Wires & øutliərs

Exploring the Shadows within Enterprise Networks

HD Moore

Texas Cyber Summit 2023

runZero

Introduction

HD Moore

- Co-founder and CEO of runZero
- Previously founder & developer of Metasploit
- Recovering penetration tester

Get in touch!

- hdm/at/runZero.com
- @hdm@infosec.exchange
- https://hdm.io

Wires & outliers

- 5 years of continuous network discovery
  - External and internal scans of global networks
  - Passive monitoring of internal & darknets
  - API connections and file imports
- Two focus areas
  - Unexpected network links
  - Outlier analysis at scale
- Security impact

Part 1: Wires

Any system with more than one IP address or interface can undermine your security controls

Wires: Unexpected network links

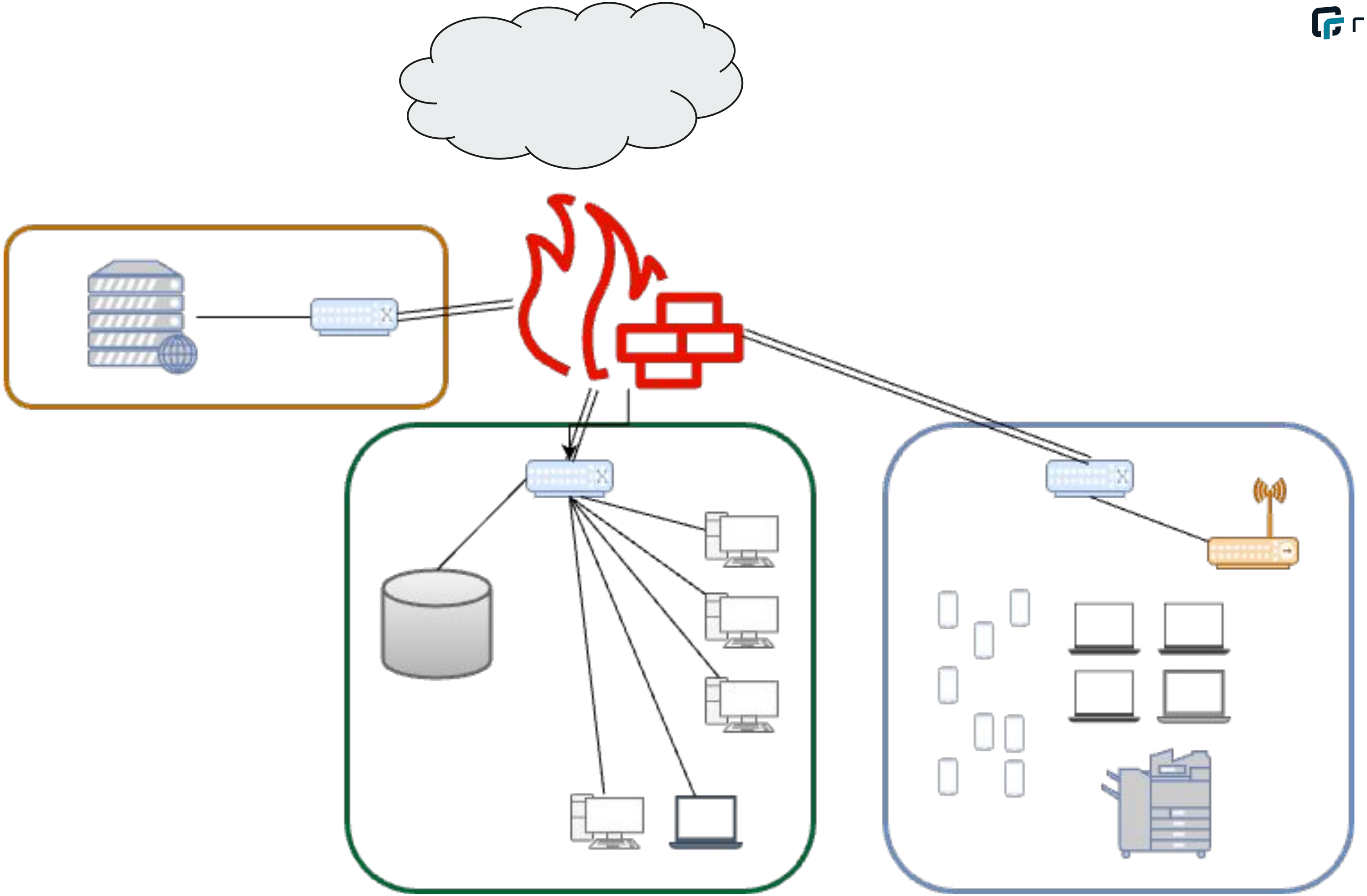
- Network diagrams rarely match reality
- Unexpected links undermine security
- Finding these reliably is difficult (!)
- A research focus for ~18+ years
    - 2005: ‘Rogue Network Links’ on full-disclosure
    - 2007: ‘Tactical Exploitation’ @ BlackHat/DEFCON
    - 2009: Metasploit: rogue_send/rogue_recv & netbios
    - 2018: github.com/hdm/nextnet
    - 2019+ runZero

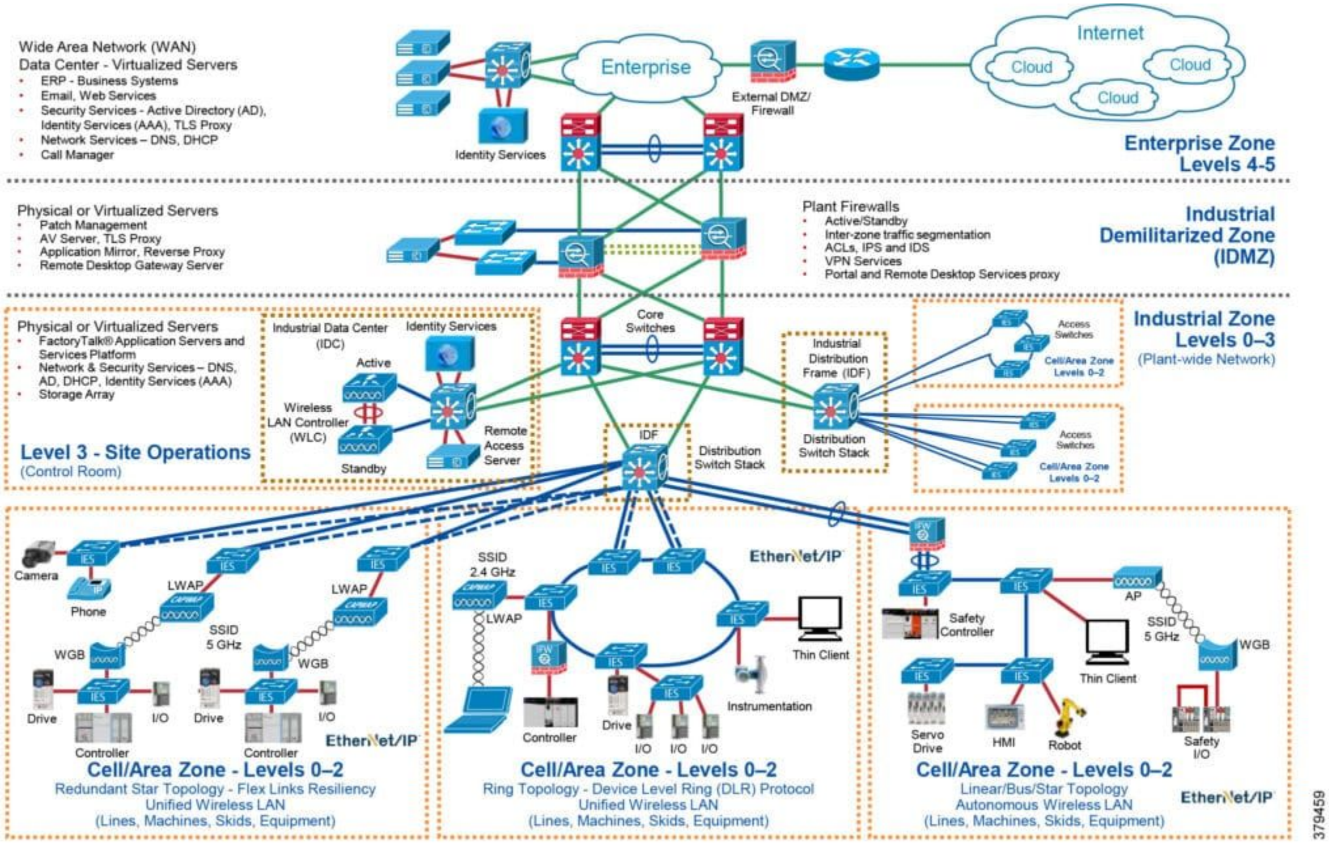
Unexpected links are common across layers

- **Multi-address node crossing security levels**
- HTTP load balancer desync and misconfigs
- Layer 4+ proxy exposure of app endpoints
- Layer 3 endpoint & routing exposures
- Layer 2 misconfigs
- Layer 1 PHY bugs

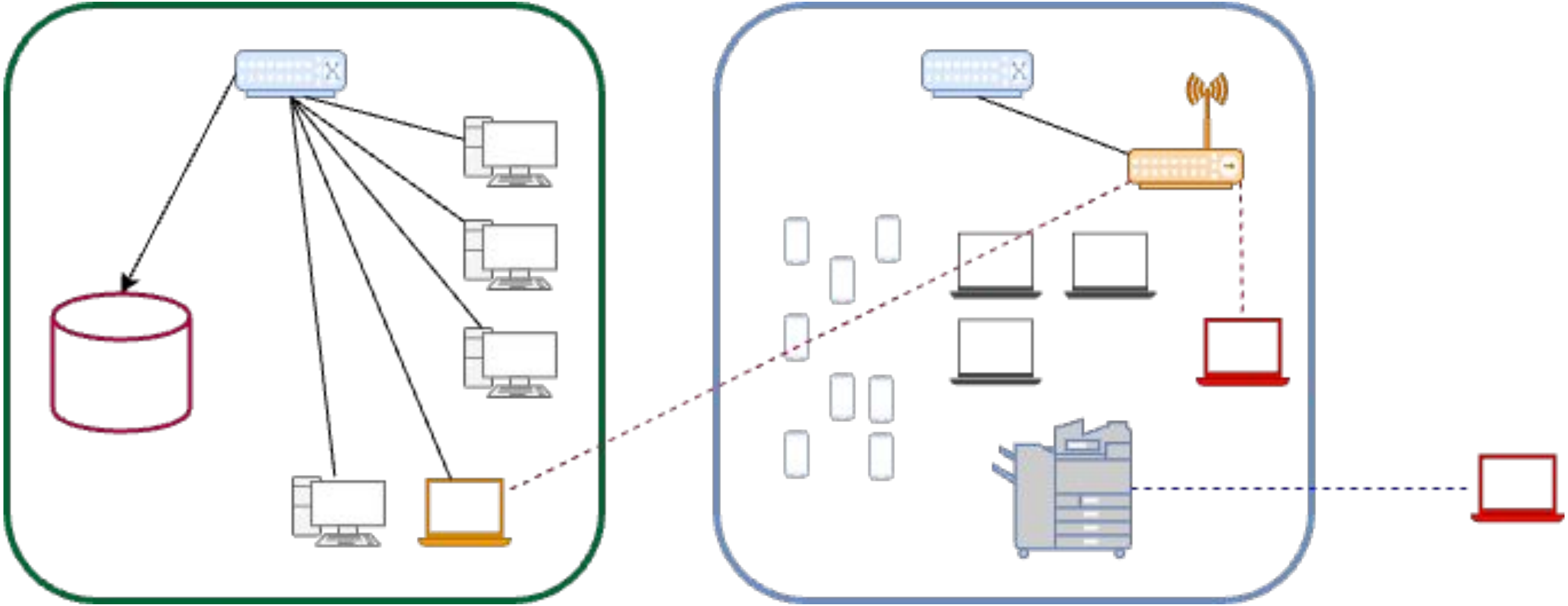
Multi-address nodes crossing security levels

