Custom Detections – Everything you need to know

Defender Boys - Mattias Borg, Stefan Schörling

Sponsors

onedeploy
Windows Deployment Reimagined

ZERO TOUCH

Insight

control up

PATCH MY PC

robopack

numecent

Microsoft

eido

Rimo3

These slides have been fixed to avoid pictures covering content when printing to pdf

Stefan Schörling – SA7STE

- Microsoft MVP · Security – SIEM & XDR

Role

- 01000011 01010100 01001111 - Onevinn

Focus

- Security & SIGINT

Blog, Hobbies and more

- blog.sec-labs.com
- Security / Intelligence
- Padel

Mattias Borg
Microsoft MVP · Security – SIEM & XDR

Role
Magician, DFIR, Offensive - Onevinn

Focus
Cyber Security & Research

Blog, Hobbies and more
Write stuff, Build stuff, Break stuff, Paint stuff

Agenda

- Why do we need custom detections
- Custom Detections – Core and Building Blocks
- Use-cases
- Wrap up and summary
- Go home and build your own Custom Detections

- Tool release (For Red and Blue)

Why do we need custom detections?

I bought a tool, now I can sit back and relax or maybe not.

Detect bad things

Hypothesis driven

Use-case driven

Proactive Threat Hunting

Tasks

Detections

Find the hidden
Proactive Threat Hunting

Query-based data review
Daily/Weekly

Custom Detections
Known bad or suspicious

MITRE ATT&CK

ATT&CK®

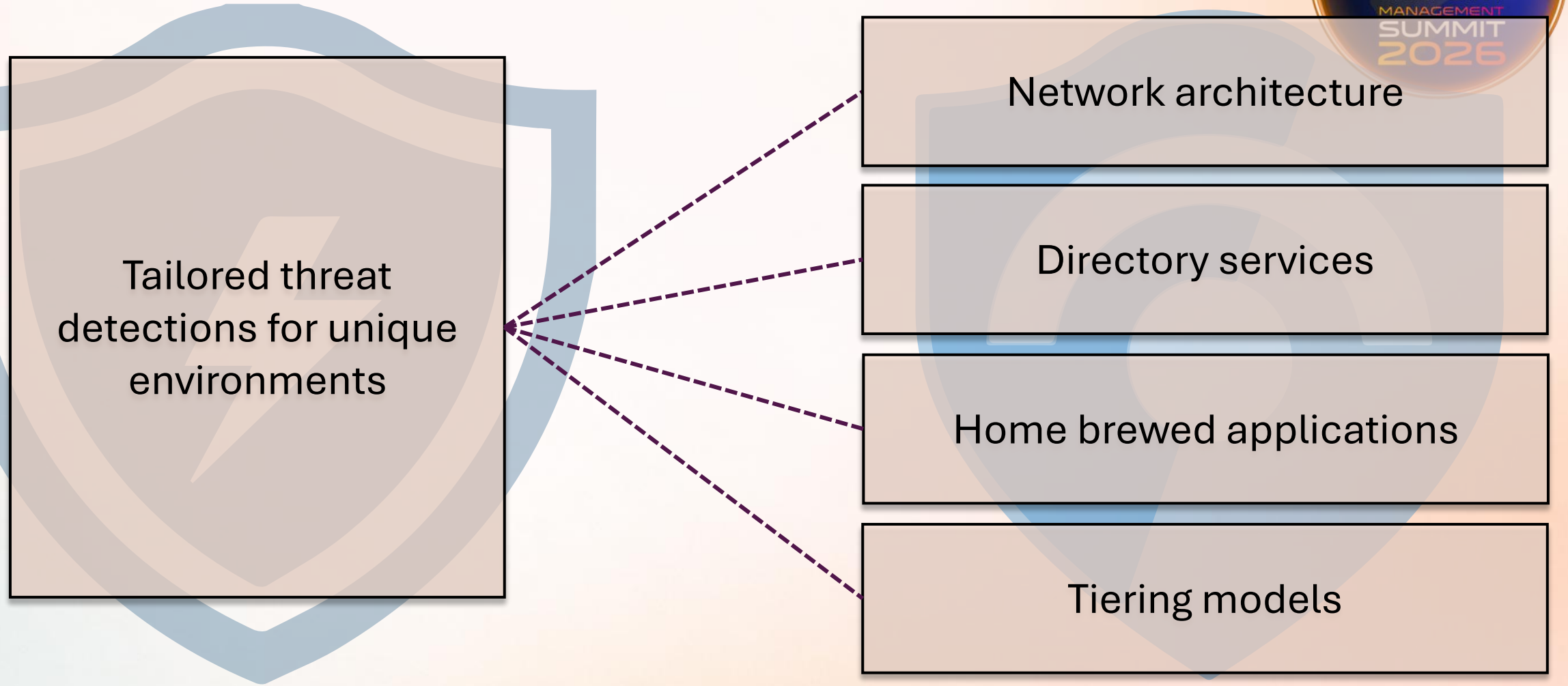
Reconnaissance	Resource Development	Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Command and Control	Exfiltration	Impact
10 techniques	8 techniques	9 techniques	14 techniques	19 techniques	13 techniques	42 techniques	17 techniques	31 techniques	9 techniques	17 techniques	16 techniques	9 techniques	13 techniques
Active Scanning (3)	Acquire Access	Drive-by Compromise	Cloud Administration Command	Account Manipulation (5)	Abuse Elevation Control Mechanism (4)	Abuse Elevation Control Mechanism (4)	Adversary-in-the-Middle (3)	Account Discovery (4)	Exploitation of Remote Services	Adversary-in-the-Middle (3)	Application Layer Protocol (4)	Automated Exfiltration (1)	Account Access Removal
Gather Victim Host Information (4)	Acquire Infrastructure (8)	Exploit Public-Facing Application	Command and Scripting Interpreter (9)	BITS Jobs	Access Token Manipulation (5)	Access Token Manipulation (5)	Brute Force (4)	Application Window Discovery	Internal Spearphishing	Archive Collected Data (3)	Communication Through Removable Media	Data Transfer Size Limits	Data Destruction
Gather Victim Identity Information (3)	Compromise Accounts (3)	External Remote Services	Container Administration Command	Boot or Logon Autostart Execution (14)	Boot or Logon Autostart Execution (14)	BITS Jobs	Credentials from Password Stores (5)	Browser Information Discovery	Lateral Tool Transfer	Audio Capture	Data Encoding (2)	Exfiltration Over Alternative Protocol (3)	Data Encrypted for Impact
Gather Victim Network Information (6)	Compromise Infrastructure (7)	Hardware Additions	Deploy Container	Boot or Logon Initialization Scripts (5)	Boot or Logon Initialization Scripts (5)	Build Image on Host	Exploitation for Credential Access	Cloud Infrastructure Discovery	Remote Service Session Hijacking (2)	Automated Collection	Data Obfuscation (3)	Exfiltration Over C2 Channel	Data Manipulation (3)
Gather Victim Org Information (4)	Develop Capabilities (4)	Phishing (3)	Exploitation for Client Execution	Browser Extensions	Create or Modify System Process (4)	Debugger Evasion	Forced Authentication	Cloud Service Dashboard	Remote Services (7)	Browser Session Hijacking	Dynamic Resolution (3)	Exfiltration Over Other Network Medium (1)	Defacement (2)
Phishing for Information (3)	Establish Accounts (3)	Replication Through Removable Media	Inter-Process Communication (3)	Compromise Client Software Binary	Domain Policy Modification (2)	Deobfuscate/Decode Files or Information	Forge Web Credentials (2)	Cloud Service Discovery	Replication Through Removable Media	Clipboard Data	Encrypted Channel (2)	Exfiltration Over Physical Medium (1)	Disk Wipe (2)
Search Closed Sources (2)	Obtain Capabilities (6)	Supply Chain Compromise (3)	Native API	Create Account (3)	Escape to Host	Deploy Container	Input Capture (4)	Cloud Storage Object Discovery	Software Deployment Tools	Data from Cloud Storage	Fallback Channels	Exfiltration Over Web Service (3)	Endpoint Denial of Service (4)
Search Open Technical Databases (5)	Stage Capabilities (6)	Trusted Relationship	Scheduled Task/Job (5)	Create or Modify System Process (4)	Event Triggered Execution (16)	Direct Volume Access	Modify Authentication Process (8)	Container and Resource Discovery	Taint Shared Content	Data from Configuration Repository (2)	Ingress Tool Transfer	Scheduled Transfer	Firmware Corruption
Search Open Websites/Domains (3)		Valid Accounts (4)	Serverless Execution	Event Triggered Execution (16)	Exploitation for Privilege Escalation	Domain Policy Modification (2)	Multi-Factor Authentication Interception	Debugger Evasion	Use Alternate Authentication Material (4)	Data from Information Repositories (3)	Multi-Stage Channels	Transfer Data to Cloud Account	Inhibit System Recovery
Search Victim-Owned Websites			Shared Modules	External Remote Services	Hijack Execution Flow (12)	Execution Guardrails (1)	Multi-Factor Authentication Request Generation	Device Driver Discovery		Data from Local System	Non-Application Layer Protocol		Network Denial of Service (2)
			Software Deployment Tools	Hijack Execution Flow (12)	Process Injection (12)	Exploitation for Defense Evasion	Network Sniffing	Domain Trust Discovery		Data from Network Shared Drive	Non-Standard Port		Resource Hijacking
			System Services (2)			File and Directory Permissions Modification (2)		File and Directory Discovery			Protocol Tunneling		Service Stop
						Hide Artifacts (10)		Group Policy Discovery			Proxy (4)		System Shutdown/Reboot
								Network Service Discovery					

