Manufacturer | Supermicro           |
| 0      | Product Name         | NONE                 |
| 0      | Product Part Number  | SYS-2029TP-HC1R      |
| 0      | Product Version      | NONE                 |
| 0      | Product Serial       | S265503X8B11038      |
| 0      | Product Asset Tag    | 00000037BD51         |
+--------+----------------------+----------------------+
```

Chassis part number, board part number, and three serials. Enough to look the machine up in somebody else's asset system.

Device power

```
oobscan ipmi -u ADMIN -P ADMIN 10.114.122.48 chassis status
```

Setting	Value
Coolingfault	no
Drivefault	no
Identifyon	yes
Lastpowerevent	none
Mainpowerfault	no
Powercontrolfault	no
Powerinterlock	no
Poweron	yes
Poweroverload	no
Powerrestorepolicy	restore
Raw	210040

Authority by layer

Width is authority over adjacent layers.

Mapping a BMC to its host

- iLO /xmldata lists the server NIC IPs
- SNMP interfaces for shared NIC MACs
- SNMP ARP cache maps those MACs to IPs
- The server often sits in another subnet

THE CONTROLLER NAMES ITS OWN SERVER

```
$ curl -sk https://192.0.2.11/xmldata?item=all
  <NIC><DESCRIPTION>iLO 5</DESCRIPTION>
   <IPADDR>192.0.2.11</IPADDR>    the controller
  <NIC><DESCRIPTION>Port 1</DESCRIPTION>
   <IPADDR>198.51.100.20</IPADDR>    its server
$ snmpwalk -v2c -c public 192.0.2.39 \
      networkDeviceTable
  ...90.1.6.1 = 3c:fd:fe:00:00:90
$ snmpwalk -v2c -c public 192.0.2.39 \
      ipNetToMediaTable
  ...2.3.192.0.2.33 = 88:18:f1:00:00:12
  ...2.3.192.0.2.36 = 3c:fd:fe:00:00:90
```

Both disclosures are documented product behaviour, and both name the machine behind the controller.

Mapping a BMC to its host

Active sweep

125,430 out-of-band devices probed. Supplies the BMC side of every join

Shodan bulk corpus

3.45 billion banners read across 25 daily files, 8.7 TB. Supplies the host side

Reverse DNS

2.2 million PTR records over every candidate subnet. The one universal oracle

Read-only SNMP

22,001 answered SNMP. At least 2,995 took the community they shipped with

Successful correlations to hosts

971

servers located

out of 125,430 out-of-band devices probed

72%

in a different /24

no subnet-scoped search reaches these

531

on internal addresses

RFC1918, disclosed from the public internet

0

needed a credential

every pair came from a pre-auth response

Every pair in this run came from one vendor path: an iLO publishing the address configured on each of its server's network ports to an unauthenticated GET. The Dell side, an iDRAC handing its own ARP cache to the shipped community, needs collection this scan predates.

High-level stats

50,945 speak IPMI	40.6 percent of everything found, on a protocol whose last revision is from 2015
23,288 leak a password hash	pre-auth, to anyone who asks them to open a session
52,695 expose KVM	remote keyboard and remote boot media, including virtual media, from outside
87,104 answer Redfish	the modern management API, reachable, 69 percent of the population
13,499 serve a shared TLS key	the private key ships inside the firmware image, identical across every unit
2,995+ answer default SNMP	the community string the vendor ships, and read access to the chassis inventory
2,023 take default logins	no bug and no bypass, the password sits on a page the vendor publishes

Movement across a management VLAN

Only step one needs a vulnerability.

The rest are documented features and a reused password.

Limits of a management VLAN

Stopped by a management VLAN

- Routed access from other segments
- Direct scanning of the controller estate
- Credential reuse from a user subnet
- Anything that has to cross a firewall

Not stopped

- A host reaching its own controller
- Anything arriving over the chassis bus
- One BMC password reused across the VLAN
- Any BMC-side vulnerabilities (IPMI, SSH)
- An attacker already inside the segment

Host security is a BMC control. The recommendation is a segment per device.