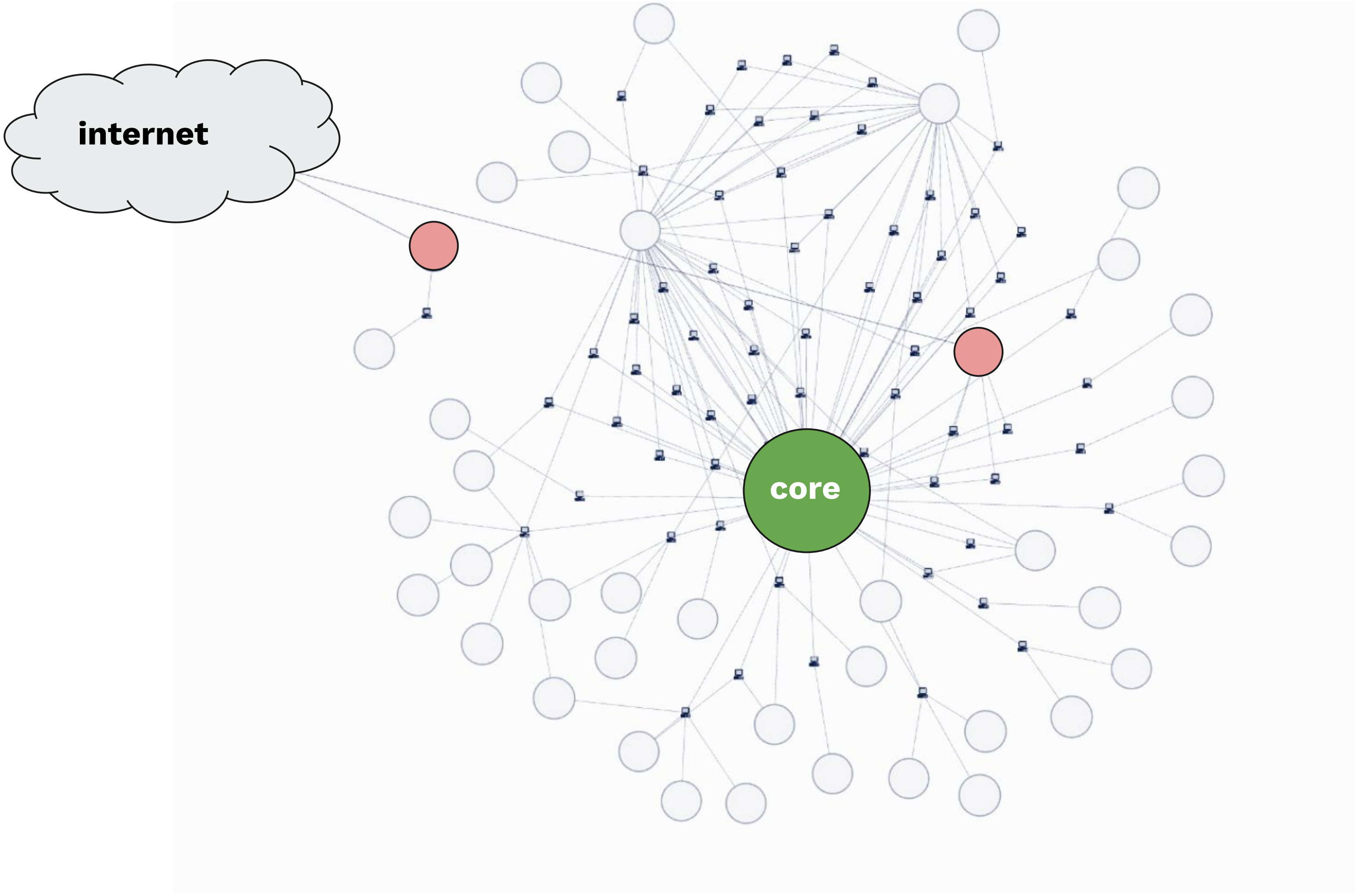
- System with more than one network connection
- Everywhere and rarely audited
    - Conference room equipment with WiFi & ethernet
    - Printers with WiFi/Bluetooth PAN & ethernet
    - Laptops with WiFi or Mobile & ethernet
    - Routers, switches, and VPN gateways
    - IT and network monitoring systems
    - VDI, Citrix, other jump boxes
    - IPv6 and IPv4

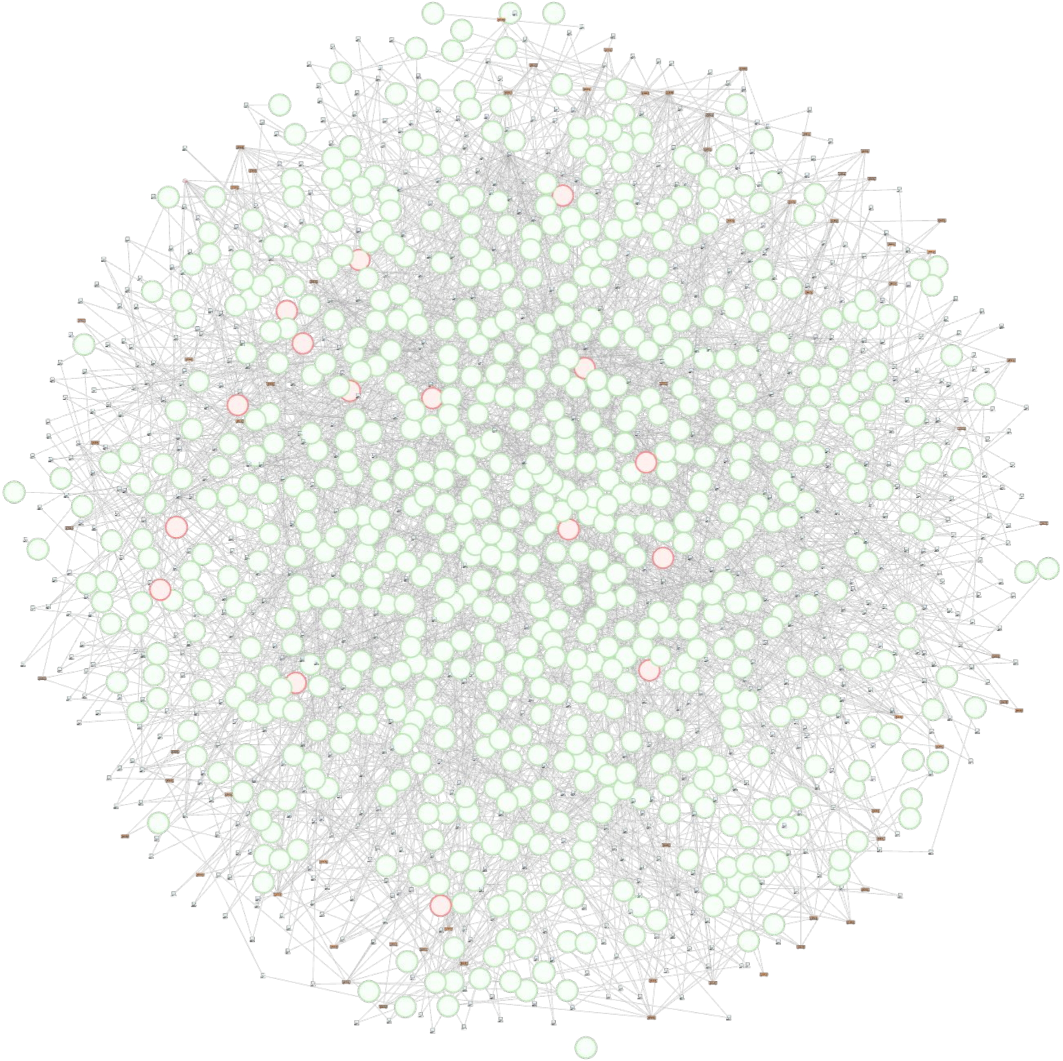


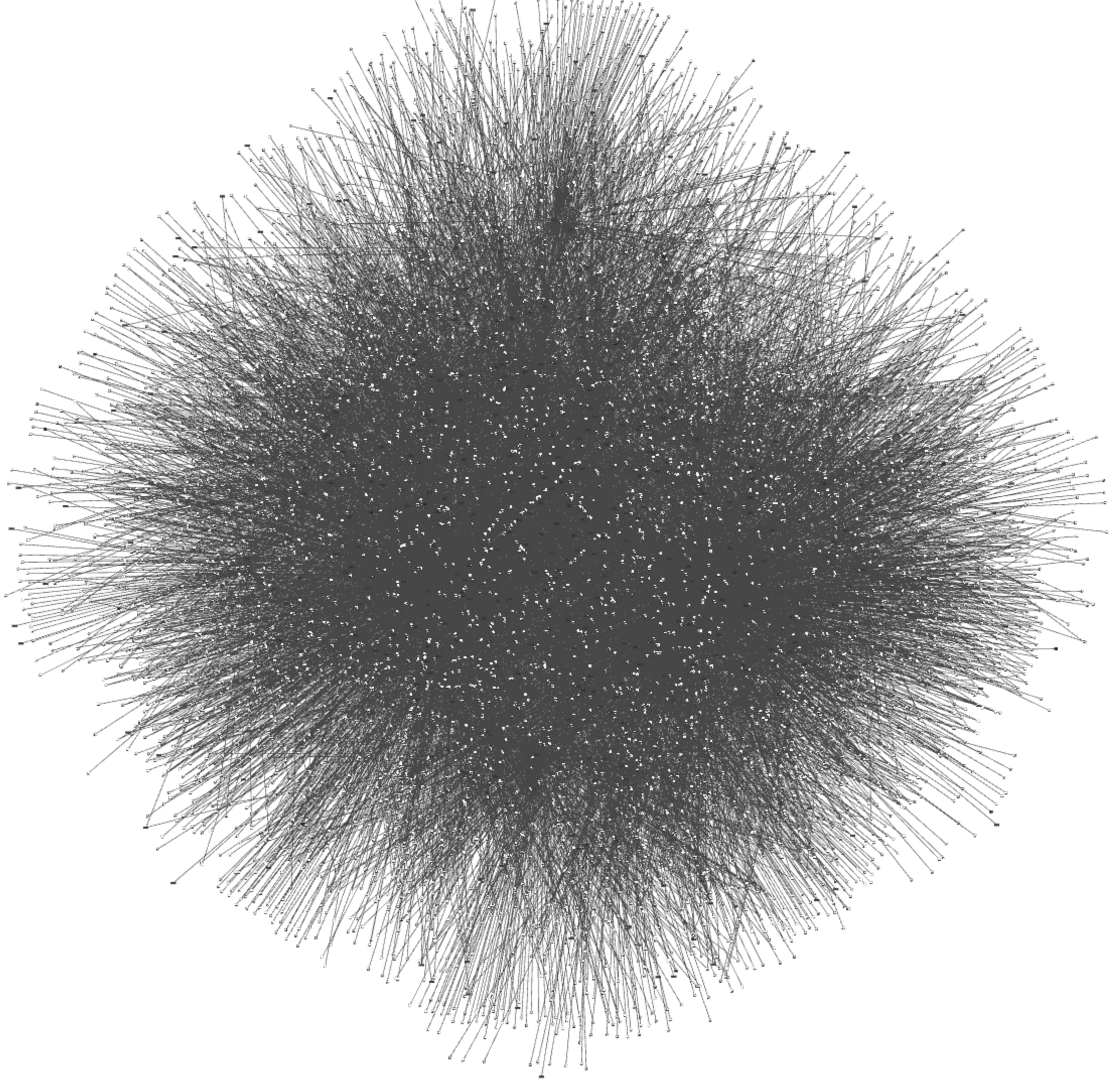

Wide Area Network (WAN)
Data Center - Virtualized Servers
ERP - Business Systems
Email, Web Services
Security Services - Active Directory (AD), Identity Services (AAA), TLS Proxy
Network Services – DNS, DHCP
Call Manager
Identity Services
Enterprise
External DMZ/ Firewall
Internet
Cloud
Cloud
Cloud
Enterprise Zone
Levels 4-5
Physical or Virtualized Servers
Patch Management
AV Server, TLS Proxy
Application Mirror, Reverse Proxy
Remote Desktop Gateway Server
Plant Firewalls
Active/Standby
Inter-zone traffic segmentation
ACLs, IPS and IDS
VPN Services
Portal and Remote Desktop Services proxy
Industrial Demilitarized Zone (IDMZ)
Physical or Virtualized Servers
FactoryTalk® Application Servers and Services Platform
Network & Security Services – DNS, AD, DHCP, Identity Services (AAA)
Storage Array
Industrial Data Center (IDC)
Identity Services
Active
Wireless LAN Controller (WLC)
Standby
Remote Access Server
Level 3 - Site Operations
(Control Room)
Core Switches
IDF
Distribution Switch Stack
Industrial Distribution Frame (IDF)
Distribution Switch Stack
Access Switches
Cell/Area Zone Levels 0–2
Access Switches
Cell/Area Zone Levels 0–2
Industrial Zone
Levels 0–3
(Plant-wide Network)
Camera
Phone
LWAP
LWAP
SSID 5 GHz
WGB
WGB
Drive
I/O
Drive
I/O
Controller
Controller
EtherNet/IP
Cell/Area Zone - Levels 0–2
Redundant Star Topology - Flex Links Resiliency
Unified Wireless LAN
(Lines, Machines, Skids, Equipment)
SSID 2.4 GHz
LWAP
EtherNet/IP
Thin Client
Instrumentation
Controller
Drive
I/O
I/O
I/O
Cell/Area Zone - Levels 0–2
Ring Topology - Device Level Ring (DLR) Protocol
Unified Wireless LAN
(Lines, Machines, Skids, Equipment)
Safety Controller
AP
SSID 5 GHz
WGB
Thin Client
Servo Drive
HMI
Robot
Safety I/O
Cell/Area Zone - Levels 0–2
Linear/Bus/Star Topology
Autonomous Wireless LAN
(Lines, Machines, Skids, Equipment)
EtherNet/IP
379459