https://attack.mitre.org

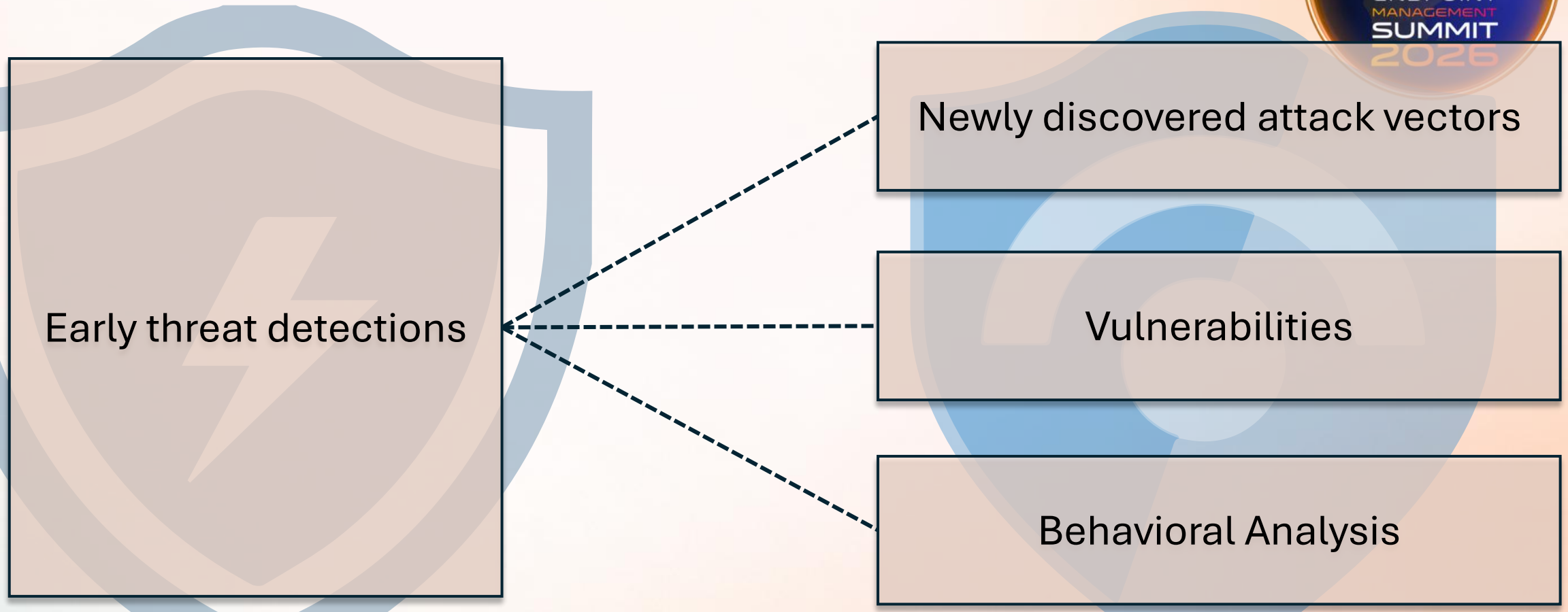
DEMO

MITRE ATT&CK & Threat Analytics

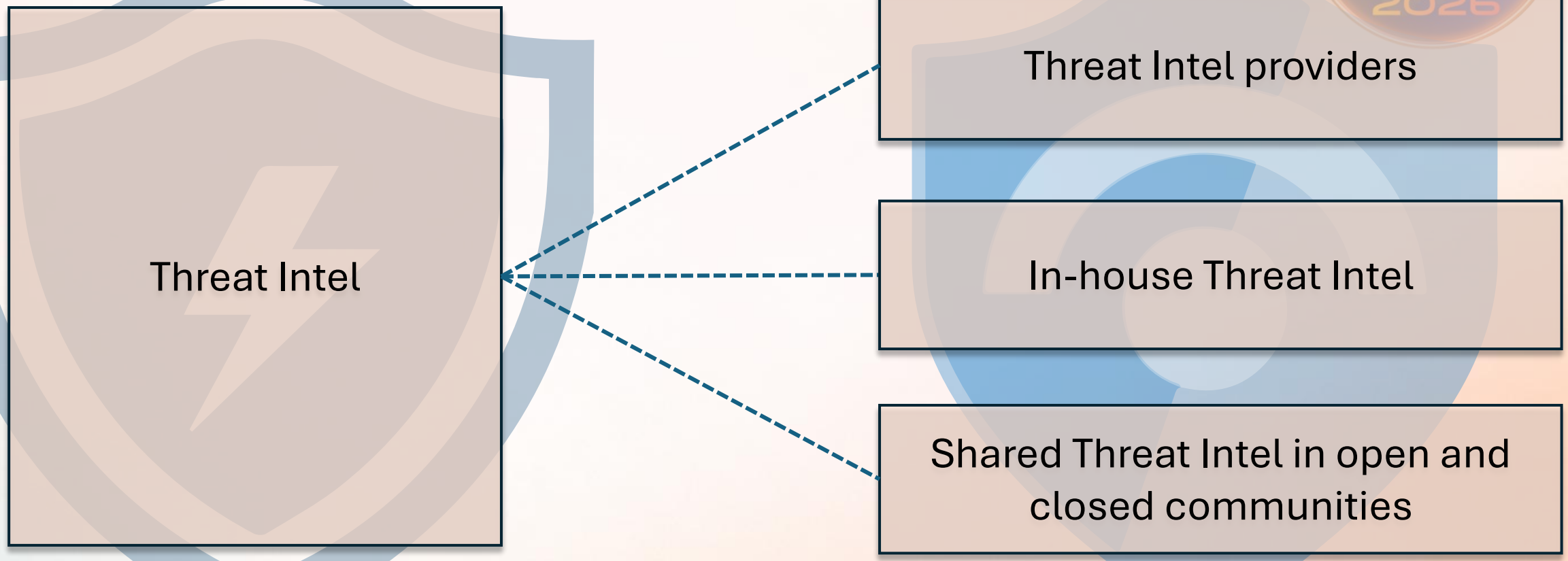
Why custom detections



Reason 1


Early threat detections
Newly discovered attack vectors
Vulnerabilities
Behavioral Analysis

Reason 2


Threat Intel
Threat Intel providers
In-house Threat Intel
Shared Threat Intel in open and closed communities

Developing Use-Cases

Core building pieces

Components

Aggregated data

Documentation

Use-case

Management

Well-defined description

Threat Intelligence

Data sources

Logic

Playbook

What is the goal

MITRE ATT&CK MAP

Severity

What is bad

Required data sources

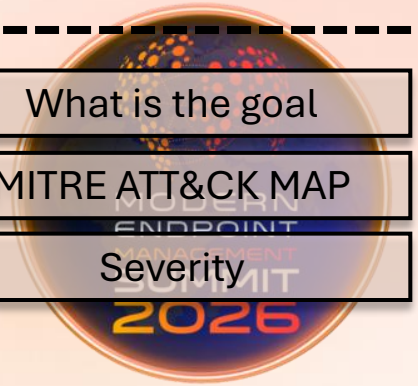
Query data

Scheduling

Thresholds

Required actions

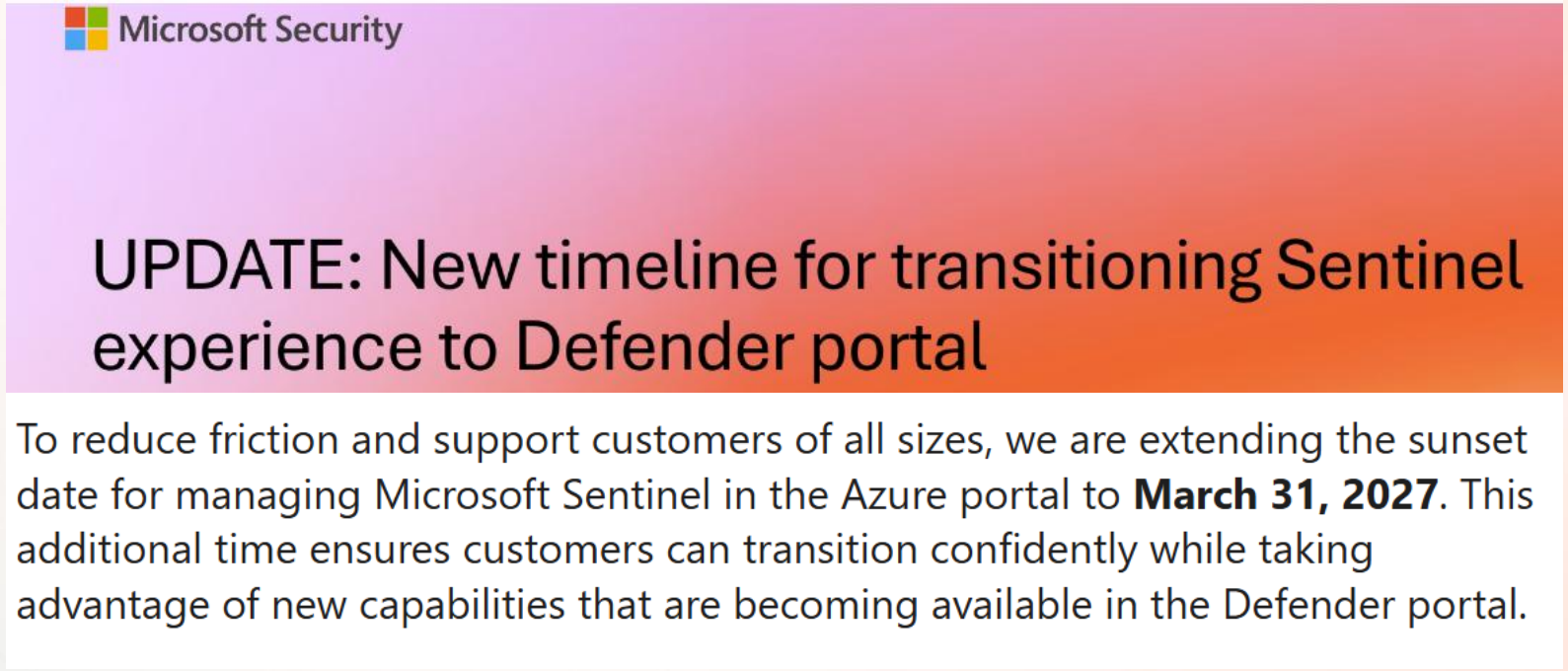
Automation



Jump into the tech side

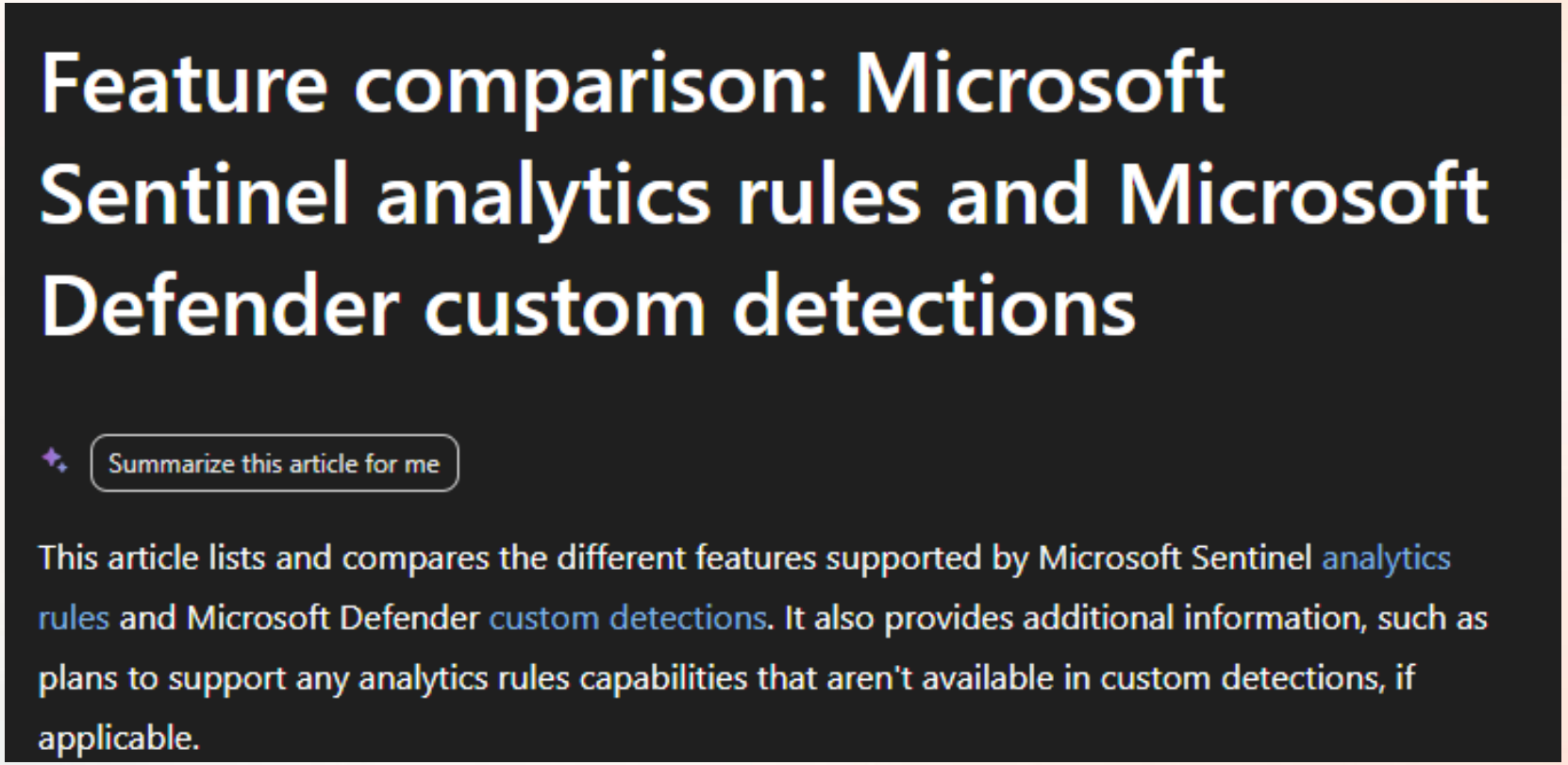
Unified Custom Detections

USX – Sentinel moves into XDR



https://learn.microsoft.com/en-us/azure/sentinel/move-to-defender

Analytic Rules vs Custom Detections



https://learn.microsoft.com/en-us/azure/sentinel/compare-analytics-rules-custom-detections