BloodHound & OpenGraph

Community Edition with many *Hounds

- Convert nearly anything to OpenGraph nodes
- Query relationships in Cypher, view in BHCE
- Scales great with PostgreSQL (vs neo4j)

Identity graphs and asset graphs

BloodHound Identity

- User to Group to Domain Admin
- GenericAll on OU to child objects
- Kerberoastable SPN to hash to credential
- Unconstrained delegation to TGT
- Cross-forest trust to SID history
- The path is the finding

Asset Relationships

- Switch to Port to VLAN to Subnet
- BMC manages Host, Host reaches BMC
- Gateway to Backplane to PLC
- One SSH key observed in two subnets
- One machine holding two addresses
- The path is still the finding

User-defined node and edge kinds

- Way beyond Active Directory
- Node kinds declared by the data
- Edge kinds too, no registration
- Upload JSON, query in Cypher
- No fork, no plugin, no patch

OPENGRAPH.JSON

```
{"graph": {
  "nodes": [
    {"id": "10.10.0.41",
     "kinds": ["RZAsset"],
     "properties": {"name": "mfp-03"}}],
  "edges": [
    {"start": {"value": "10.10.0.41"},
     "end":   {"value": "10.99.0.0/24"},
     "kind":  "RZInsideOfSubnet"}]
}}
```

Any domain expressible as nodes and edges inherits the pathfinding.

Unauthenticated collection

RMCP presence ping, UDP 623

ASF pong byte 20 bit 7 carries an explicit IPMI-supported flag

Get Channel Cipher Suites, 0x54