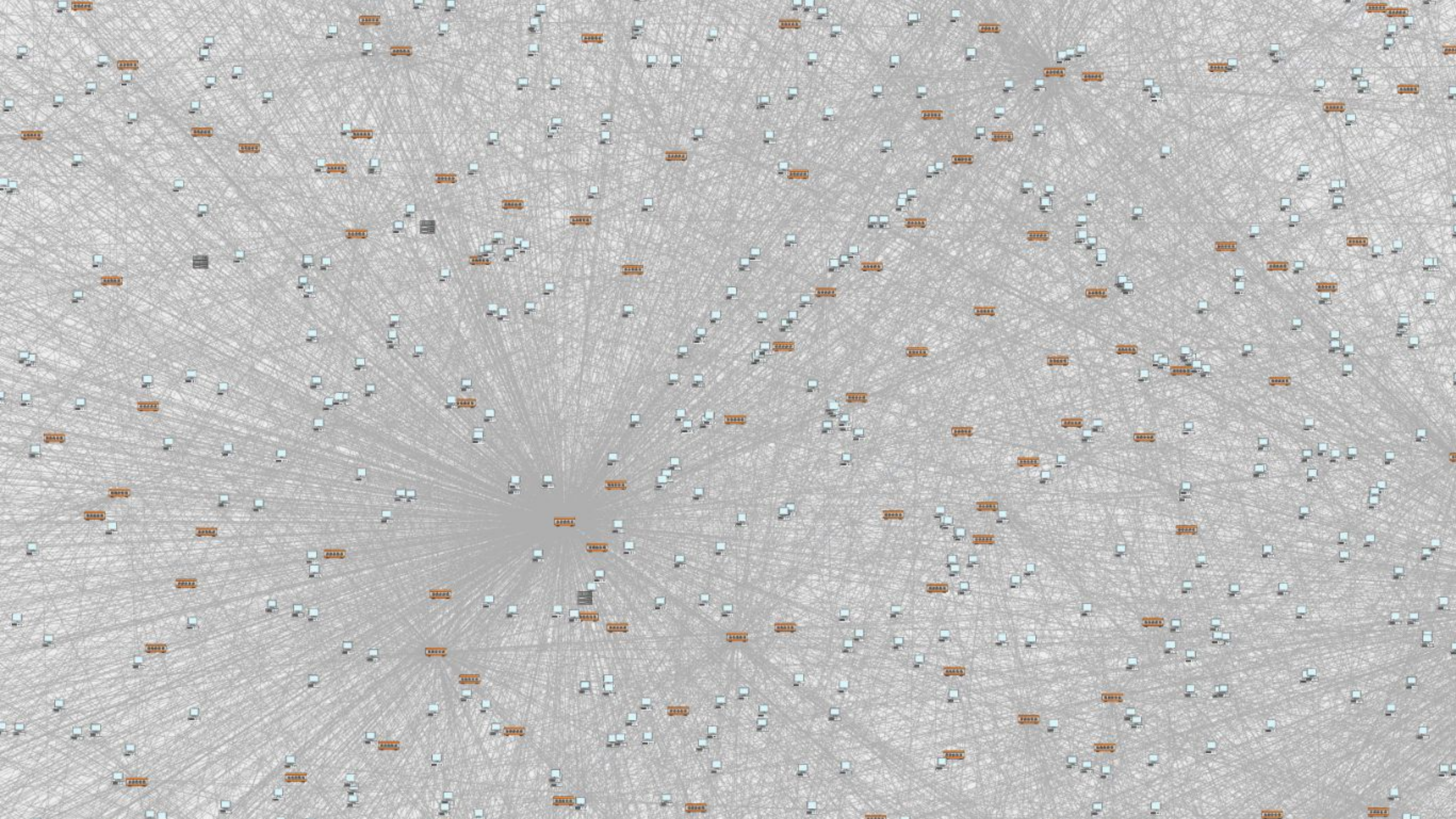
https://www.goagilix.com/industrial-network-design-best-practices/

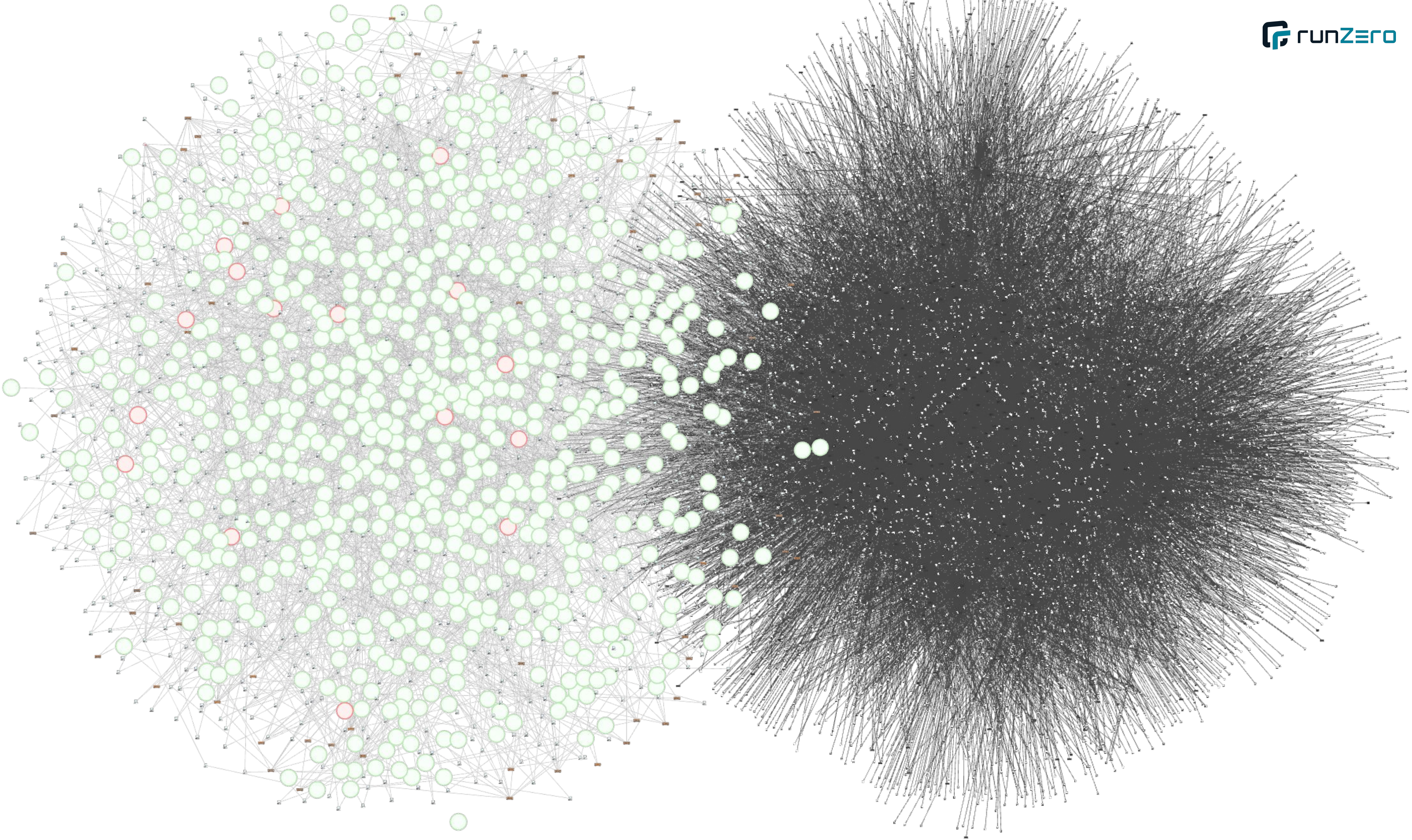



internet
core









Example: Solarwinds Orion on Windows

- Centrally deployed for network monitoring
- Devices allow SNMP + SSH from Solarwinds
- Solarwinds stores creds in SQL + DPStore

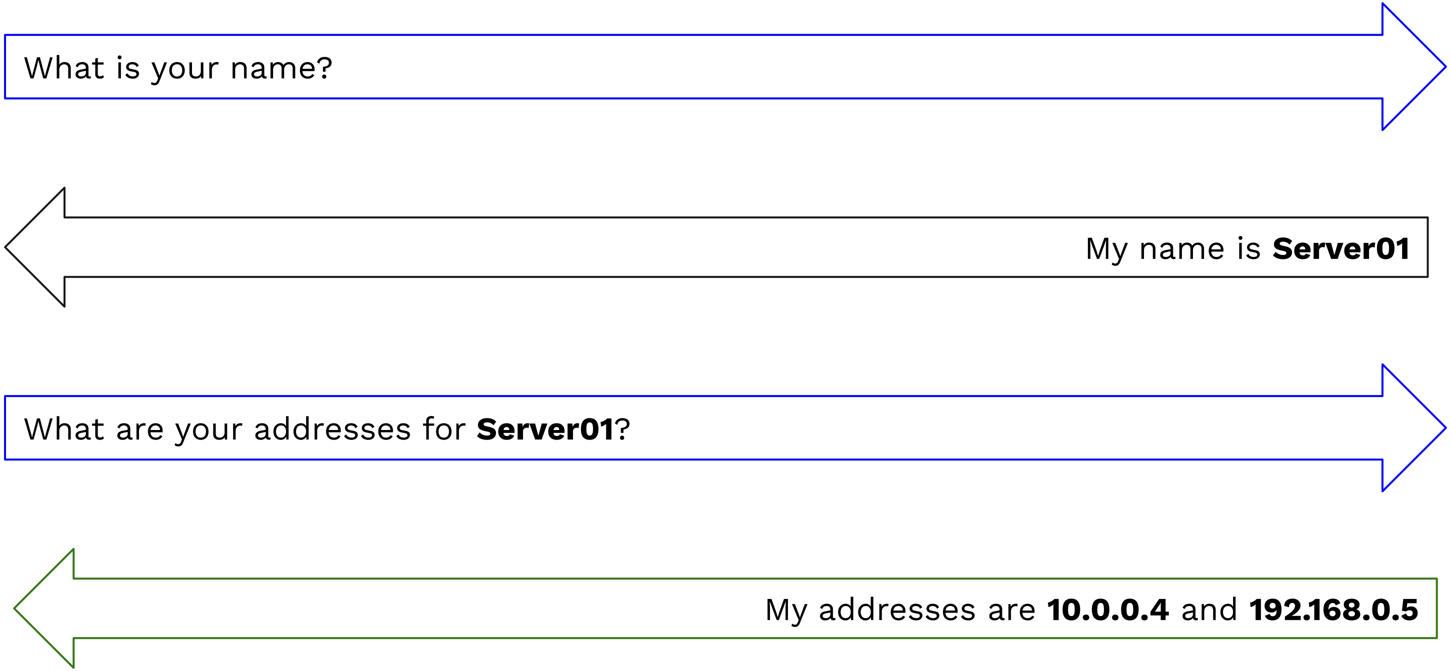
Result

- Full de-segmentation + compromise

Detection

- Two-pass NB scan (137/udp) (metasploit/nextnet)

Two-pass NetBIOS (137/udp) discovery



What is your name?

My name is **Server01**

What are your addresses for **Server01**?