Important update: Custom detections are now the unified experience for creating detections in Microsoft Defender

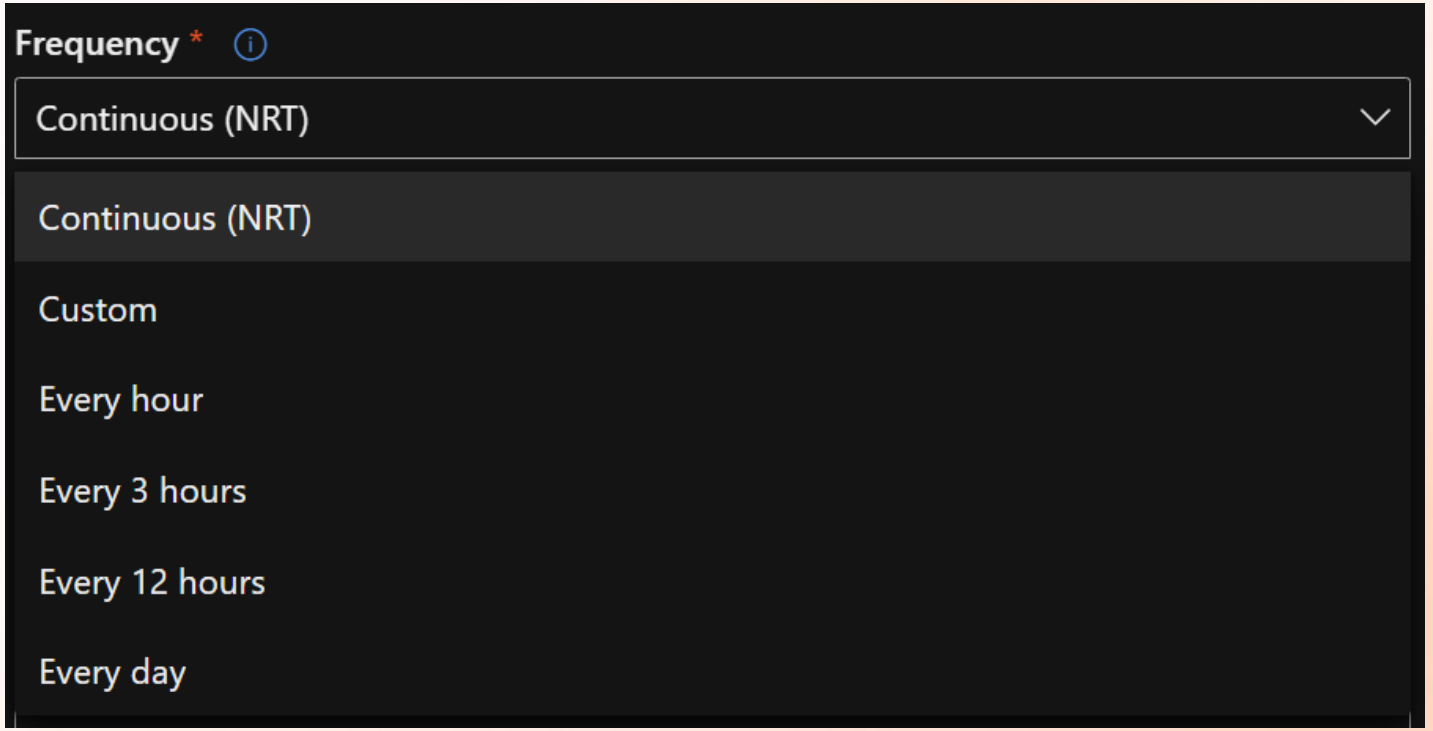
https://techcommunity.microsoft.com/blog/microsoftthreatprotectionblog/custom-detections-are-now-the-unified-experience-for-creating-detections-in-micr/4463875

Watchlists

- Can be used in Custom Detections (unified)
- Exclusions – Same query in multi-tenant scenarios

Scheduling

- **Challenge**
  - NRT is limited to Single Tables
  - NRT does not support joins, unions or external data



Query Lookback

Lookback

The lookback period of your custom detections can range from five minutes to 30 days, depending on the target data and frequency of your query.

If your custom detections include Defender XDR data, a fixed lookback period is applied depending on the rule frequency that you choose:

- For detections set to run **every 24 hours**, the lookback period is **30 days**.
- For detections set to run **every 12 hours**, the lookback period is **48 hours**.
- For detections set to run **every three hours**, the lookback period is **12 hours**.
- For detections set to run **hourly**, the lookback period is **four hours**.

If your custom detections target Microsoft Sentinel data only, you can customize the lookback period depending on the rule frequency that you set:

- For detections set to run in frequencies **higher (more frequent) than one hour**, the lookback period is limited to **less than 48 hours**.
- For detections set to run in frequencies **higher than one day**, the lookback can be set **up to 14 days**.
- For detections set to run in frequencies of **one day or less**, the lookback can be set **up to 30 days**.

ⓘ Important

Custom detections evaluate `ingestion_time()` to account for ingestion delays. Because of this condition, events with `Timestamp` or `TimeGenerated` values older than the configured lookback period might still be included in the rule evaluation.

When the lookback period is longer than the frequency, duplicate events might occur. However, custom detections group and deduplicate them automatically to reduce alert noise and fatigue.

https://learn.microsoft.com/en-us/defender-xdr/custom-detection-rules

External data

Dynamically enrich

Can be somewhat secured

Use AI to create lists?

```
let PortInfo = externaldata (Port:int,Protocol:string,Service:string,Category:string,Risk:string,Description:string)
[
@"https://raw.githubusercontent.com/mattiasborg82/Hunting/refs/heads/main/General/PortsInfo.csv"
]
with(format="csv",ignoreFirstRecord=true);
DeviceNetworkEvents
| take 100
| join kind=leftouter PortInfo on $left.RemotePort == $right.Port
```

Tuning

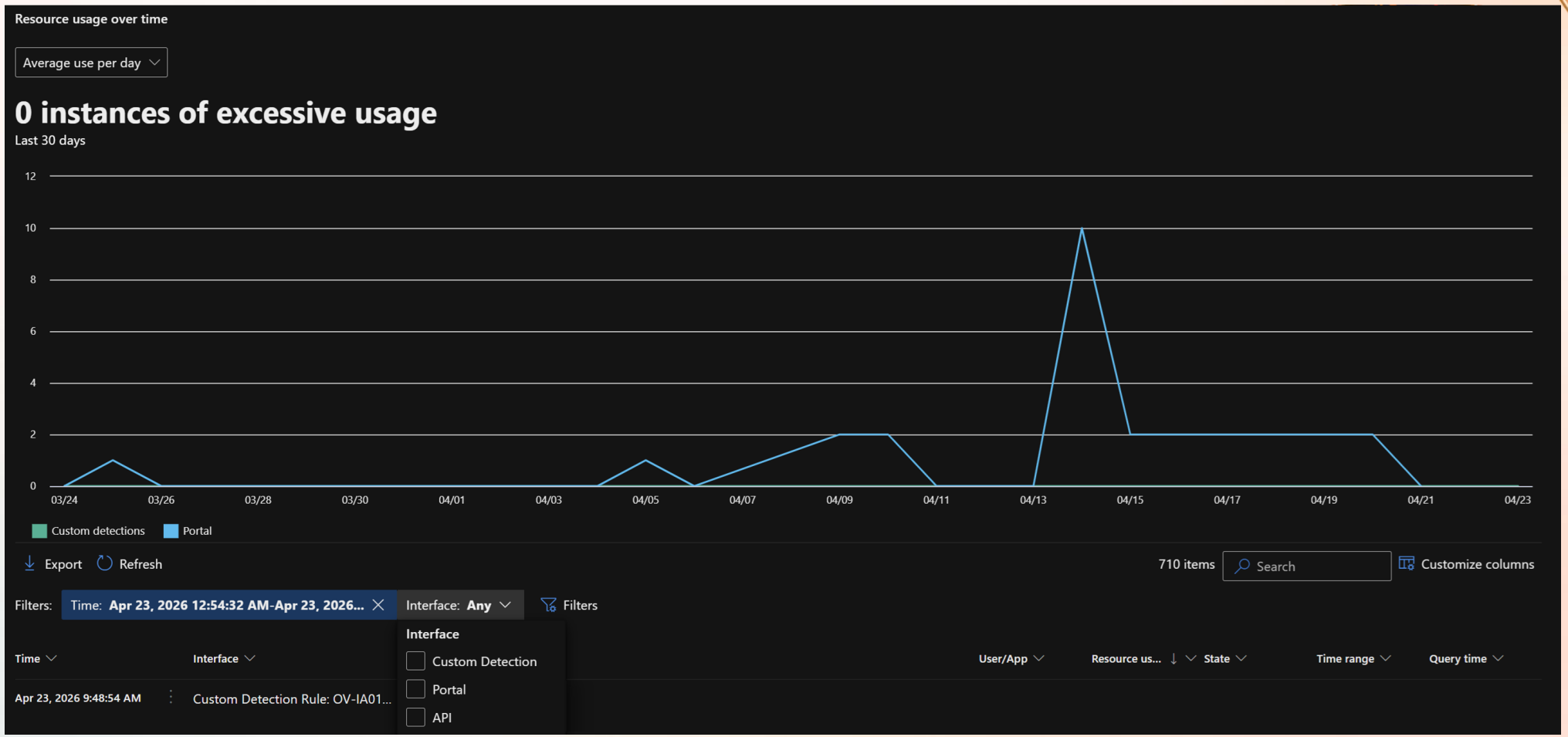
Avoid change query

Delegate

Multi-tenant

```
let Exclusions = _GetWatchlist('EXCL-RULE-XX');
DeviceProcessEvents
| where Bad = true
| where AccountName !in~(Exclusions)
```