The reliable IPMI 2.0 indicator. A suite-17-only reply is a hardened policy, seen on several stacks

Get System GUID, 0x37

A 16-octet identity token with vendor structure

BACnet Who-Is and CIP ListIdentity

The enumeration primitives from part three, which also locate the gateway itself

Vendor UDP discovery

The pre-authentication services from part four, on 30718, 4800 and 2362

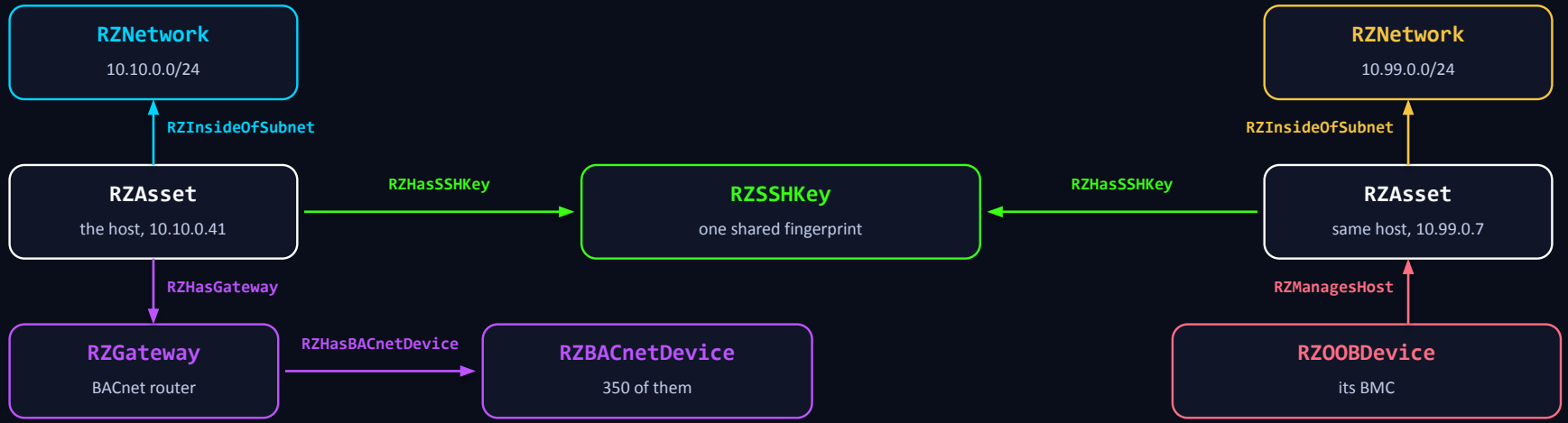
SSH, TLS, SNMP and SMB negotiation

Collected for the host key, certificate and engine identifier

Join keys for cross-asset matching

SSH host key fingerprint	Stable until a rekey. A mismatch is weak evidence of two machines
TLS certificate fingerprint	Stable until regeneration, which happens on hostname and clock changes
SMB2 server GUID	From the NEGOTIATE response, and regenerated on reboot. A mismatch proves nothing
SNMPv3 engine identifier	Stable on most stacks, regenerated by a minority. Corroborate against engine boots
BMC-derived MAC, IP, name	Easily polled via API endpoint pre-author via IPMI with incremental detail
Device serial number	Recoverable from several protocols. Survives reimaging and address changes

runZeroHound: Parses tons of tools



Bridging is one asset with two RZInsideOfSubnet edges. **RZManagesHost is what the controller discloses about its own host.**

network identity management OT gateway

runZeroHound

github.com/runZeroInc/runZeroHound

Cypher over the asset graph

SSH HOST KEY OBSERVED IN MORE THAN ONE SUBNET