My addresses are **10.0.0.4** and **192.168.0.5**

Example: Mobile LTE in executive laptop

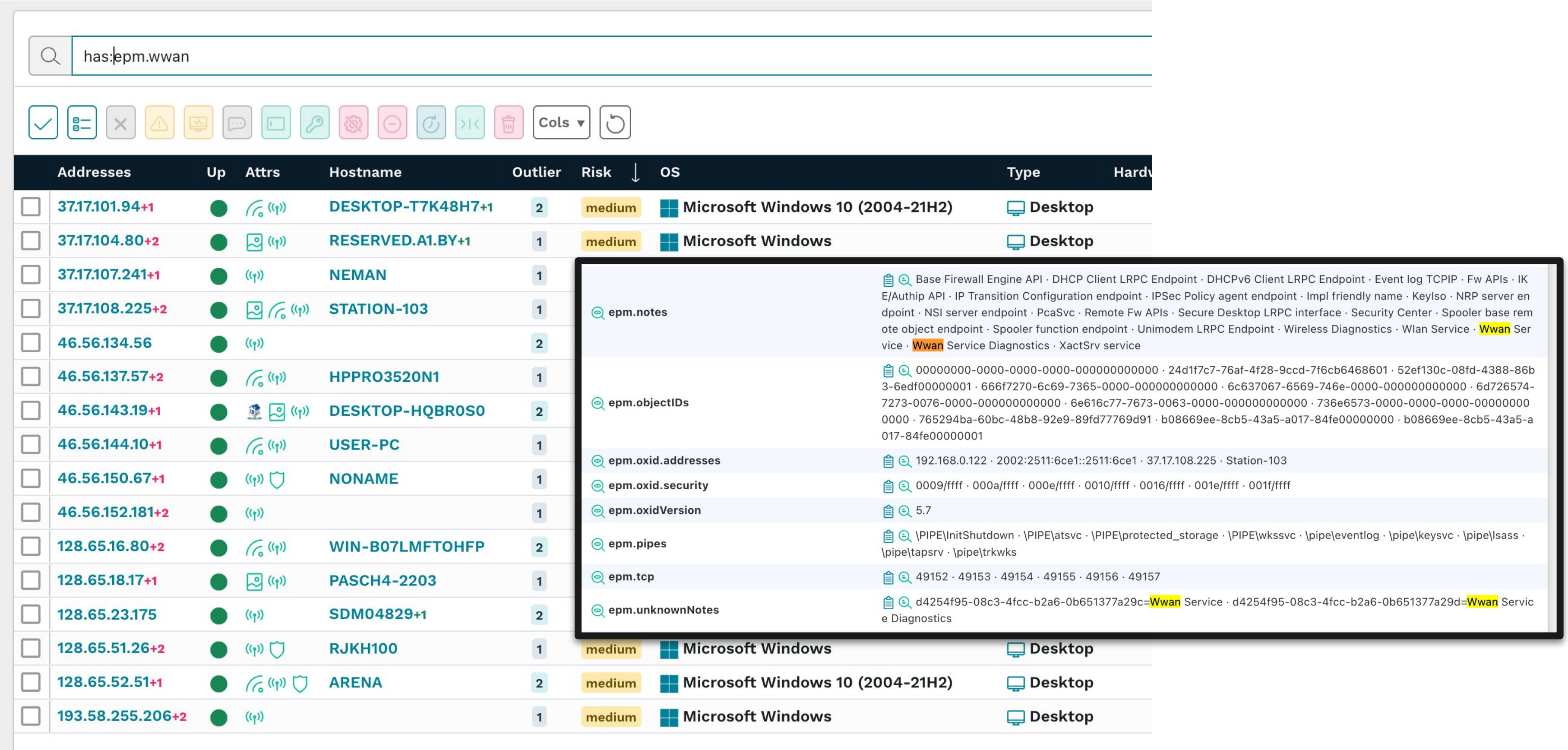
- Semi-frequently exposed RDP to the internet
- Exposure depended on the provider
- IT didn’t realize it was enabled

Result

- Caught before compromise due to weak local user

Detection

- DCERPC EPM internal scan + FP (nmap/runZero)



has:epm.wwan

□	Addresses	Up	Attrs	Hostname	Outlier	Risk	OS	Type	Hardw
□	37.17.101.94+1	●		DESKTOP-T7K48H7+1	2	medium	Microsoft Windows 10 (2004-21H2)	Desktop	
□	37.17.104.80+2	●		RESERVED.A1.BY+1	1	medium	Microsoft Windows	Desktop	
□	37.17.107.241+1	●		NEMAN	1				
□	37.17.108.225+2	●		STATION-103	1				
□	46.56.134.56	●			2				
□	46.56.137.57+2	●		HPPRO3520N1	1				
□	46.56.143.19+1	●		DESKTOP-HQBR0S0	2				
□	46.56.144.10+1	●		USER-PC	1				
□	46.56.150.67+1	●		NONAME	1				
□	46.56.152.181+2	●			1				
□	128.65.16.80+2	●		WIN-B07LMFTOHFP	2				
□	128.65.18.17+1	●		PASCH4-2203	1				
□	128.65.23.175	●		SDM04829+1	2				
□	128.65.51.26+2	●		RJKH100	1	medium	Microsoft Windows	Desktop	
□	128.65.52.51+1	●		ARENA	2	medium	Microsoft Windows 10 (2004-21H2)	Desktop	
□	193.58.255.206+2	●			1	medium	Microsoft Windows	Desktop	

epm.notes	Base Firewall Engine API · DHCP Client LRPC Endpoint · DHCPv6 Client LRPC Endpoint · Event log TCPIP · Fw APIs · IKE/Authip API · IP Transition Configuration endpoint · IPSec Policy agent endpoint · Impl friendly name · KeyIso · NRP server endpoint · NSI server endpoint · PcaSvc · Remote Fw APIs · Secure Desktop LRPC interface · Security Center · Spooler base remote object endpoint · Spooler function endpoint · Unimodem LRPC Endpoint · Wireless Diagnostics · Wlan Service · Wwan Service · Wwan Service Diagnostics · XactSrv service
epm.objectIDs	00000000-0000-0000-0000-000000000000 · 24d1f7c7-76af-4f28-9ccd-7f6cb6468601 · 52ef130c-08fd-4388-86b3-6edf00000001 · 666f7270-6c69-7365-0000-000000000000 · 6c637067-6569-746e-0000-000000000000 · 6d726574-7273-0076-0000-000000000000 · 6e616c77-7673-0063-0000-000000000000 · 736e6573-0000-0000-0000-000000000000 · 765294ba-60bc-48b8-92e9-89fd77769d91 · b08669ee-8cb5-43a5-a017-84fe00000000 · b08669ee-8cb5-43a5-a017-84fe00000001
epm.oxid.addresses	192.168.0.122 · 2002:2511:6ce1::2511:6ce1 · 37.17.108.225 · Station-103
epm.oxid.security	0009/ffff · 000a/ffff · 000e/ffff · 0010/ffff · 0016/ffff · 001e/ffff · 001f/ffff
epm.oxidVersion	5.7
epm.pipes	\PIPE\InitShutdown · \PIPE\atsvc · \PIPE\protected_storage · \PIPE\wkssvc · \pipe\eventlog · \pipe\keysvc · \pipe\lsass · \pipe\tapsrv · \pipe\trkwks
epm.tcp	49152 · 49153 · 49154 · 49155 · 49156 · 49157
epm.unknownNotes	d4254f95-08c3-4fcc-b2a6-0b651377a29c=Wwan Service · d4254f95-08c3-4fcc-b2a6-0b651377a29d=Wwan Service Diagnostics

Example: IPv6-only exposures (link-local)

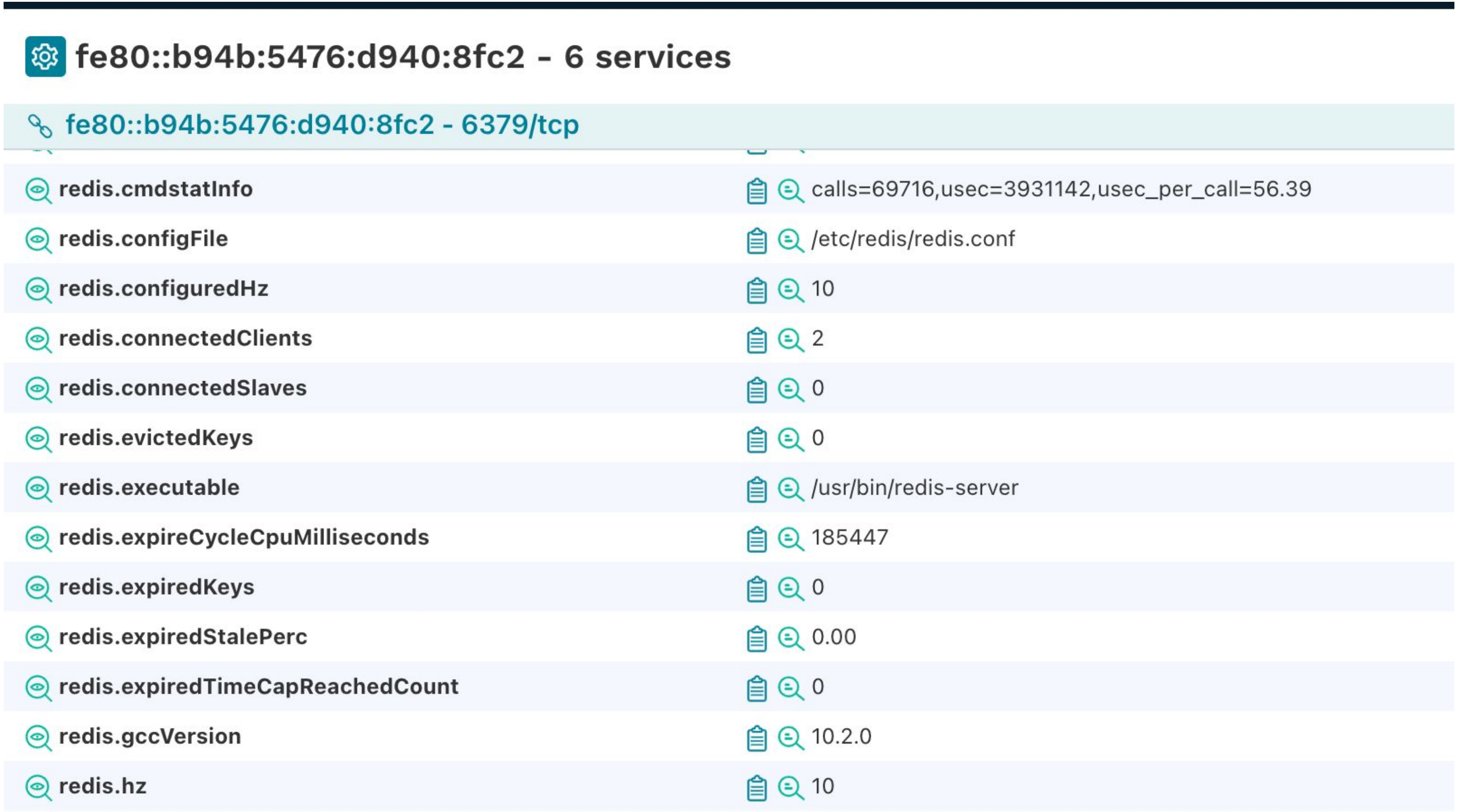
- Still a common problem with appliances/devices
- VoIP server exposed redis and mongoDB on IPv6

Result

- Dumped all data from both databases (no auth)

Detection

- FF02::1 UDP ping + TCP SYN scan (nmap/runZero)



fe80::b94b:5476:d940:8fc2 - 6 services

fe80::b94b:5476:d940:8fc2 - 6379/tcp

redis.cmdstatInfo	calls=69716,usec=3931142,usec_per_call=56.39
redis.configFile	/etc/redis/redis.conf
redis.configuredHz	10
redis.connectedClients	2
redis.connectedSlaves	0
redis.evictedKeys	0
redis.executable	/usr/bin/redis-server
redis.expireCycleCpuMilliseconds	185447
redis.expiredKeys	0
redis.expiredStalePerc	0.00
redis.expiredTimeCapReachedCount	0
redis.gccVersion	10.2.0
redis.hz	10

Example: IPv6-only exposures (global)