Query Resources Report



DEMO

What does Unified Custom Detections look like

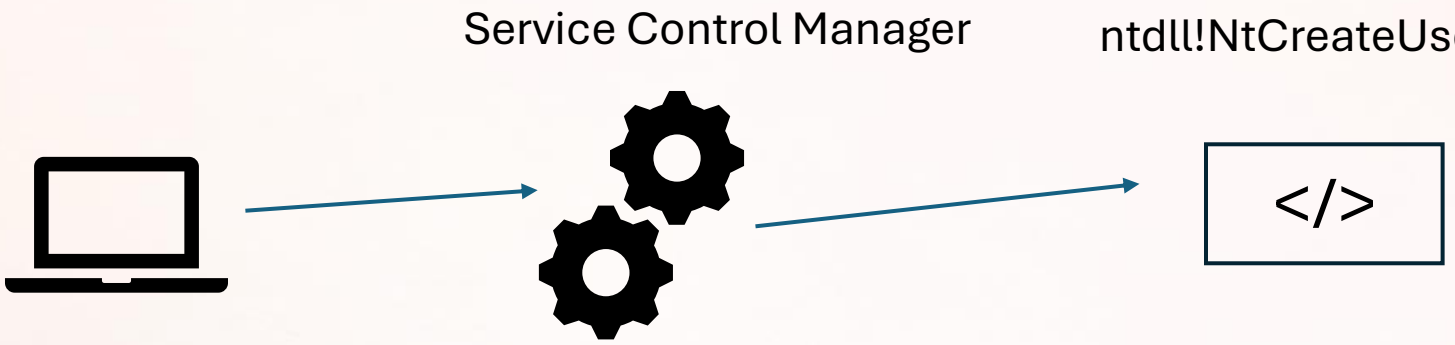
Custom Detections API

Method	Return type	Description
List	microsoft.graph.security.detectionRule collection	Get a list of the microsoft.graph.security.detectionRule objects and their properties.
Get	microsoft.graph.security.detectionRule	Read the properties and relationships of a microsoft.graph.security.detectionRule object.
Create	microsoft.graph.security.detectionRule	Create a microsoft.graph.security.detectionRule.
Update	microsoft.graph.security.detectionRule	Update the properties of a microsoft.graph.security.detectionRule object.
Delete	None	Delete a microsoft.graph.security.detectionRule object.

https://learn.microsoft.com/en-us/graph/api/resources/security-detectionrule?view=graph-rest-beta

USE-CASES

Abusing unquoted service path



- interpret the first token as the executable
- If it fails, expand to the next space
- Repeat until something exists on disk

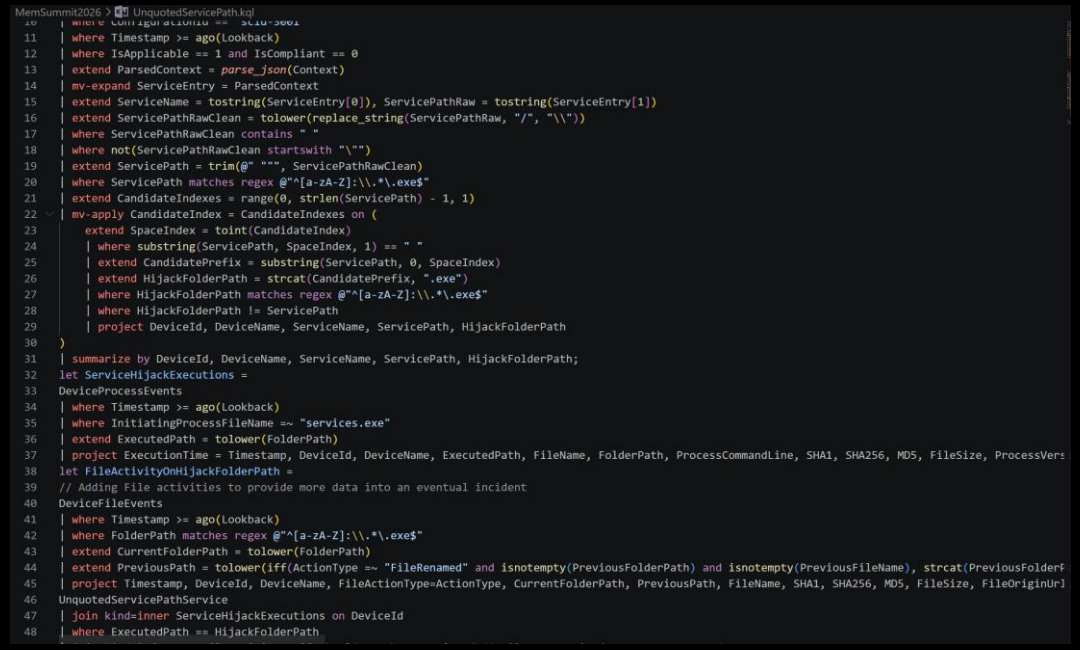
Sample Service

- **C:\Program.exe**
- **C:\Program Files\Super.exe**
- **C:\Program Files\Super Security.exe**
- **C:\Program Files\Super Security Vendor\SuperService.exe**

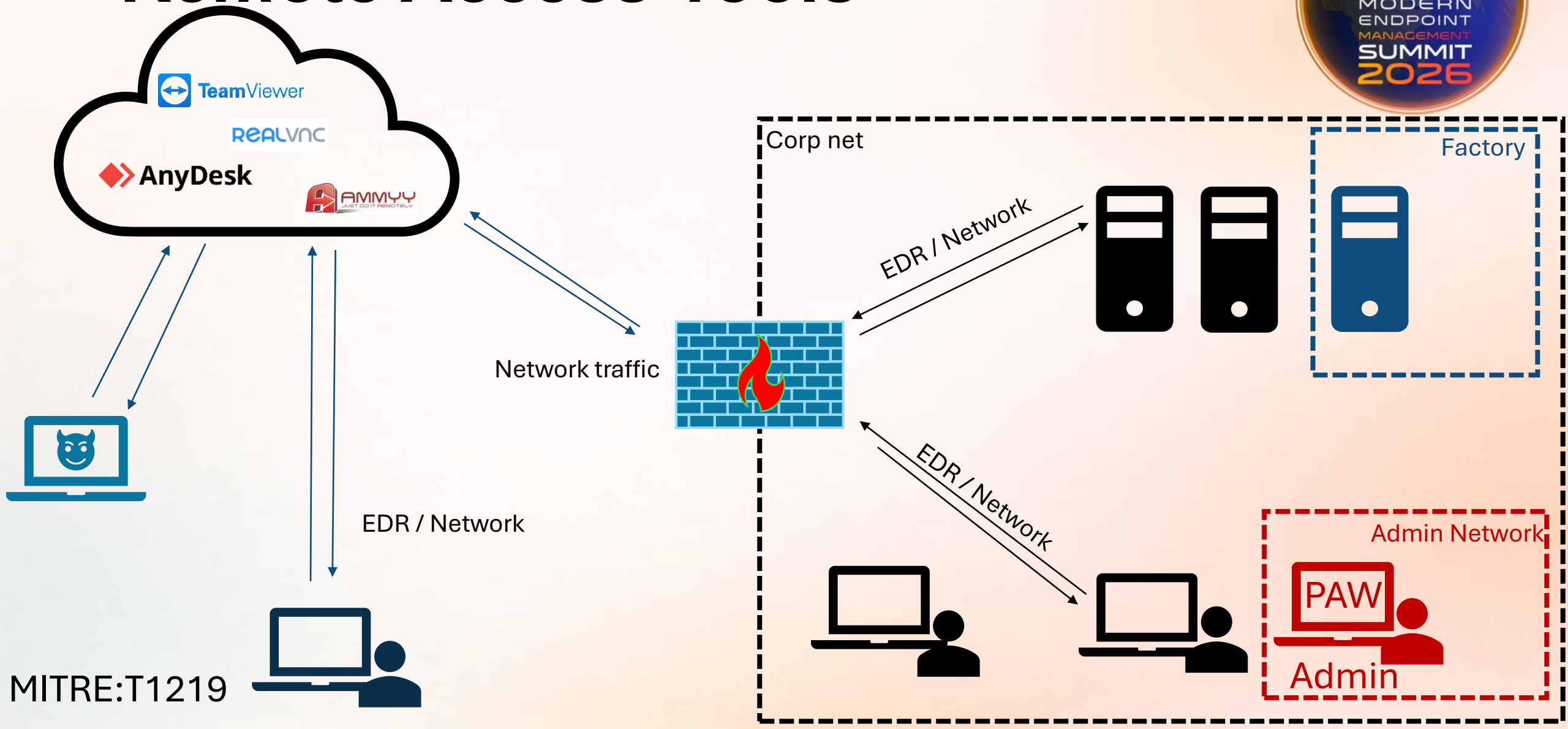
Abusing unquoted service path

To long for a slide…

https://github.com/mattiasborg82/Talks/tree/main/MemSummit2026



Remote Access Tools



MITRE:T1219

Remote Access Tools

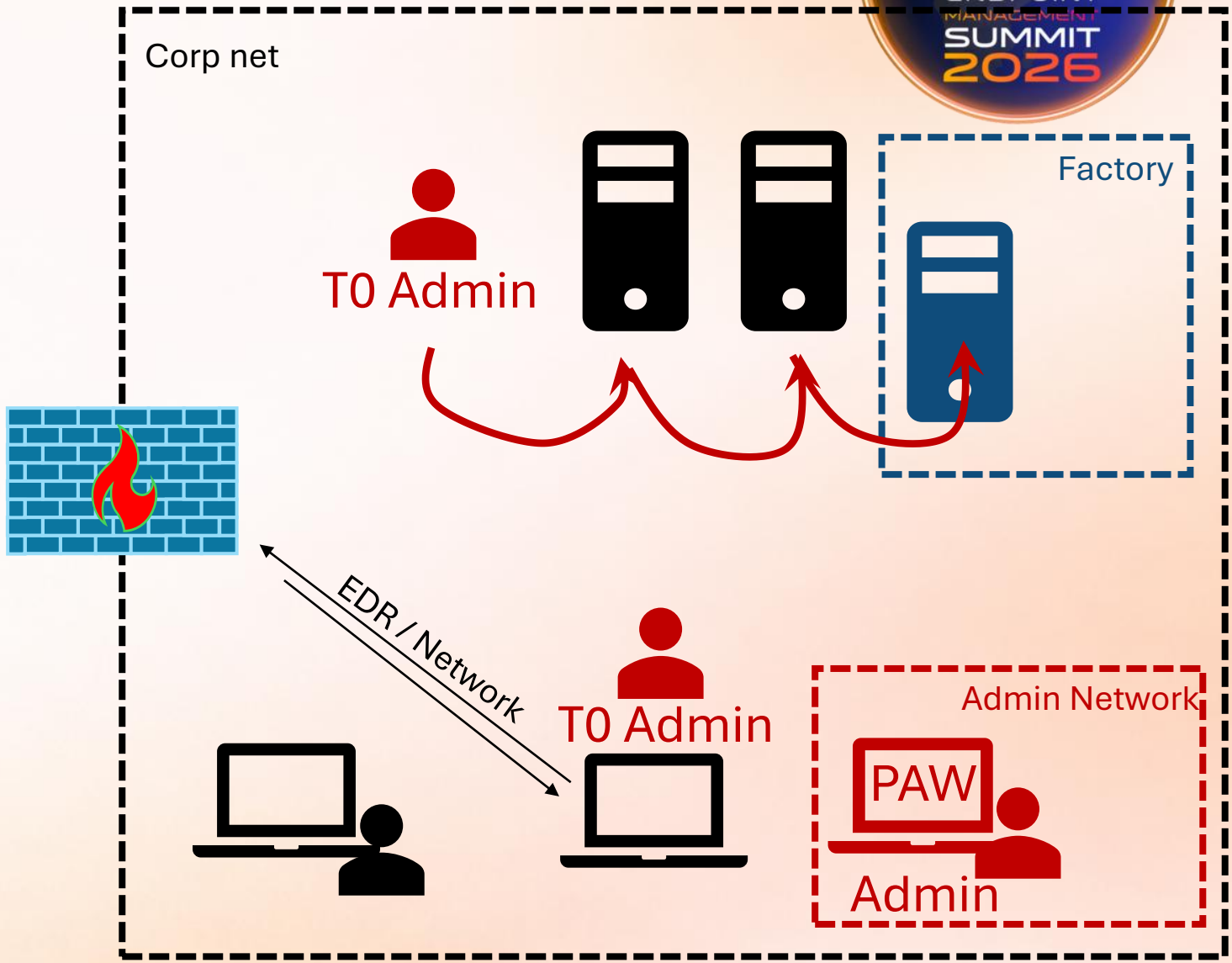

```
let KnownNames = dynamic(["teamviewer","anydesk"]);
let RATProcess =
DeviceProcessEvents
| where ProcessVersionInfoCompanyName has_any(KnownNames)
| extend DetectionMethod = "Process";
let RATMetwork =
DeviceNetworkEvents
| extend DNSRequest = (parse_json(AdditionalFields)).query,
DNSResponse = (parse_json(AdditionalFields)).answers, HTTP = (parse_json(AdditionalFields)).host
| where (
        RemoteUrl has_any (KnownNames) or
        RemotePort == 5938 or
        DNSRequest has_any (KnownNames) or
        DNSResponse has_any (KnownNames) )
| extend DetectionMethod = "Network";
union RATMetwork,RATProcess
```