```
MATCH (a1:RZAsset)-[:RZHasSSHKey]->(key:RZSSHKey)<-[:RZHasSSHKey]-(a2:RZAsset),
      (a1)-[:RZInsideOfSubnet]->(n1:RZNetwork),
      (a2)-[:RZInsideOfSubnet]->(n2:RZNetwork)
WHERE n1 <> n2 AND id(a1) < id(a2)
RETURN key.fingerprint, a1.displayname, n1.displayname, a2.displayname, n2.displayname
```

GATEWAYS FRONTING MORE THAN FIFTY DEVICES

```
MATCH (gw:RZGateway)-[:RZHasGatewayAssets]->(device:RZAsset)
WITH gw, count(device) AS cnt
WHERE cnt > 50
RETURN gw.displayname, gw.protocol, gw.ip, cnt ORDER BY cnt DESC
```

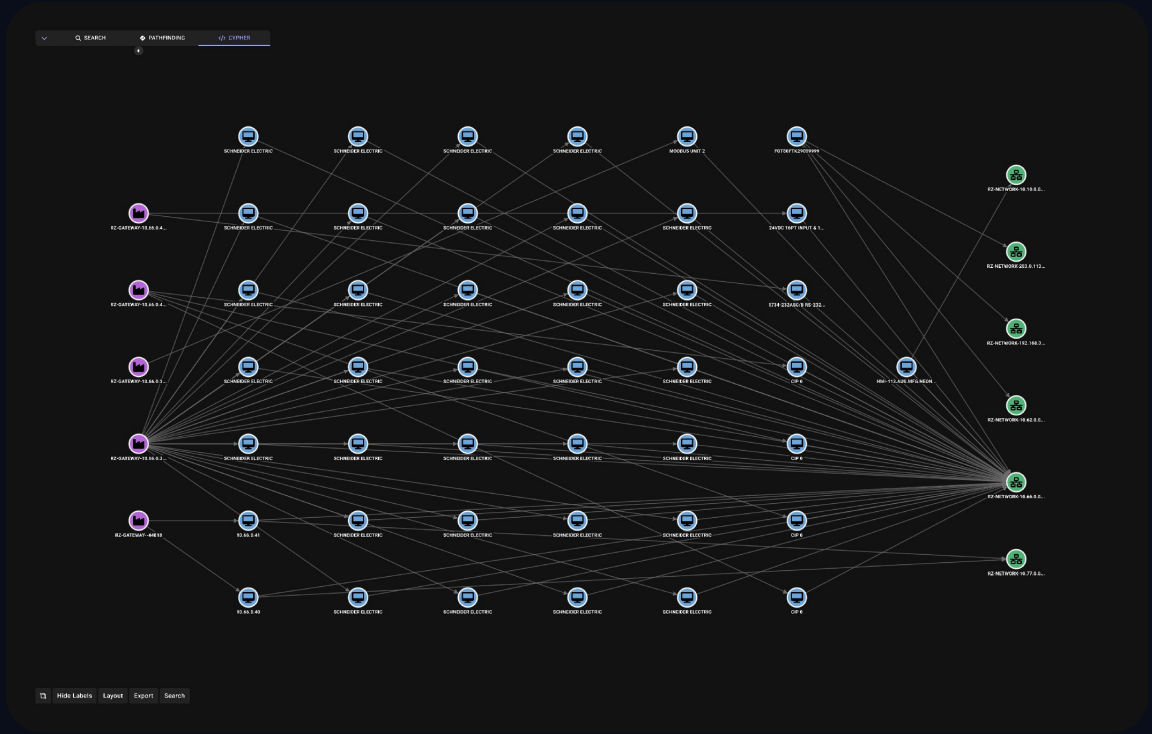
TLS CERTIFICATE PRESENTED IN MORE THAN ONE SUBNET

```
MATCH (a:RZAsset)-[:RZHasTLSCert]->(cert:RZTLSCert),
      (a)-[:RZInsideOfSubnet]->(net:RZNetwork)
WITH cert, collect(DISTINCT net.displayname) AS subnets, count(DISTINCT a) AS hosts
WHERE size(subnets) > 1 RETURN cert.cn, cert.sha1, subnets, hosts
```

IN THE GRAPH

From a fieldbus to another network

ONE QUERY, FOUR NODE KINDS, NINETY EDGES



Left: the protocol gateways from part three. Middle: the devices on the bus behind them. Right: the segments those devices sit in.

The two nodes bridging out are an HMI and the boundary firewall. Between them they carry the OT segment to five other networks.

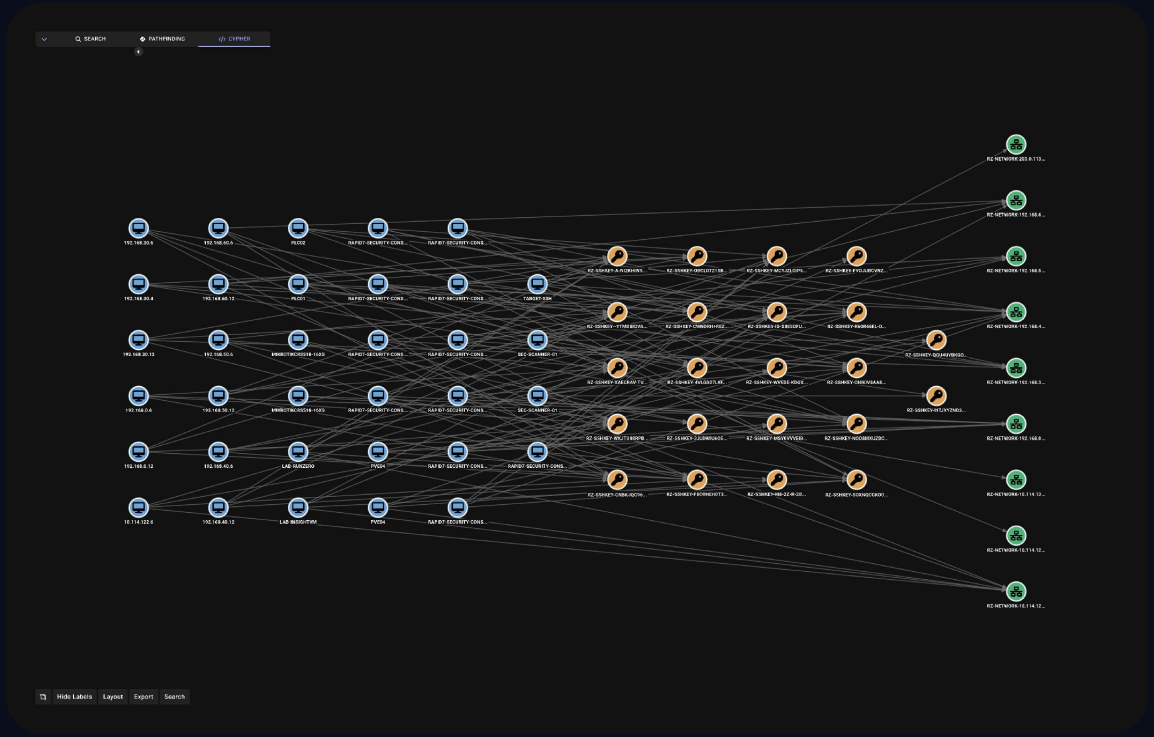
Nothing here is inferred from a diagram. Every edge is an observed address, an enumerated bus member, or a subnet membership.