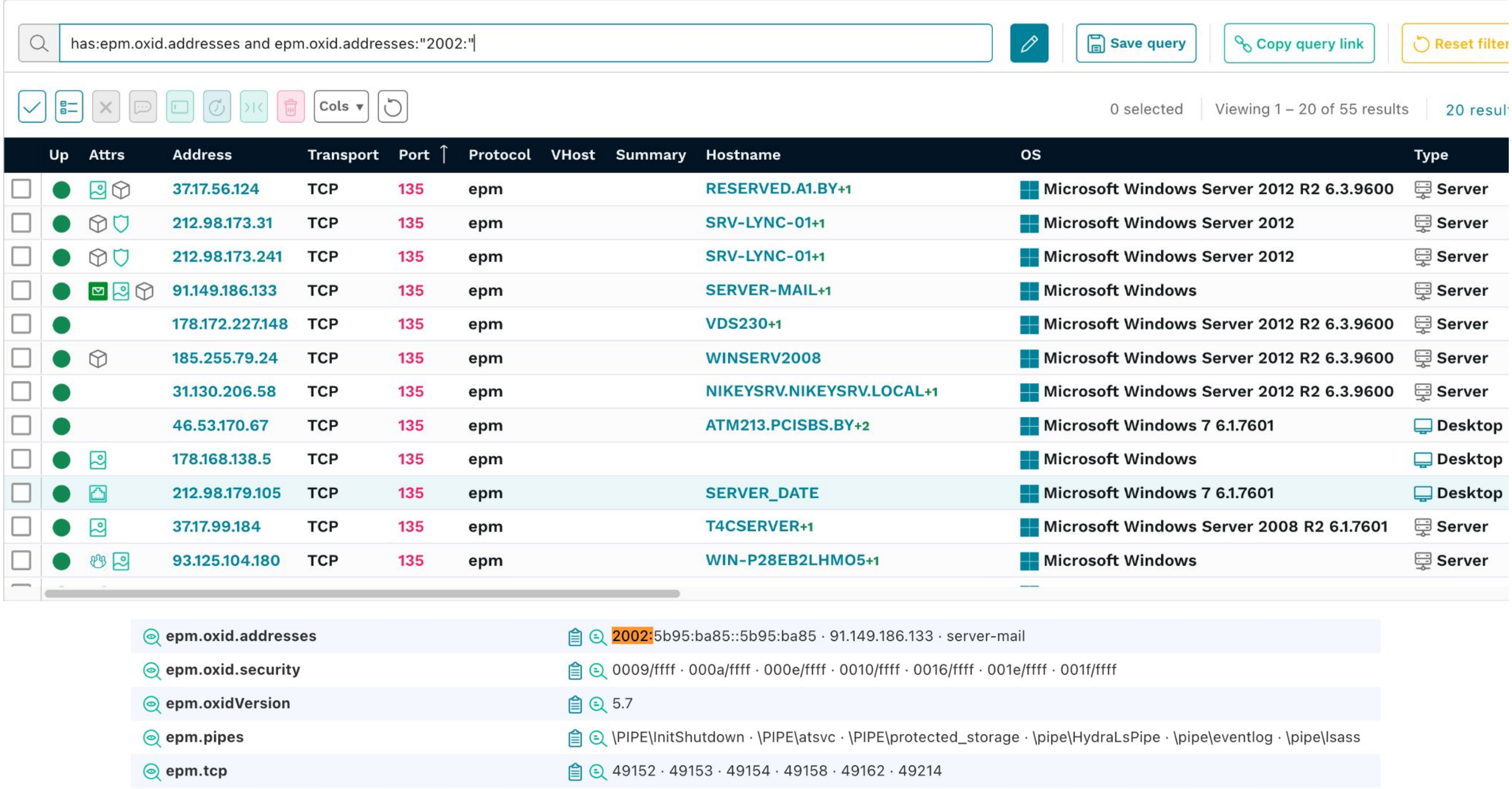
- ISP anycast 6to4 gateways lead to surprises
- IPv6 GW as 192.88.99.1 can auto-allocate 6to4
- Hosts reachable via the 2002::/16 IPv6 subnet

Result

- External notification of exposed SMB/RDP

Detection

- DCERPC Oxid2Resolver scan (impacket/runZero)



```
has:epm.oxid.addresses and epm.oxid.addresses:"2002:"
```

	Up	Attrs	Address	Transport	Port ↑	Protocol	VHost	Summary	Hostname	OS	Type
□	●		37.17.56.124	TCP	135	epm			RESERVED.A1.BY+1	Microsoft Windows Server 2012 R2 6.3.9600	Server
□	●		212.98.173.31	TCP	135	epm			SRV-LYNC-01+1	Microsoft Windows Server 2012	Server
□	●		212.98.173.241	TCP	135	epm			SRV-LYNC-01+1	Microsoft Windows Server 2012	Server
□	●		91.149.186.133	TCP	135	epm			SERVER-MAIL+1	Microsoft Windows	Server
□	●		178.172.227.148	TCP	135	epm			VDS230+1	Microsoft Windows Server 2012 R2 6.3.9600	Server
□	●		185.255.79.24	TCP	135	epm			WINSERV2008	Microsoft Windows Server 2012 R2 6.3.9600	Server
□	●		31.130.206.58	TCP	135	epm			NIKEYSRV.NIKEYSRV.LOCAL+1	Microsoft Windows Server 2012 R2 6.3.9600	Server
□	●		46.53.170.67	TCP	135	epm			ATM213.PCISBS.BY+2	Microsoft Windows 7 6.1.7601	Desktop
□	●		178.168.138.5	TCP	135	epm				Microsoft Windows	Desktop
□	●		212.98.179.105	TCP	135	epm			SERVER_DATE	Microsoft Windows 7 6.1.7601	Desktop
□	●		37.17.99.184	TCP	135	epm			T4CSERVER+1	Microsoft Windows Server 2008 R2 6.1.7601	Server
□	●		93.125.104.180	TCP	135	epm			WIN-P28EB2LHMO5+1	Microsoft Windows	Server

epm.oxid.addresses	2002:5b95:ba85::5b95:ba85 · 91.149.186.133 · server-mail
epm.oxid.security	0009/ffff · 000a/ffff · 000e/ffff · 0010/ffff · 0016/ffff · 001e/ffff · 001f/ffff
epm.oxidVersion	5.7
epm.pipes	\PIPE\InitShutdown · \PIPE\atsvc · \PIPE\protected_storage · \pipe\HydraLsPipe · \pipe\eventlog · \pipe\lsass
epm.tcp	49152 · 49153 · 49154 · 49158 · 49162 · 49214

Tragically undervalued by security teams

- A strangely underappreciated attack vector
- A graveyard of commercialization attempts
- Less exciting than RCE vulnerabilities
- Still a recurring weak point
- Difficult to assess
- Worse in 2023

Detecting multi-address nodes at scale

- Actively scan the network for secondary links
    - Extract encoded fields that expose addresses
    - Send tagged packets, receive from other address
    - Query SNMP devices to leak neighbor info
    - Use IPv6 to identify IPv4 and vice-versa

- Scan/Sniff everything and compare unique attrs
    - Match unique assets across networks

IP forwarding is not just for routers

- System receives a packet meant for another IP
  - Some systems forward by default
  - Bypasses layer-2 controls

- Common examples
  - Linux laptops/servers running containers
  - Many printers across all vendors

- Identify these by sending low TTL packets

IP reflection is still effective after 18 years

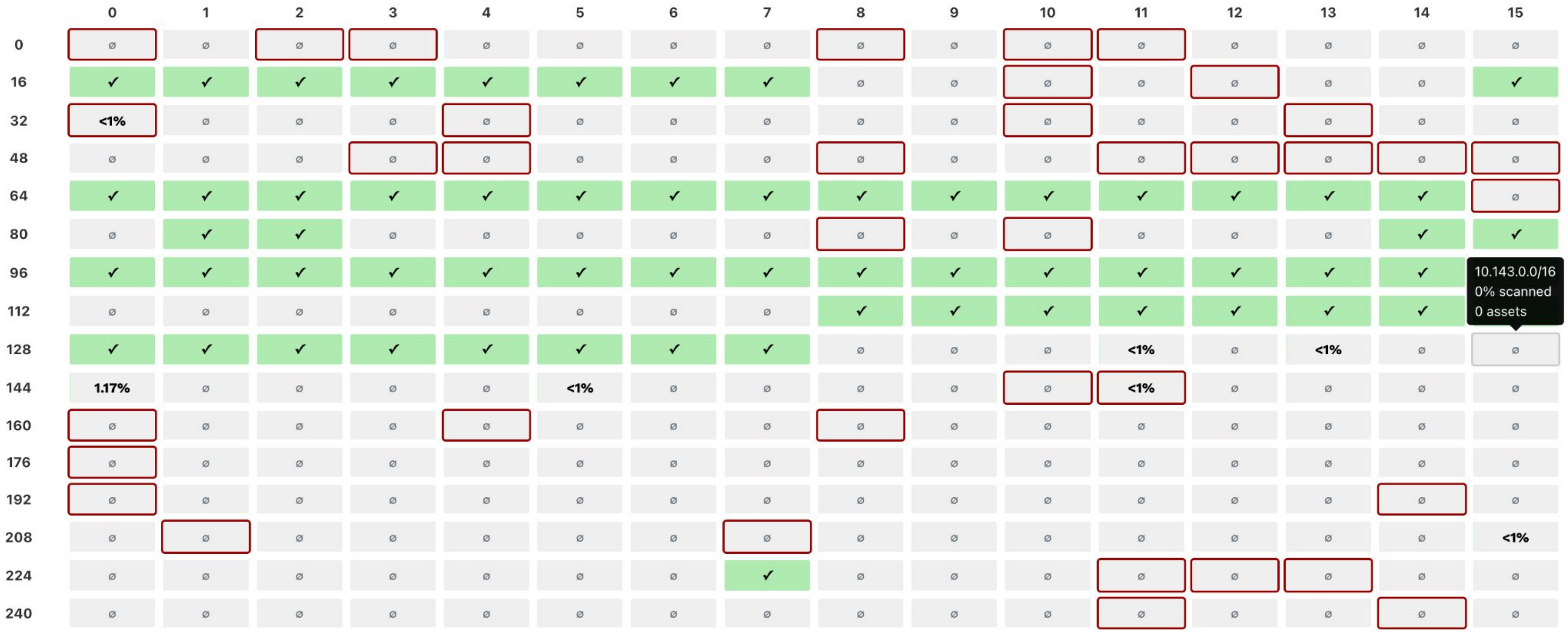
- Send a ping that triggers a response
  - Send this from a public IP address
  - Send this to every internal IP address

- Multi-homed machines reply via default route
  - Tricky since not all replies go through NAT
  - Requires an internet-facing monitor

Making sense of the data

- What nodes are in the sensitive networks?
- Do any nodes bridge security levels?
- What controls segmentation?
- Strange, but mostly harmless
  - Use of the N.N.N.N IPs for router p2p links
  - IPs in the non-RFC 1918 ranges (CGNAT, Test)
  - Static IPs shared across many laptops (VoIP)

Found a new network? Keep hunting!



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	⌀	⌀	⌀	⌀	⌀	⌀	⌀	⌀	⌀	⌀	⌀	⌀	⌀	⌀	⌀	⌀
16	✓	✓	✓	✓	✓	✓	✓	✓	⌀	⌀	⌀	⌀	⌀	⌀	⌀	✓
32	<1%	⌀	⌀	⌀	⌀	⌀	⌀	⌀	⌀	⌀	⌀	⌀	⌀	⌀	⌀	⌀
48	⌀	⌀	⌀	⌀	⌀	⌀	⌀	⌀	⌀	⌀	⌀	⌀	⌀	⌀	⌀	⌀
64	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	⌀
80	⌀	✓	✓	⌀	⌀	⌀	⌀	⌀	⌀	⌀	⌀	⌀	⌀	⌀	✓	✓
96	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
112	⌀	⌀	⌀	⌀	⌀	⌀	⌀	⌀	✓	✓	✓	✓	✓	✓	✓	
128	✓	✓	✓	✓	✓	✓	✓	✓	⌀	⌀	⌀	<1%	⌀	<1%	⌀	⌀
144	1.17%	⌀	⌀	⌀	⌀	<1%	⌀	⌀	⌀	⌀	⌀	<1%	⌀	⌀	⌀	⌀
160	⌀	⌀	⌀	⌀	⌀	⌀	⌀	⌀	⌀	⌀	⌀	⌀	⌀	⌀	⌀	⌀
176	⌀	⌀	⌀	⌀	⌀	⌀	⌀	⌀	⌀	⌀	⌀	⌀	⌀	⌀	⌀	⌀
192	⌀	⌀	⌀	⌀	⌀	⌀	⌀	⌀	⌀	⌀	⌀	⌀	⌀	⌀	⌀	⌀
208	⌀	⌀	⌀	⌀	⌀	⌀	⌀	⌀	⌀	⌀	⌀	⌀	⌀	⌀	⌀	<1%
224	⌀	⌀	⌀	⌀	⌀	⌀	⌀	✓	⌀	⌀	⌀	⌀	⌀	⌀	⌀	⌀
240	⌀	⌀	⌀	⌀	⌀	⌀	⌀	⌀	⌀	⌀	⌀	⌀	⌀	⌀	⌀	⌀

10.143.0.0/16
0% scanned
0 assets

Part 2: Outliers

Any system that looks weird is a potential security risk and is worth investigating

Outliers: Security use cases are tricky

- Anomaly detection is rediscovered constantly
- Tough to depend on for many reasons
  - Behavior baselines continuously change
  - Attackers can push/pull the baseline
  - Sometimes the weird is normal
  - Noisy when things go wrong
  - Learning can take too long

- New ML can help, but same core issue

Two ways of identifying bad things fast

- Things that should NOT be shared, but are
  - TLS fingerprints on unrelated services
  - SSH host key fingerprints
  - TCP sequence numbers

- Things that should be shared, but are NOT
  - Operating system name & version
  - Installed software name & version
  - Service ports for SSH & RDP
  - TCP window size

Find the unexpectedly common things

- GROUP BY and COUNT(*) a dynamic field
- Any count over 1 is typically a problem
- Anything encryption related is dodgy
- Uncovers unique host identifiers
    - Serial numbers in exposed fields
    - Hostnames can be mostly-unique
    - Identify multi-address nodes!