Tiering misbehavior

Tier 0 **attempt/successful** logging on **to/from** Tier 1 systems

Tier 1 **attempt/successful** logging on **to/from** Tier 2 systems or Tier 0 systems

MITRE:T1021



Tiering misbehavior

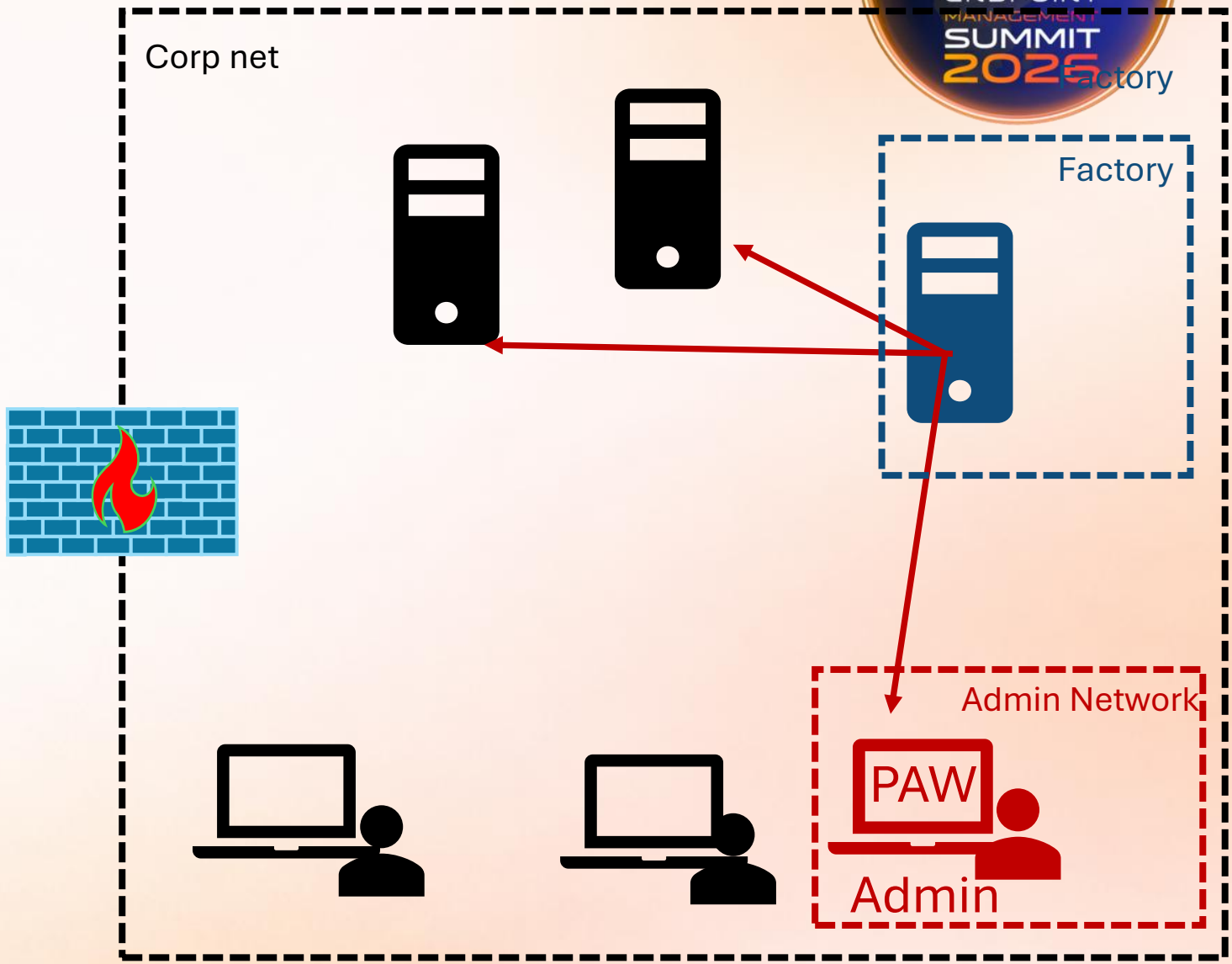

```
IdentityLogonEvents
| where Account startswith "a0-" or Account starts with "a1-“
| where ActionType == "RemoteInteractive" or ActionType == "Interactive“
| where DeviceName startswith "cli-"
```

Dynamically get Domain Administrators (to be used with querys related to Domain Admin activities such as logins)

```
ExposureGraphNodes
| extend json = (parse_json(NodeProperties)).rawData
| where json.nestedAdGroupNames has "Domain Admins“
| where NodeLabel == "user“
| mv-apply EntityIds on (summarize Identifiers = make_bag(pack(tostring(EntityIds.type),
EntityIds.id)))
| project AccountName = tostring(json.accountName)
| distinct AccountName
```

Unexpected network traffic

Traffic from factory network or other segmented or sensitive networks where no traffic is expected



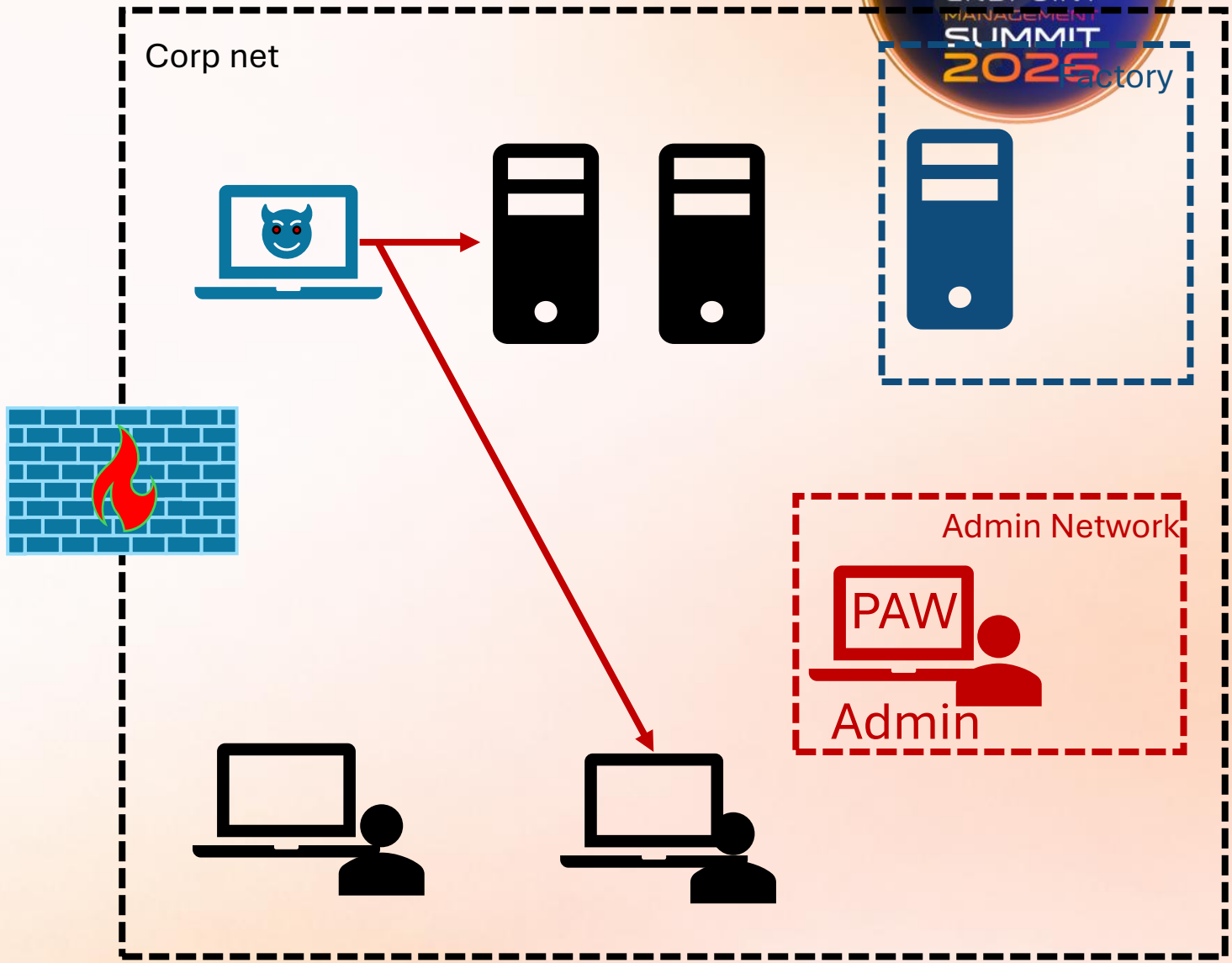
MITRE:TA0008

Unexpected network traffic


```
let MgmtPorts = dynamic([3389,22,5985]);
let UnexpectedNetwork = dynamic(["10.10.10","10.10.11"]); //Factory
DeviceNetworkEvents
| where LocalPort in(MgmtPorts)
| extend RemoteSubnet = extract(@"\d{1,3}\.\d{1,3}\.\d{1,3}",0,RemoteIP),
LocalSubnet = extract(@"\d{1,3}\.\d{1,3}\.\d{1,3}",0,LocalIP)
| where RemoteSubnet in(UnexpectedNetwork)
```