BloodHound Community Edition, one Cypher query.

IN THE GRAPH

Matching assets across segments

HOST KEYS AS THE HUB, AND THE SEGMENTS EITHER SIDE



Middle column: SSH host keys. An asset either side of one key is running the same operating system image.

That has two causes and the key cannot separate them. One machine holding several addresses, or several machines cloned from one template. Both are present here.

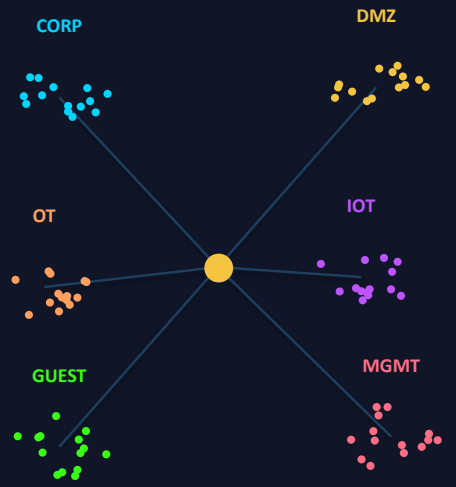
Either way the segments on the right are joined by something the network diagram does not draw.

BloodHound Community Edition, RZSSHKey as the join.

RESULT

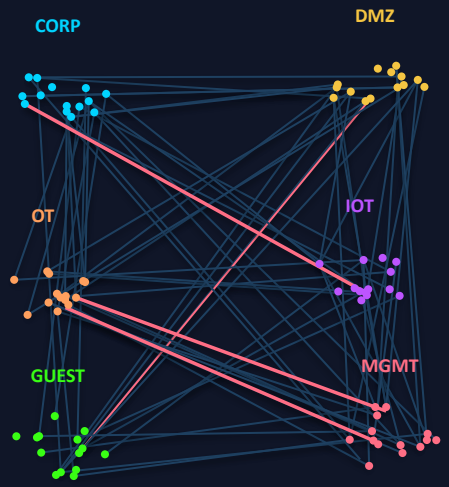
The network map vs reality

THE PUBLISHED NETWORK MAP



one chokepoint, every zone through it

YOUR ACTUAL MAP



every adjacency the hosts actually have

NEXT STEPS

Applying this to your network

Collect data you already have; correlate and query

1. **Probe the ports outside the default set**
   623/udp and 623/tcp, 47808, 44818, 502, 3671, 5900, 30718, 4800, 2362
2. **Correlate on identity**
   SSH host keys, TLS certificates, SMB GUIDs, IPMI system GUIDs, serial numbers
3. **Enumerate behind gateways before counting them**
   Walk the bus first, then record the population
4. **Segment per device**
   A shared management VLAN makes one compromised controller adjacent to all of them
5. **Define Open Graph kinds for your own data**
   Any domain expressible as nodes and edges inherits the pathfinding
6. **Measure your own management segments**
   The controller-to-host edge ships. What it cannot see is whether that segment is flat

runZero

Thank you!

TOOL	github.com/runZeroInc/runZeroHound
ICS DATA	hdm.io/data/ics.html
OPEN GRAPH	bloodhound.specterops.io/opengraph/overview
CONTACT	x /at/ hdm.io

PS. At Black Hat on Weds or DEFCON on Sat? Join for “Lights Out” on all the zero-day!