Example: SSH host keys

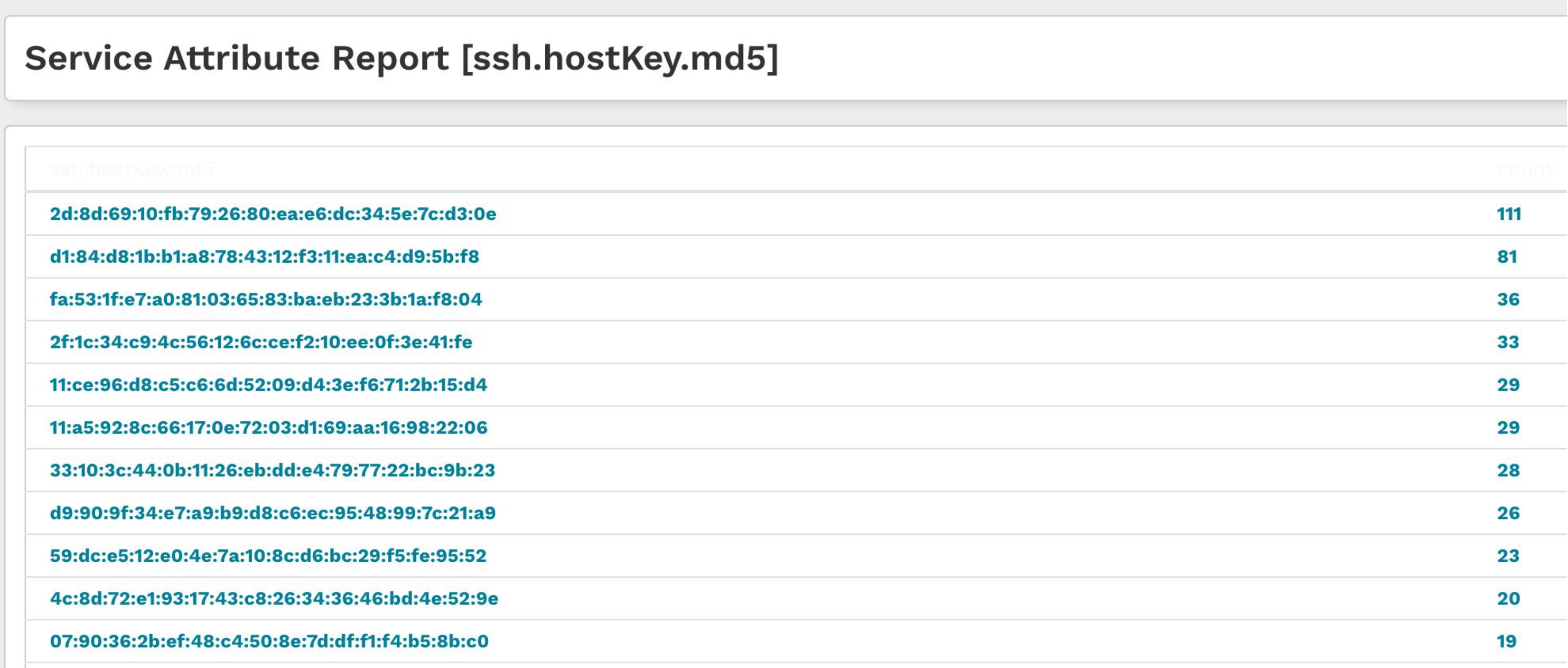
- SSH host keys should be unique per asset
- Duplication leads to weaker security
- Pop any node, now MITM any other

Result

- Locate VMs that share SSH encryption keys

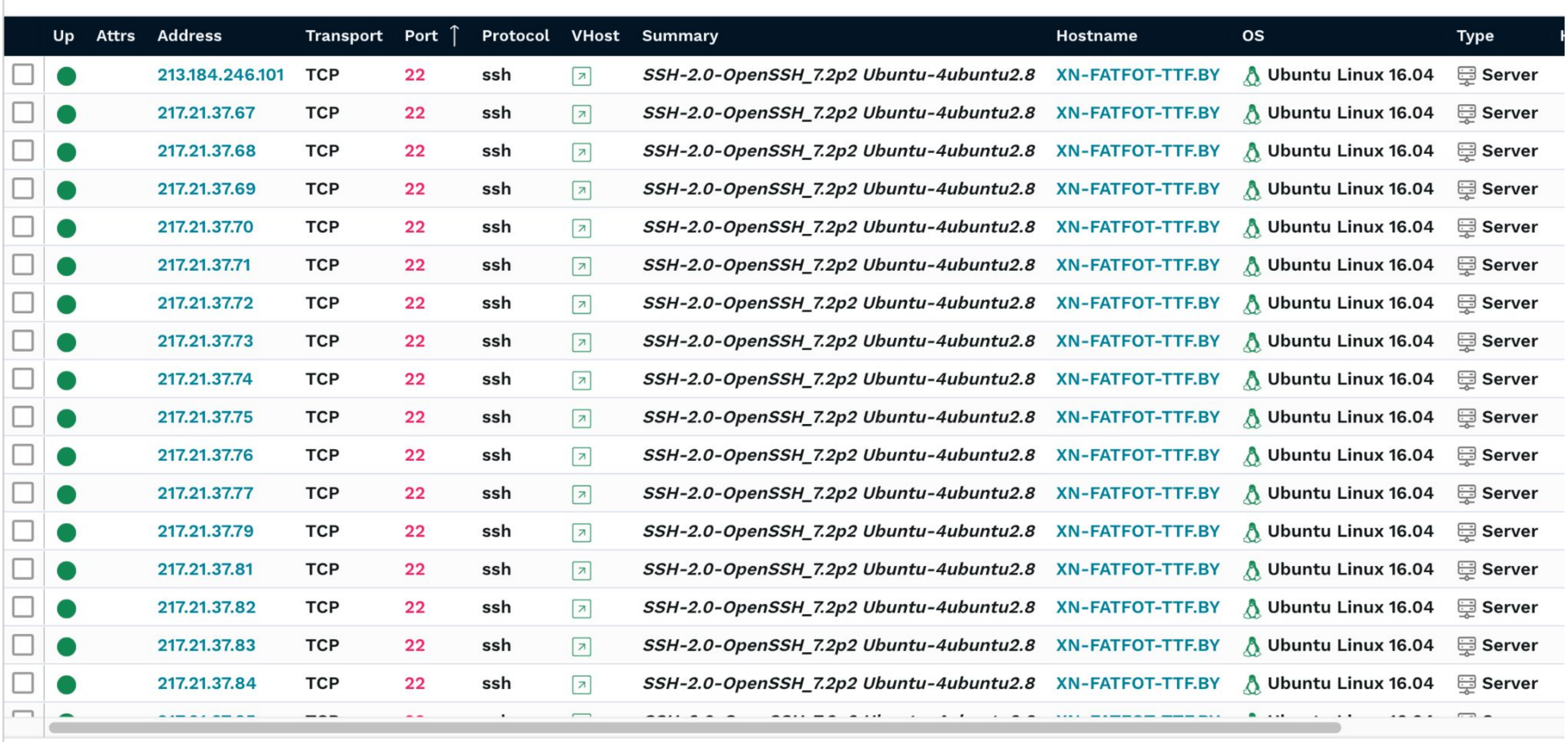
Detection

- SSH scans (ssh-keyscan/nmap/runzero)

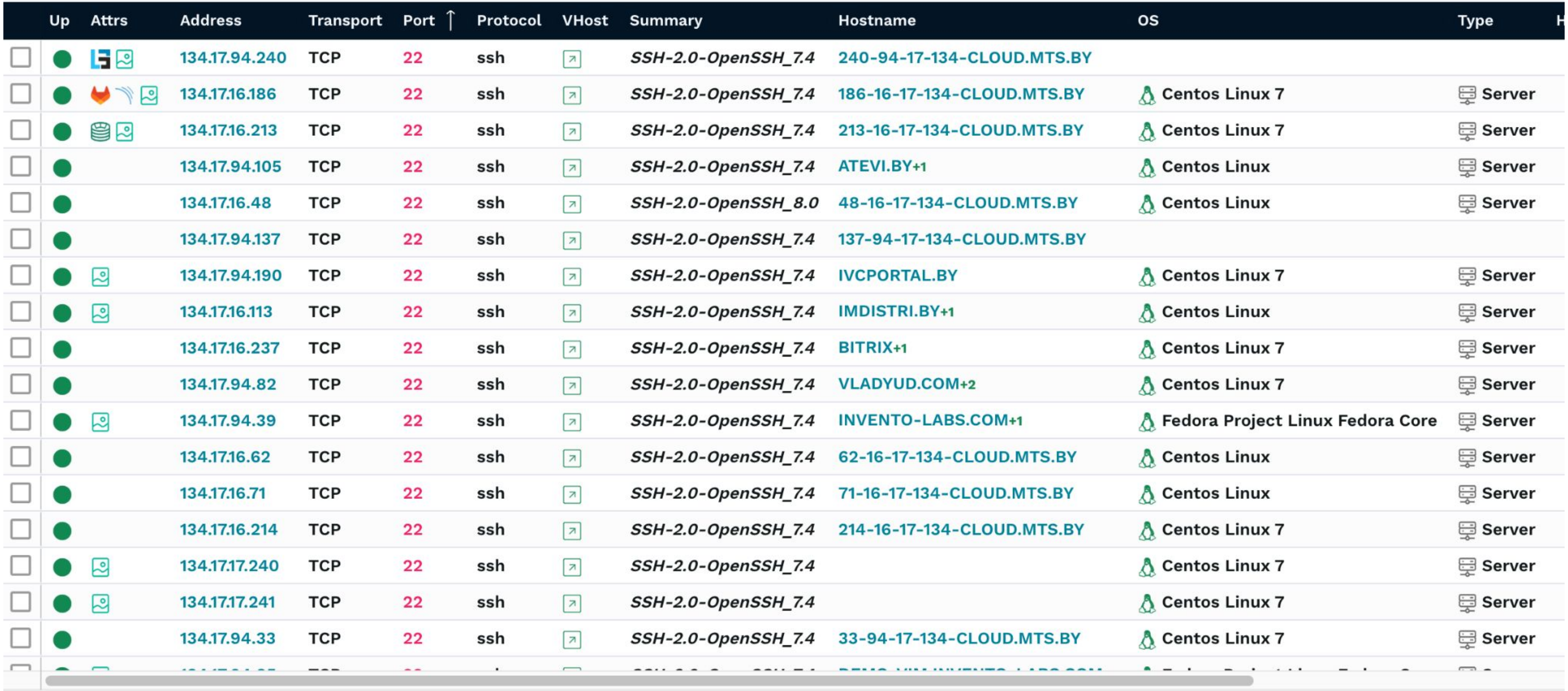


Service Attribute Report [ssh.hostKey.md5]

ssh.hostKey.md5	count
2d:8d:69:10:fb:79:26:80:ea:e6:dc:34:5e:7c:d3:0e	111
d1:84:d8:1b:b1:a8:78:43:12:f3:11:ea:c4:d9:5b:f8	81
fa:53:1f:e7:a0:81:03:65:83:ba:eb:23:3b:1a:f8:04	36
2f:1c:34:c9:4c:56:12:6c:ce:f2:10:ee:0f:3e:41:fe	33
11:ce:96:d8:c5:c6:6d:52:09:d4:3e:f6:71:2b:15:d4	29
11:a5:92:8c:66:17:0e:72:03:d1:69:aa:16:98:22:06	29
33:10:3c:44:0b:11:26:eb:dd:e4:79:77:22:bc:9b:23	28
d9:90:9f:34:e7:a9:b9:d8:c6:ec:95:48:99:7c:21:a9	26
59:dc:e5:12:e0:4e:7a:10:8c:d6:bc:29:f5:fe:95:52	23
4c:8d:72:e1:93:17:43:c8:26:34:36:46:bd:4e:52:9e	20
07:90:36:2b:ef:48:c4:50:8e:7d:df:f1:f4:b5:8b:c0	19