Traffic from unknown device

Traffic from devices which are not onboarded to Defender

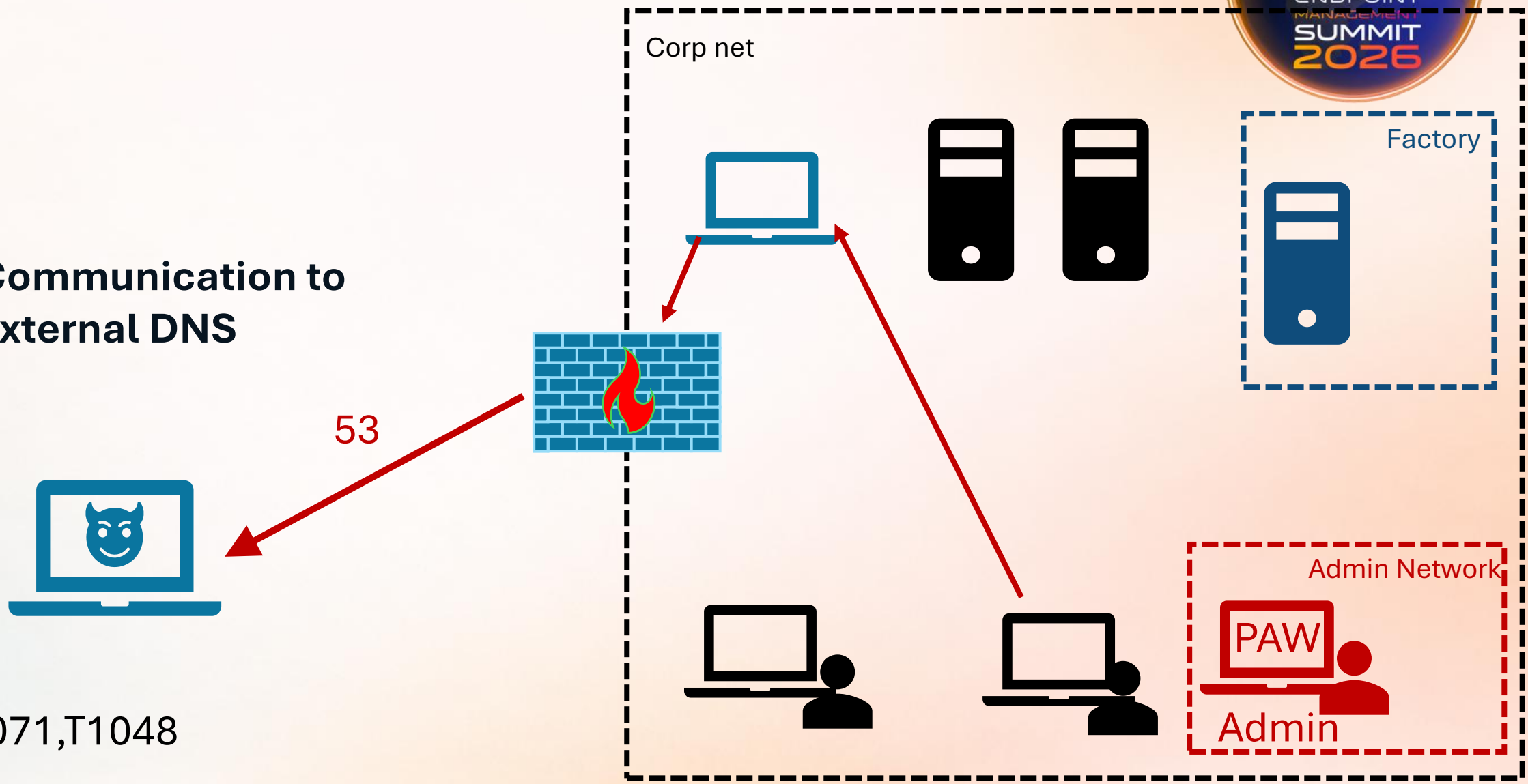


MITRE:T1200

Traffic from unknown device


```
DeviceNetworkEvents
| where ActionType == "InboundConnectionAccepted"
| where LocalPort in (3389,389,445,636,139,22) //Port filter or maybe add network filter
| extend IP = RemoteIP //IP is used in DeviceFromIP, and we want the remote IP
| invoke DeviceFromIP()
| where isempty(DeviceId1)
```

DNS Exfiltration or C2



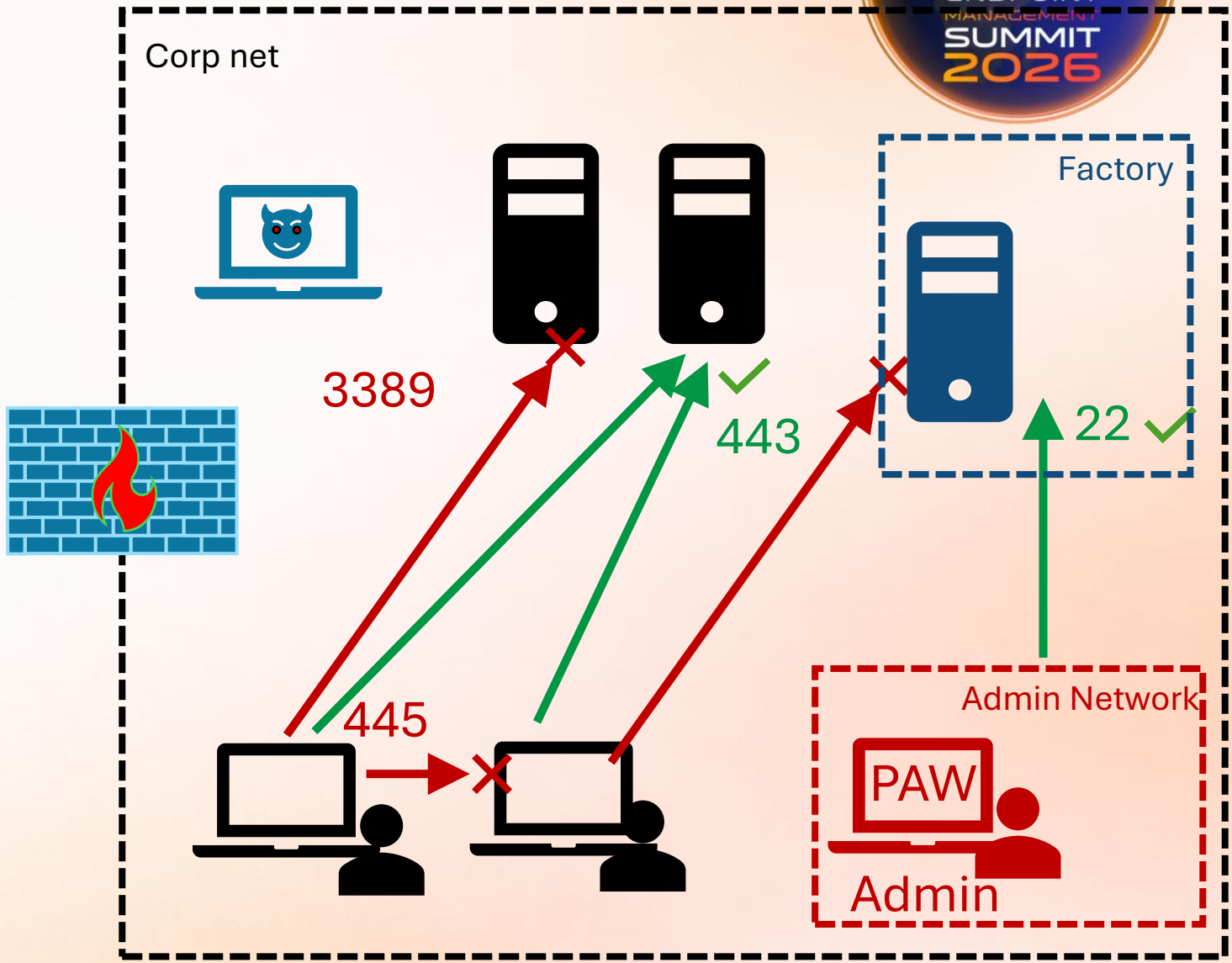
MITRE:T1071,T1048

DNS Exfiltration or C2


```
DeviceEvents
| where ActionType contains "DnsQueryResponse"
| extend jsondata = parse_json(AdditionalFields)
| extend DNSQuerySubDomain = jsondata["DnsQueryString"], QueryResult =
jsondata["DnsQueryResult"]
| mv-expand QueryResult
| extend DNSQueryType = parse_json(tostring(QueryResult)).DnsQueryType
| extend DNSQueryResult = parse_json(tostring(QueryResult)).Result
| project-away jsondata, QueryResult
| where DNSQueryType == "TEXT"
| extend DNSParentDomain = extract(@"((\w+|-){1,}.([^.]*))$",0,tostring(DNSQuerySubDomain))
| summarize NumberofSubDomains = dcount(tostring(DNSQuerySubDomain)) by
tostring(DNSParentDomain)
| where NumberofSubDomains > 5 //Trim
```

Remote Services

SMB
RDP
SSH
WINRM
...etc

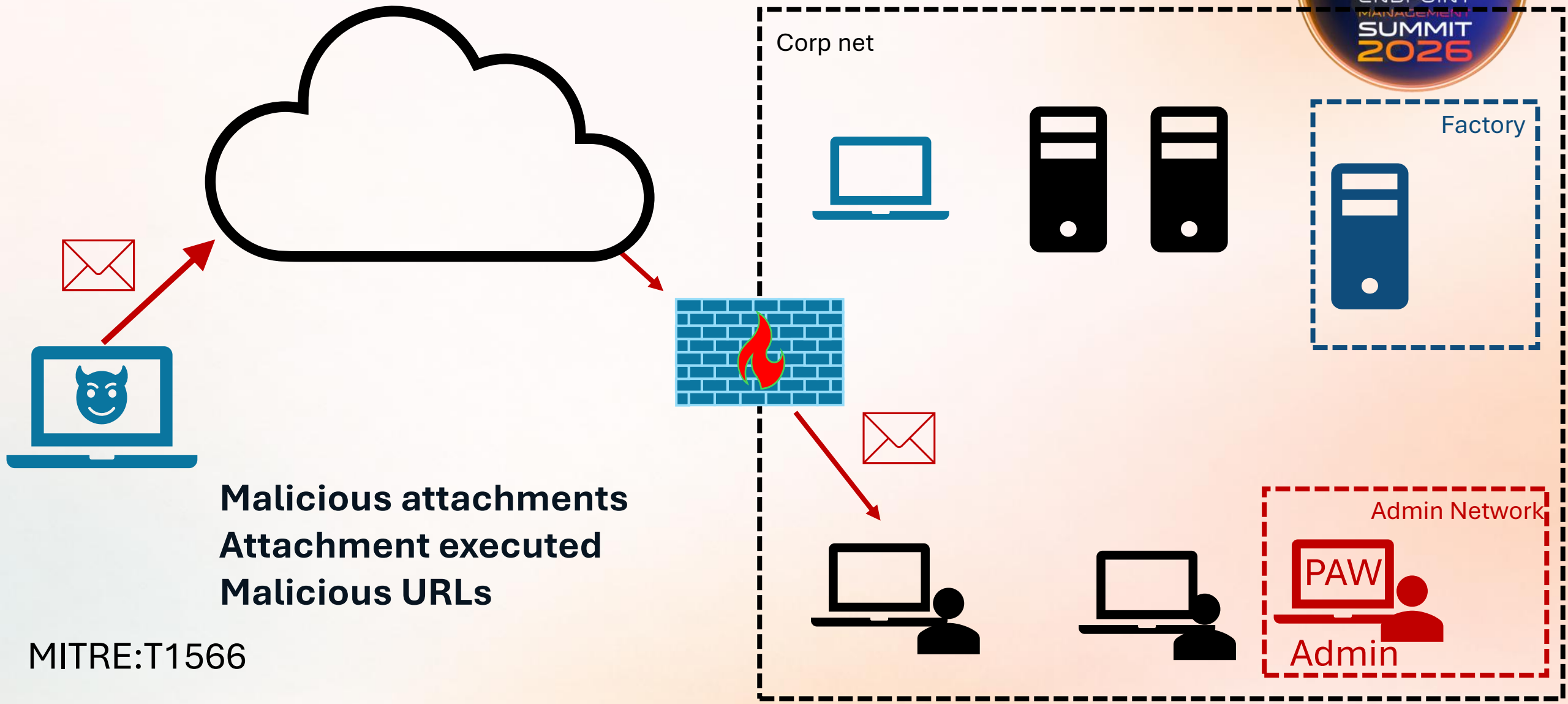


MITRE:T1021

Remote Services


```
let MgmtPorts = dynamic([3389,22,5985]); //Add mgmt. ports
let PAWNetwork = dynamic(["10.10.10","10.10.11"]); //PAWs
DeviceNetworkEvents
| where ActionType == "InboundConnectionAccepted"
| where LocalPort in(MgmtPorts)
| extend RemoteSubnet = extract(@"\d{1,3}\.\d{1,3}\.\d{1,3}",0,RemoteIP),
LocalSubnet = extract(@"\d{1,3}\.\d{1,3}\.\d{1,3}",0,LocalIP)
| where RemoteSubnet !in(PAWNetwork)
```

Malicious inbound e-mail



Malicious attachments
Attachment executed
Malicious URLs

MITRE:T1566

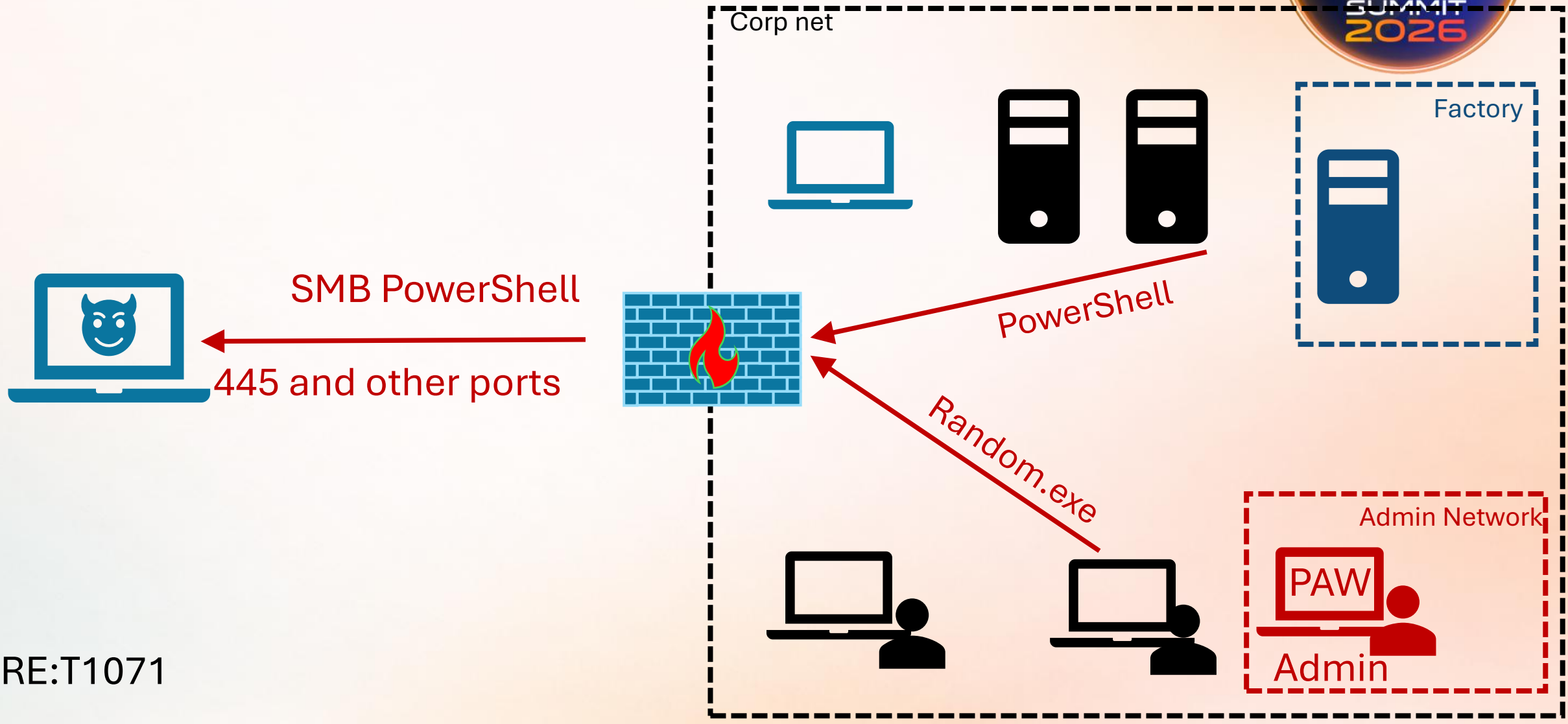
Malicious inbound e-mail


```
let phishurls =
externaldata(Domain:string,rem_foo:string,rem_foo1:string,URI:string)
[
  h@“<INSERT THREAT FEED HERE“
]
with(format="csv",ignoreFirstRecord=true)
|project-away rem*
| distinct Domain;
let safelink= "https://nam.safelink.emails.azure.net/redirect/?destination=";
EmailUrlInfo
| extend Url = iff(Url startswith
safelink,url_decode(tostring(split(Url,safelink,1)[0])),Url)
| extend UrlDomain = iff(UrlDomain ==
"nam.safelink.emails.azure.net",extract(@"https?://([^/]+)",1,Url),UrlDomain)
| where UrlDomain in~ (phishurls)
```

Malicious inbound e-mail


```
let includeurlenc = false; //change to true to include urlencoded mail domain as well (causes more false positives)
let regexString =
EmailEvents
| where EmailDirection == "Inbound“
| extend emaildomain = extract(@"@.*",0,RecipientEmailAddress) //Get all inbound email domains to prepare for regex and base64 encode
| extend urlencoded = url_encode(emaildomain)
| distinct emaildomain,urlencoded
| extend Base64 = replace("=","",base64_encode_tostring(emaildomain)) //padding chars removed
| distinct Base64,urlencoded
| summarize make_list(Base64),make_list(urlencoded) //make a list of all inbound email domains
| extend urlrexString = replace_regex(tostring(list_urlencoded),@'(",")',"|") //cleanup
| extend urlrexString = tostring(replace_regex(urlrexString,@'\"|\[|\]',"")) //cleanup
| extend base64rexString = replace_regex(tostring(list_Base64),@'","',"|") //cleanup
| extend base64rexString = tostring(replace_regex(base64rexString,@'\"|\[|\]',"")) //cleanup
| project-away list_Base64,list_urlencoded
| extend rexstring  = iff(includeurlenc == 1,strcat(@"(",urlrexString,@")|(",base64rexString,@")"),base64rexString) //Create regex
| extend b64regexString = strcat(@"https?:\/\/.*http.*(",rexstring,").*") //Final regex string matching http/https <anything> https <anything> base64 encoded email <anything>
| distinct b64regexString;
let potentialbadmails=EmailUrlInfo
| where Url matches regex toscalar(regexString)
| join (EmailEvents
| where Subject !contains "Newsletter“
| where DeliveryAction !contains "blocked") on NetworkMessageId
| where Url !contains "safelinks.protection.outlook.com“
;
potentialbadmails
| extend extractedDomain = extract(@'^https?://([A-Za-z0-9.-]+)(:[0-9]+)?/.*$',1,Url) //check if multiple domains remove to get more results if anything is missed
| extend removefirstdomain = replace_string(Url,extractedDomain,"") ) //check if multiple domains remove to get more results if anything is missed
| extend RedirectToNew = iff(removefirstdomain !contains extractedDomain,true,false) ) //check if multiple domains remove to get more results if anything is missed
| where RedirectToNew == 1 //check if redirect (experimental)
```

Outbound internet communication



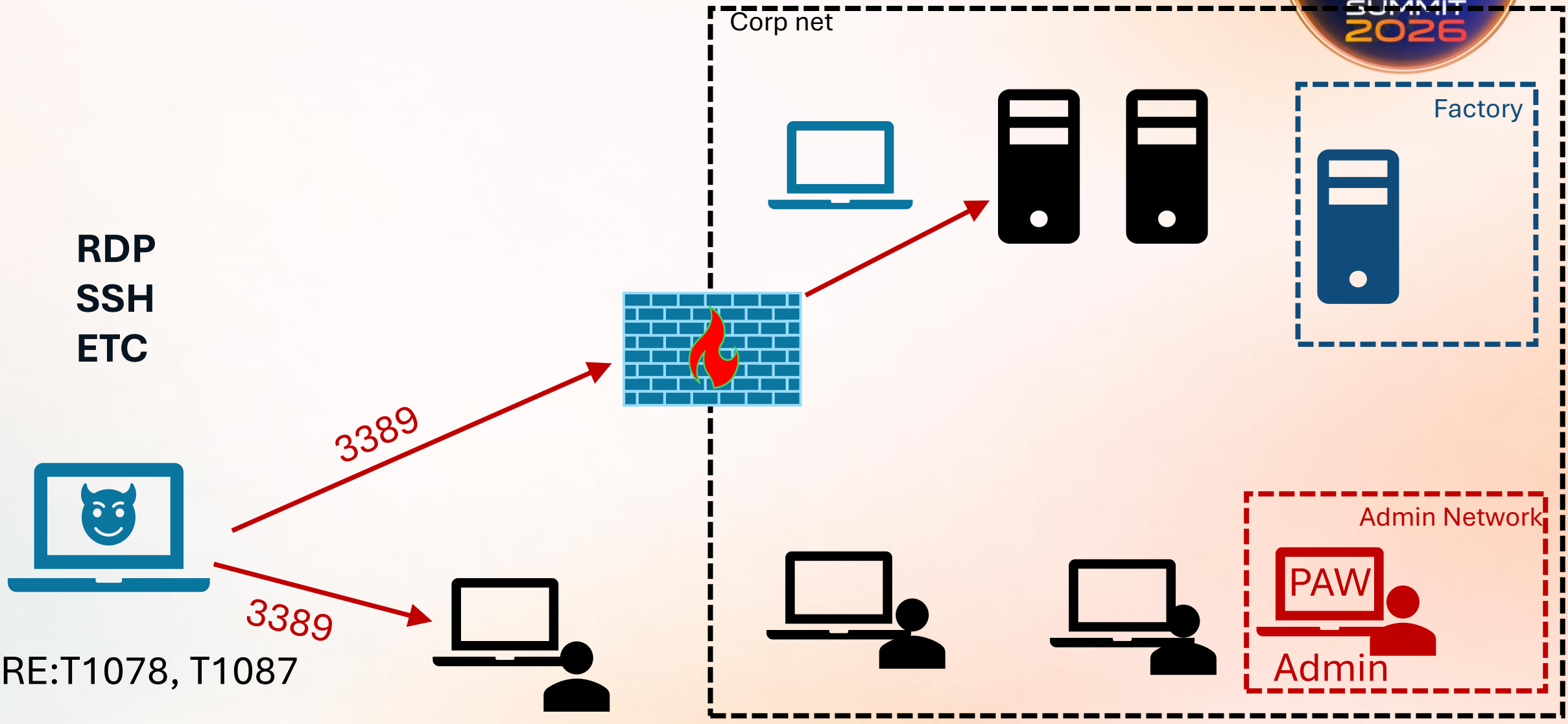
MITRE:T1071

Outbound internet communication


```
DeviceNetworkEvents
| where RemoteIPType == "PubliC"
| where RemotePort == "445" //Or use allowed port list
// Exclude
| where RemoteIP == "x.x.x.x"// MSP Hosted FileServer

DeviceNetworkEvents
| where InitiatingProcessFileName =~ "powershell.exe"
| where RemoteIPType == "Public"
//Exclude
// Some Admin running Exchange PowerShell from Jump server
```

DeviceLogon from Public IP



MITRE:T1078, T1087

DeviceLogon from Public IP


```
DeviceLogonEvents
| where ActionType == "LogonSuccess“ //Separate attempted and success
| where RemoteIPType == "Public"
```