	Up	Attrs	Address	Transport	Port ↑	Protocol	VHost	Summary	Hostname	OS	Type
□	●		213.184.246.101	TCP	22	ssh	↗	*SSH-2.0-OpenSSH_7.2p2 Ubuntu-4ubuntu2.8*	XN-FATFOT-TTF.BY	Ubuntu Linux 16.04	Server
□	●		217.21.37.67	TCP	22	ssh	↗	*SSH-2.0-OpenSSH_7.2p2 Ubuntu-4ubuntu2.8*	XN-FATFOT-TTF.BY	Ubuntu Linux 16.04	Server
□	●		217.21.37.68	TCP	22	ssh	↗	*SSH-2.0-OpenSSH_7.2p2 Ubuntu-4ubuntu2.8*	XN-FATFOT-TTF.BY	Ubuntu Linux 16.04	Server
□	●		217.21.37.69	TCP	22	ssh	↗	*SSH-2.0-OpenSSH_7.2p2 Ubuntu-4ubuntu2.8*	XN-FATFOT-TTF.BY	Ubuntu Linux 16.04	Server
□	●		217.21.37.70	TCP	22	ssh	↗	*SSH-2.0-OpenSSH_7.2p2 Ubuntu-4ubuntu2.8*	XN-FATFOT-TTF.BY	Ubuntu Linux 16.04	Server
□	●		217.21.37.71	TCP	22	ssh	↗	*SSH-2.0-OpenSSH_7.2p2 Ubuntu-4ubuntu2.8*	XN-FATFOT-TTF.BY	Ubuntu Linux 16.04	Server
□	●		217.21.37.72	TCP	22	ssh	↗	*SSH-2.0-OpenSSH_7.2p2 Ubuntu-4ubuntu2.8*	XN-FATFOT-TTF.BY	Ubuntu Linux 16.04	Server
□	●		217.21.37.73	TCP	22	ssh	↗	*SSH-2.0-OpenSSH_7.2p2 Ubuntu-4ubuntu2.8*	XN-FATFOT-TTF.BY	Ubuntu Linux 16.04	Server
□	●		217.21.37.74	TCP	22	ssh	↗	*SSH-2.0-OpenSSH_7.2p2 Ubuntu-4ubuntu2.8*	XN-FATFOT-TTF.BY	Ubuntu Linux 16.04	Server
□	●		217.21.37.75	TCP	22	ssh	↗	*SSH-2.0-OpenSSH_7.2p2 Ubuntu-4ubuntu2.8*	XN-FATFOT-TTF.BY	Ubuntu Linux 16.04	Server
□	●		217.21.37.76	TCP	22	ssh	↗	*SSH-2.0-OpenSSH_7.2p2 Ubuntu-4ubuntu2.8*	XN-FATFOT-TTF.BY	Ubuntu Linux 16.04	Server
□	●		217.21.37.77	TCP	22	ssh	↗	*SSH-2.0-OpenSSH_7.2p2 Ubuntu-4ubuntu2.8*	XN-FATFOT-TTF.BY	Ubuntu Linux 16.04	Server
□	●		217.21.37.79	TCP	22	ssh	↗	*SSH-2.0-OpenSSH_7.2p2 Ubuntu-4ubuntu2.8*	XN-FATFOT-TTF.BY	Ubuntu Linux 16.04	Server
□	●		217.21.37.81	TCP	22	ssh	↗	*SSH-2.0-OpenSSH_7.2p2 Ubuntu-4ubuntu2.8*	XN-FATFOT-TTF.BY	Ubuntu Linux 16.04	Server
□	●		217.21.37.82	TCP	22	ssh	↗	*SSH-2.0-OpenSSH_7.2p2 Ubuntu-4ubuntu2.8*	XN-FATFOT-TTF.BY	Ubuntu Linux 16.04	Server
□	●		217.21.37.83	TCP	22	ssh	↗	*SSH-2.0-OpenSSH_7.2p2 Ubuntu-4ubuntu2.8*	XN-FATFOT-TTF.BY	Ubuntu Linux 16.04	Server
□	●		217.21.37.84	TCP	22	ssh	↗	*SSH-2.0-OpenSSH_7.2p2 Ubuntu-4ubuntu2.8*	XN-FATFOT-TTF.BY	Ubuntu Linux 16.04	Server



	Up	Attrs	Address	Transport	Port ↑	Protocol	VHost	Summary	Hostname	OS	Type	H
□	●		134.17.94.240	TCP	22	ssh	↗	*SSH-2.0-OpenSSH_7.4*	240-94-17-134-CLOUD.MTS.BY			
□	●		134.17.16.186	TCP	22	ssh	↗	*SSH-2.0-OpenSSH_7.4*	186-16-17-134-CLOUD.MTS.BY	Centos Linux 7	Server	
□	●		134.17.16.213	TCP	22	ssh	↗	*SSH-2.0-OpenSSH_7.4*	213-16-17-134-CLOUD.MTS.BY	Centos Linux 7	Server	
□	●		134.17.94.105	TCP	22	ssh	↗	*SSH-2.0-OpenSSH_7.4*	ATEVI.BY+1	Centos Linux	Server	
□	●		134.17.16.48	TCP	22	ssh	↗	*SSH-2.0-OpenSSH_8.0*	48-16-17-134-CLOUD.MTS.BY	Centos Linux	Server	
□	●		134.17.94.137	TCP	22	ssh	↗	*SSH-2.0-OpenSSH_7.4*	137-94-17-134-CLOUD.MTS.BY			
□	●		134.17.94.190	TCP	22	ssh	↗	*SSH-2.0-OpenSSH_7.4*	IVCPORTAL.BY	Centos Linux 7	Server	
□	●		134.17.16.113	TCP	22	ssh	↗	*SSH-2.0-OpenSSH_7.4*	IMDISTRI.BY+1	Centos Linux	Server	
□	●		134.17.16.237	TCP	22	ssh	↗	*SSH-2.0-OpenSSH_7.4*	BITRIX+1	Centos Linux 7	Server	
□	●		134.17.94.82	TCP	22	ssh	↗	*SSH-2.0-OpenSSH_7.4*	VLADYUD.COM+2	Centos Linux 7	Server	
□	●		134.17.94.39	TCP	22	ssh	↗	*SSH-2.0-OpenSSH_7.4*	INVENTO-LABS.COM+1	Fedora Project Linux Fedora Core	Server	
□	●		134.17.16.62	TCP	22	ssh	↗	*SSH-2.0-OpenSSH_7.4*	62-16-17-134-CLOUD.MTS.BY	Centos Linux	Server	
□	●		134.17.16.71	TCP	22	ssh	↗	*SSH-2.0-OpenSSH_7.4*	71-16-17-134-CLOUD.MTS.BY	Centos Linux	Server	
□	●		134.17.16.214	TCP	22	ssh	↗	*SSH-2.0-OpenSSH_7.4*	214-16-17-134-CLOUD.MTS.BY	Centos Linux 7	Server	
□	●		134.17.17.240	TCP	22	ssh	↗	*SSH-2.0-OpenSSH_7.4*		Centos Linux 7	Server	
□	●		134.17.17.241	TCP	22	ssh	↗	*SSH-2.0-OpenSSH_7.4*		Centos Linux 7	Server	
□	●		134.17.94.33	TCP	22	ssh	↗	*SSH-2.0-OpenSSH_7.4*	33-94-17-134-CLOUD.MTS.BY	Centos Linux 7	Server	

Example: TLS certificate hashes

- TLS certificates shouldn't cross security levels
- Similar impact as shared SSH hostkeys

Result

- Flag cloned Windows and insecure cert sharing

Detection

- TLS scans (sslyze/nmap/runzero)

Find the unexpectedly uncommon things

- Dashboards like to show most common + *other*
- The interesting stuff is always in *other*
- Flip your reporting to least common
- Dynamic fields need more…

Calculate outlier as distance from common

- Re-analyze the entire population on every change
- Baseline is conditional on SUM(TopX) > Y%
- Least frequent values mapped to ranks
- Ranks can drive an outlier score
- Simple stat calcs, not AI/ML
- Ignores noisy data

Example: Server-side TCP MSS values

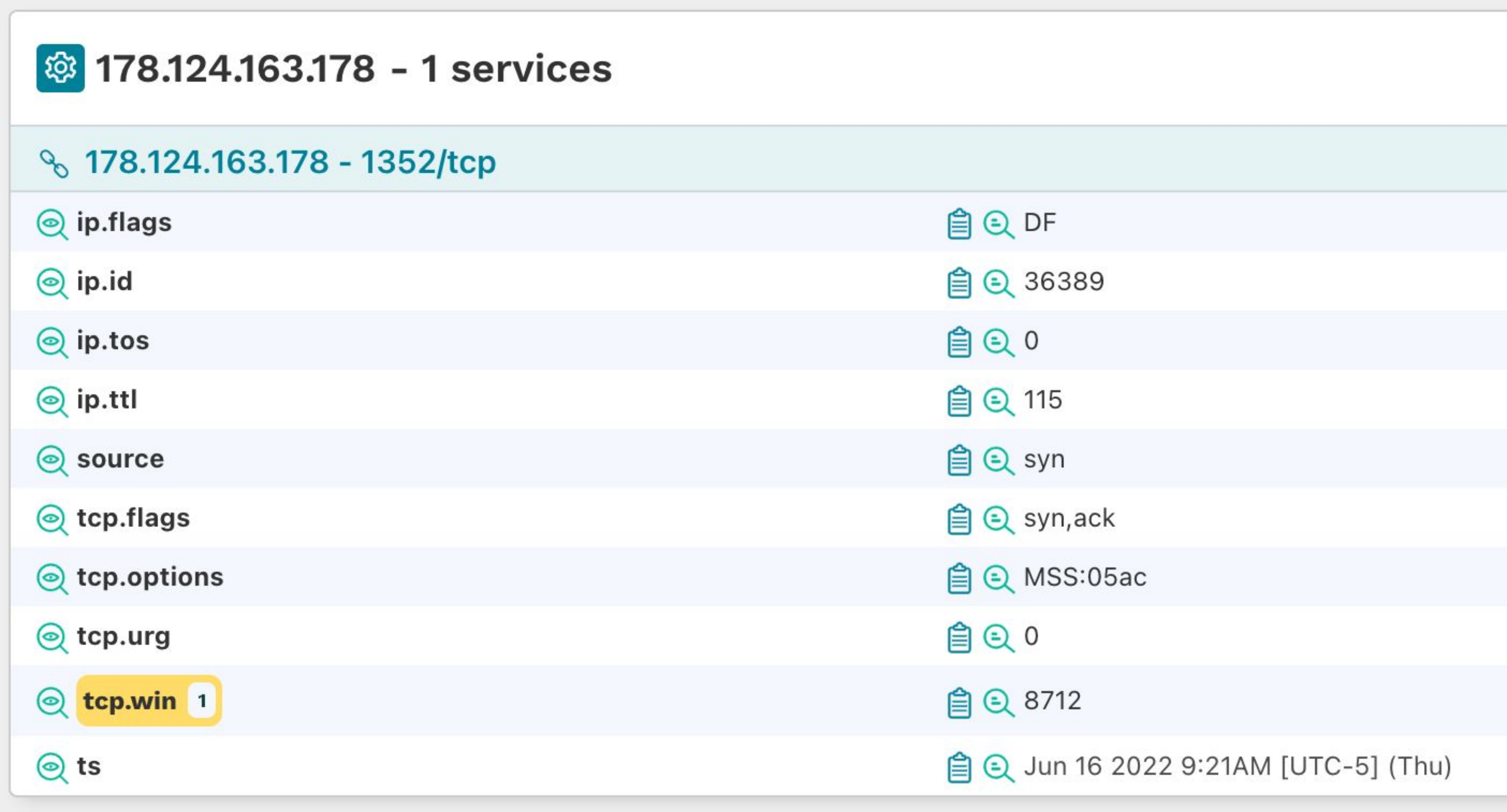
- Only a handful of common values (Win/Lin/Mac)
- Anything else is typically an embedded OS
- Ex: **NOT** 28960, 14480, 65160, 65535

Result

- Immediate detection of all “weird” devices

Detection

- TCP SYN on any open port (nmap/runzero)



178.124.163.178 - 1 services

178.124.163.178 - 1352/tcp

ip.flags	DF
ip.id	36389
ip.tos	0
ip.ttl	115
source	syn
tcp.flags	syn,ack
tcp.options	MSS:05ac
tcp.urg	0
tcp.win 1	8712
ts	Jun 16 2022 9:21AM [UTC-5] (Thu)