Break-glass account usage

- Detect Usage of Break Glass Account AD/Entra ID

MITRE:T1078

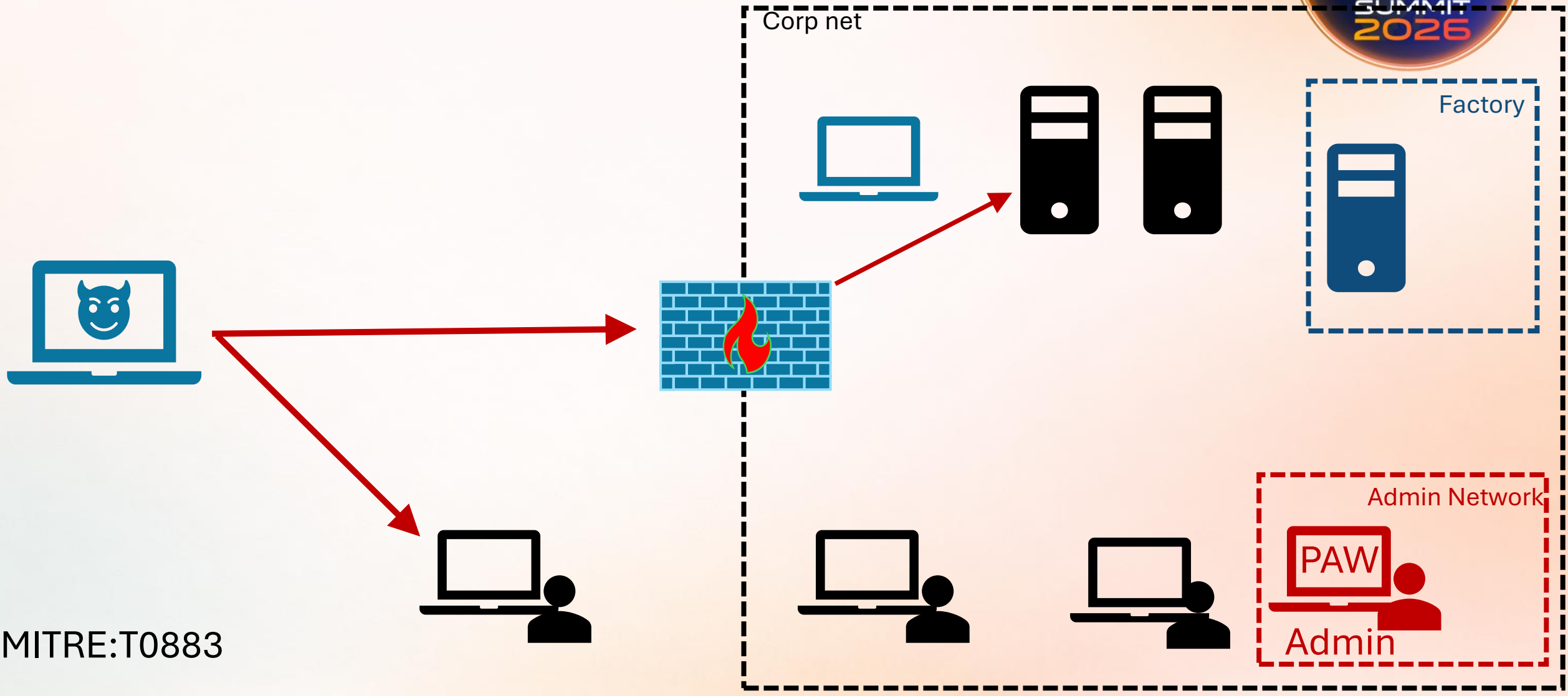
Break-glass account usage


```
IdentityLogonEvents
| where AccountName == "secret-break-glass-account"
| where ActionType == "LogonSuccess"



AADSignInEventsBeta
| where AccountUpn == "secret-break-glass-account@myupn.com"
| where ErrorCode == 0
```

Publicly Exposed devices



MITRE:T0883

Publicly Exposed devices


```
DeviceNetworkEvents
| where ActionType == "InboundConnectionAccepted"
| where RemoteIPType == "Public“

DeviceInfo
| where IsInternetFacing == 1
```

Helper queries

Contextualize data

Domain Admins


```
ExposureGraphNodes
| extend json = (parse_json(NodeProperties)).rawData
| where json.nestedAdGroupNames has "Domain Admins"
| where NodeLabel == "user"
| mv-apply EntityIds on (summarize Identifiers =
make_bag(pack(tostring(EntityIds.type), EntityIds.id)))
| project accountName =
json.accountName,Identifiers,accountEnabled=json.accountEnabled,disti
nguishedName=json.distinguishedName,adminCount=json.adminCount,Accoun
tControl=json.userAccountControl
```

Domain Admins


```
ExposureGraphNodes
| extend _jsonBlob = (parse_json(NodeProperties)).rawData
| extend Roles = _jsonBlob.deviceRole
| where Roles has "DomainController"
```

Some good places to find Custom Detections from the community

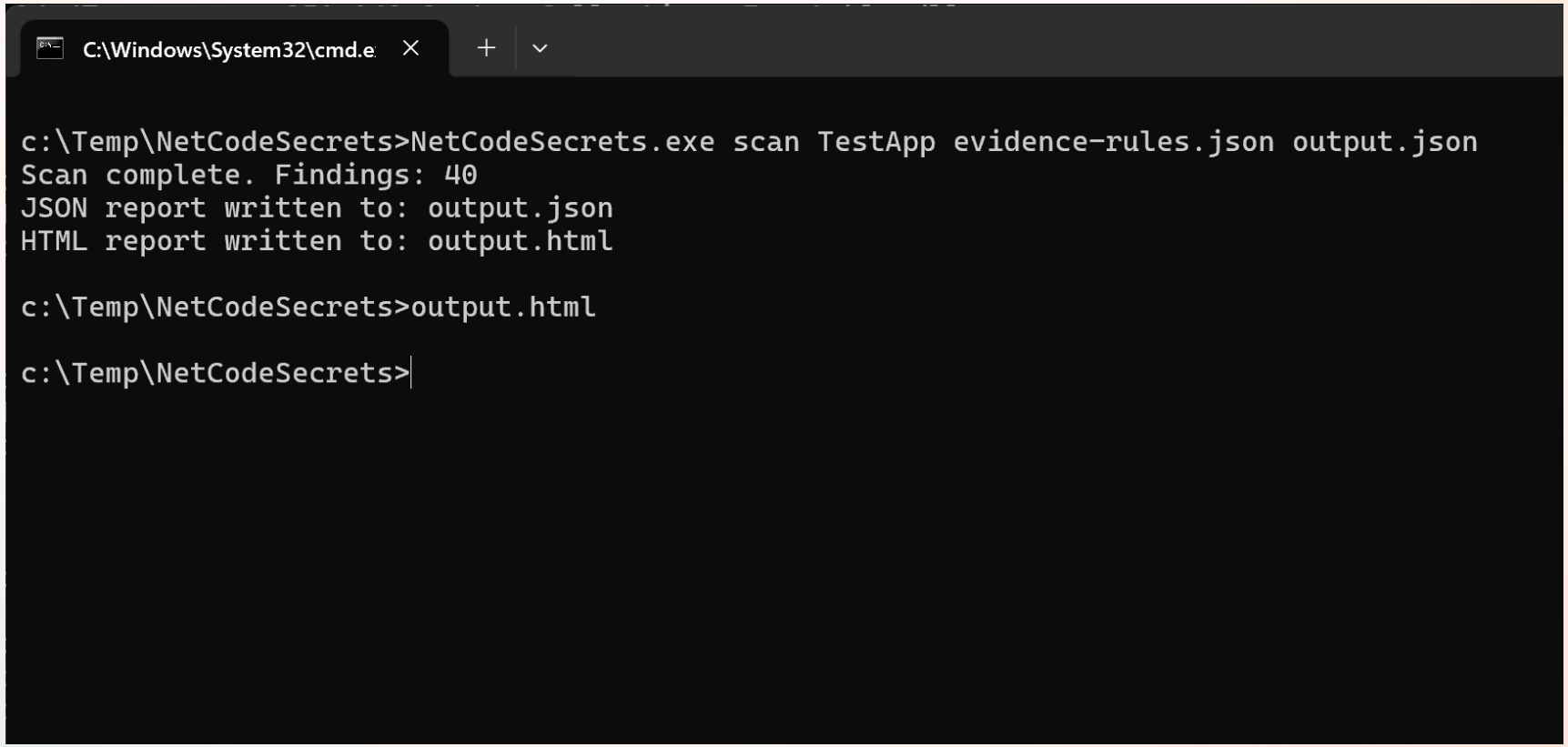
- http://blog.sec-labs.com / www.defenderboys.com
- https://github.com/bert-janp
- https://github.com/f-bader
- https://github.com/alexverboon
- https://github.com/falconforceteam

and more..

Wrap up and summary

- Start moving to Custom Detections
- Make use of MITRE ATT&CK
- Threat Modelling
  - Find the risks in your environment
- Purple Team Exercise
- Document your use-cases not only your detections
- Lifecycle Management of Detections is important
  - Relevant | Not Relevant | Microsoft Coverage

DEFENDER BOYS NETCODESECRETS



https://github.com/mattiasborg82/PublicTools/tree/main/Defender%20Boys/Bin/NetCodeSecrets

NetCodeSecrets

by Defender Boys | Dotnet Secrets Finder Tool

TOTAL FINDINGS	HIGH SEVERITY	MEDIUM SEVERITY	RULE MATCHES	ENTROPY MATCHES	HOST/FILE GROUPS
40	14	26	21	19	4

Rule Known pattern matched by a configured rule. Entropy Unknown high-entropy candidate that may require analyst review.

Host Summary Dashboard

Show all hosts

Host	Total	High	Medium	Entropy	Risk Score
LAPTOP-CSFTG61B	14	26	19	141	

FILTER FINDINGS

Search host, file path, rule, source, matched value, or snippet...

Examples: server01, appsettings.json, bearer-token, managed-strings, Password=, entropy

Host: LAPTOP-CSFTG61B

appsettings.json

Full path: c:\Temp\NetCodeSecrets\TestApp\appsettings.json

Findings: 27 | High: 9 | Medium: 18

Type	Rule	Severity	Confidence	Source	Matched Value	Snippet
Rule	aws-access-key-id	high	high	config	AKIA1234567890ABCDEF	}, "Aws": { "AccessKeyId": "AKIA1234567890ABCDEF", "SecretAccessKey": "aws_secret_a
Rule	aws-secret-access-key	high	high	config	aws_secret_access_key = abcdefghijklmnopqrstuvwxyzABCD1234567890	567890ABCDEF", "SecretAccessKey": "aws_secret_access_key = abcdefghijklmnopqrstuvwxyzABCD1234567890/+", "SessionToken": "aws_session_t
Rule	azure-sas-token	high	high	config	sv=2024-08-04&ss=b&srt=sco&sp=rwdlacupitfx&se=2027-12-31T23:59:59Z&st=2026-01-01T00:00:00Z&spr=https&sig=abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789%2F%2B%3D	VWXYZ0123456789+/==", "SasToken": "sv=2024-08-04&ss=b&srt=sco&sp=rwdlacupitfx&se=2027-12-31T23:59:59Z&st=2026-01-01T00:00:00Z&spr=https&sig=abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ012
Rule	azure-storage-account-key	high	high	config	AccountKey=abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789+/==	rotocol=https;AccountName=storagetest01;AccountKey=abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789+/==;EndpointSuffix=core.windows.net",

Defender Boys

Connect with Mattias

Twitter: https://twitter.com/mattiasborg82

LinkedIn: https://www.linkedin.com/in/mattiasborg82

Connect with Stefan

Twitter: https://x.com/stefanschorling

LinkedIn: https://www.linkedin.com/in/stefanschorling

Thank you for attending the session!

Please rate this session on Sched.com

We would love to hear what you liked and how we could improve!

Thanks!