Example: SSH service attributes

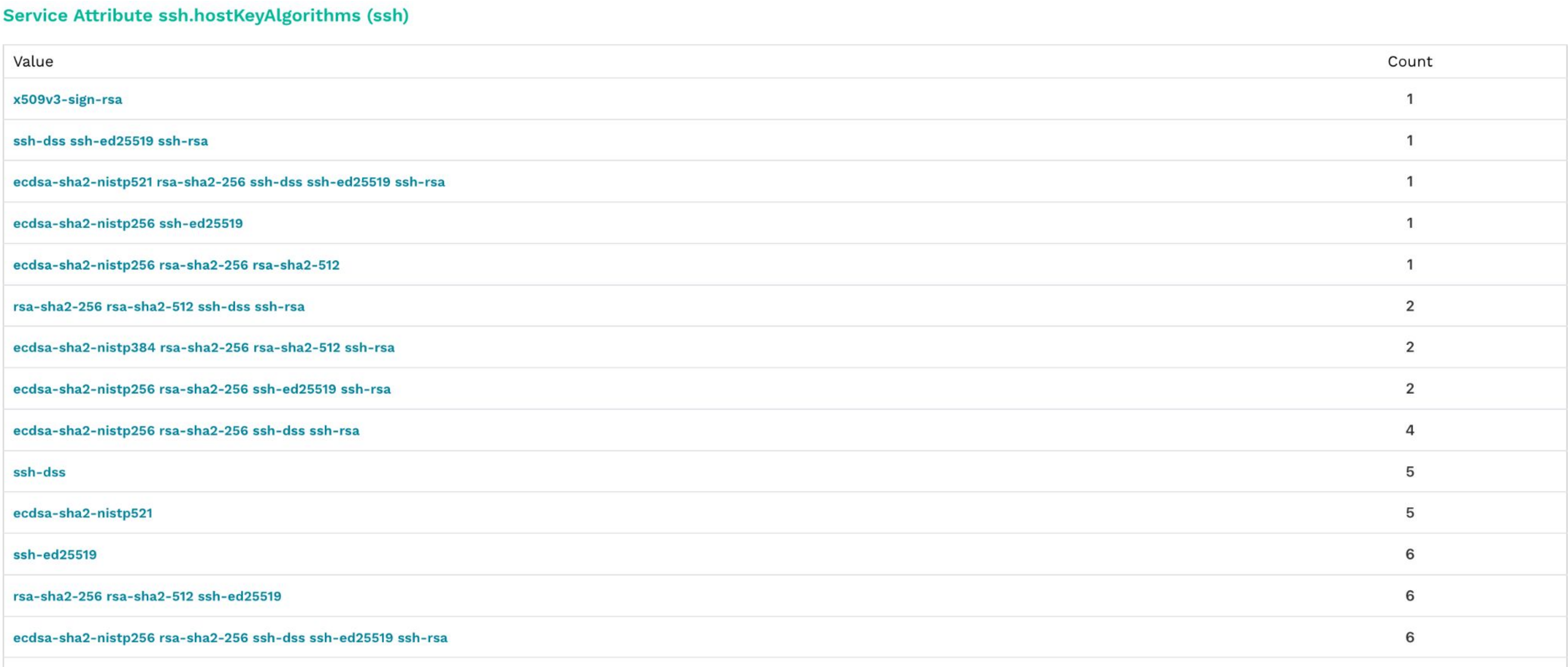
- Banners typically tied to OS & version
- Oddball key exchanges and auths
- The least common are usually bad

Result

- Quickly triage embedded and unmanaged devices

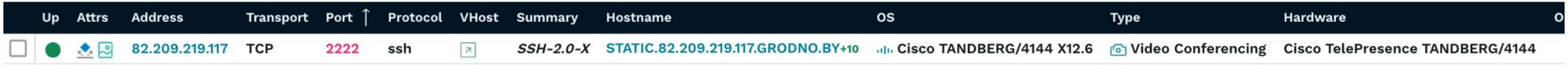
Detection

- TCP connect on SSH ports (nmap/runzero)



Service Attribute ssh.hostKeyAlgorithms (ssh)

Value	Count
x509v3-sign-rsa	1
ssh-dss ssh-ed25519 ssh-rsa	1
ecdsa-sha2-nistp521 rsa-sha2-256 ssh-dss ssh-ed25519 ssh-rsa	1
ecdsa-sha2-nistp256 ssh-ed25519	1
ecdsa-sha2-nistp256 rsa-sha2-256 rsa-sha2-512	1
rsa-sha2-256 rsa-sha2-512 ssh-dss ssh-rsa	2
ecdsa-sha2-nistp384 rsa-sha2-256 rsa-sha2-512 ssh-rsa	2
ecdsa-sha2-nistp256 rsa-sha2-256 ssh-ed25519 ssh-rsa	2
ecdsa-sha2-nistp256 rsa-sha2-256 ssh-dss ssh-rsa	4
ssh-dss	5
ecdsa-sha2-nistp521	5
ssh-ed25519	6
rsa-sha2-256 rsa-sha2-512 ssh-ed25519	6
ecdsa-sha2-nistp256 rsa-sha2-256 ssh-dss ssh-ed25519 ssh-rsa	6



	Up	Attrs	Address	Transport	Port ↑	Protocol	VHost	Summary	Hostname	OS	Type	Hardware	O
□	●		82.209.219.117	TCP	2222	ssh		*SSH-2.0-X*	STATIC.82.209.219.117.GRODNO.BY+10	Cisco TANDBERG/4144 X12.6	Video Conferencing	Cisco TelePresence TANDBERG/4144	

Example: Windows domain values

- Obtain through NTLMSSP, SMB, NetBIOS
- The majority are in a known domain
- Everything else is possibly unmanaged

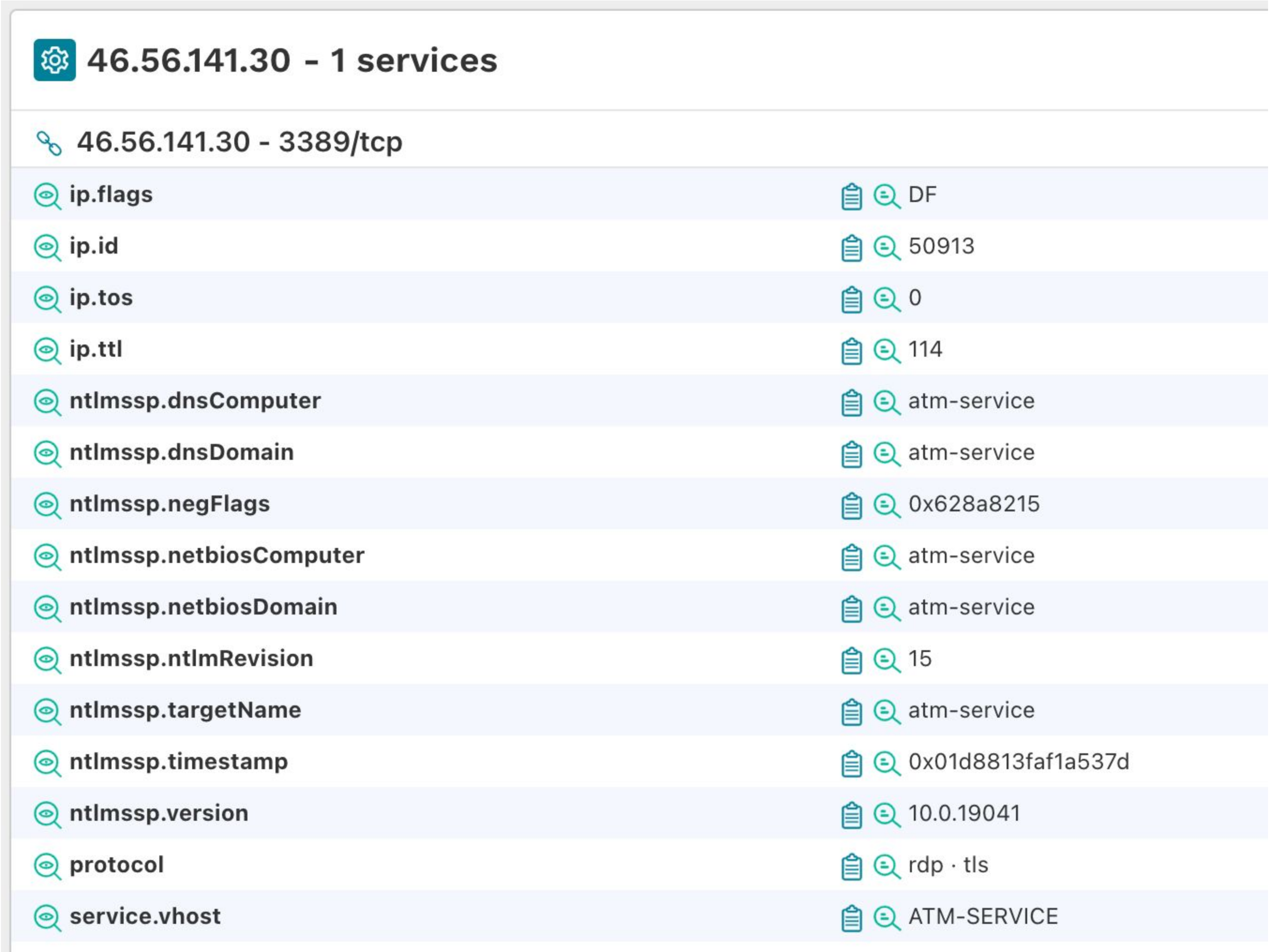
Result

- Find abandoned Windows systems

Detection

- TCP/UDP probes x many protocols (nmap/runzero)

runZero



46.56.141.30 - 1 services

46.56.141.30 - 3389/tcp

ip.flags	DF
ip.id	50913
ip.tos	0
ip.ttl	114
ntlmssp.dnsComputer	atm-service
ntlmssp.dnsDomain	atm-service
ntlmssp.negFlags	0x628a8215
ntlmssp.netbiosComputer	atm-service
ntlmssp.netbiosDomain	atm-service
ntlmssp.ntlmRevision	15
ntlmssp.targetName	atm-service
ntlmssp.timestamp	0x01d8813faf1a537d
ntlmssp.version	10.0.19041
protocol	rdp · tls
service.vhost	ATM-SERVICE

Example: Hardware models

- Pull data from scans, captures, or EDR/MDM APIs
- Review the least common models
- Flag everything else for review

Result

- Find IoT gadgets & end-of-life platforms

Detection

- Fingerprints + integrations (nmap/curl/runzero)

Asset Field HW

Value	Count
iRobot Roomba	1
Zyxel USG310	1
Zyxel USG1100	1
Zyxel USG110	1
Zyxel GS1920	1
Zyxel Firewall	1
ZTE ZXHN H208N	1
Yealink VoIP	1
Yealink SIP-T46U	1
Yealink SIP-T19P_E2	1
Yamaha RX-V781	1
VirtualBox VM	1
Uniview NVR302-16S	1

Is an outlier usually insecure?

- Let’s find out by correlating with vulnerability data
- Sample size of 500k hosts with outliers + vulns
- Ranked vulnerabilities from 0-4 (4 = critical)
- Ranked outliers from 0-5 (5 = super weird)

Outlier vs average risk correlation

- Yes, an almost perfect (AVG) correlation!

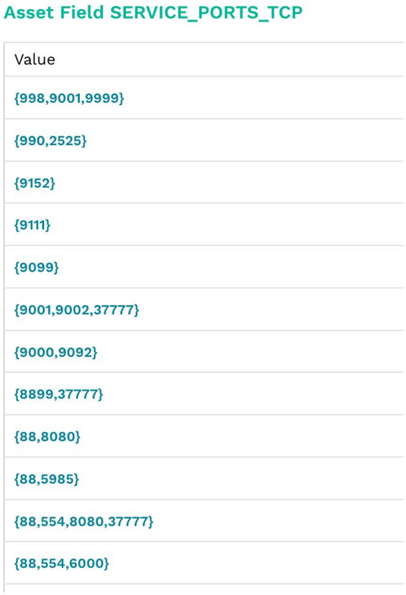
Outlier Rank (0-5, 5 = weirdest)	Average Risk (0-4, 4 = critical)
0	0.49
1	1.09
2	1.29
3	1.93
4	3.13
5	3.67

Why does this work in general?

- The attributes chosen for outliers are important
  - OS, OS Version, Hardware, Firmware Version
  - Rarity tracks strongly with exposure
  - Systems that have been forgotten
  - Vendor-managed devices

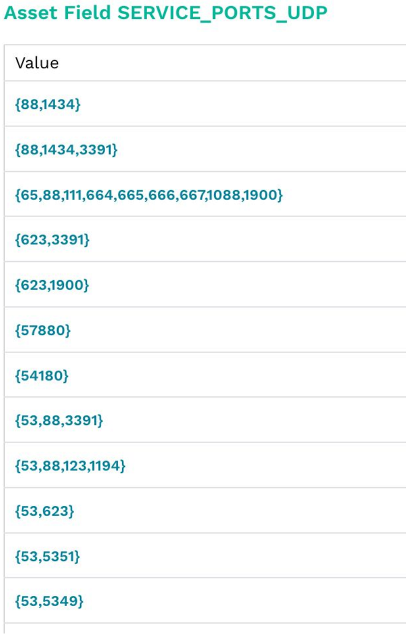
Unusual attributes can be predictive

- TCP MSS, port combinations, IP ToS fields



Asset Field SERVICE_PORTS_TCP

Value
{998,9001,9999}
{990,2525}
{9152}
{9111}
{9099}
{9001,9002,37777}
{9000,9092}
{8899,37777}
{88,8080}
{88,5985}
{88,554,8080,37777}
{88,554,6000}



Asset Field SERVICE_PORTS_UDP

Value
{88,1434}
{88,1434,3391}
{65,88,111,664,665,666,667,1088,1900}
{623,3391}
{623,1900}
{57880}
{54180}
{53,88,3391}
{53,88,123,1194}
{53,623}
{53,5351}
{53,5349}

Do we still need vulnerability scanners?

- Yes! The risk-to-outlier correlation is weaker
- This correlation is still based on averages
- Easy to miss things using outliers alone

Outliers are a high-signal starting point

- You already have this data from existing tools
- Export to CSV, load into Excel/Google Sheets
- Pivot table or otherwise group + count
- Start hunting the weird stuff!

Q & A

Get in touch!

- hdm/at/runZero.com
- @hdm@infosec.exchange
- https://hdm.io

Keep Assets Weird