KustoCon

Learn | Share | Practice

Practical Kusto for Defenders

Mattias Borg | Stefan Schörling

KustoCon
Learn | Share | Practice

Thanks to our Sponsors

baseVISION

glueck kanja

INVICTUS INCIDENT RESPONSE

water IT Security & Defense

Microsoft

scopewyse we are what's next

KustoWorks

SecMinds SOLUTIONS

KustoCon Learn | Share | Practice

Stefan Schörling

CTO - Onevinn

https://x.com/stefanschorling

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

Microsoft® MVP Most Valuable Professional

KustoCon
Learn | Share | Practice

Mattias Borg

Threat Hunter - Onevinn

https://x.com/matteborg82

https://www.linkedin.com/in/matteborg82/

Microsoft® MVP Most Valuable Professional

KustoCon
Learn | Share | Practice

Practical Kusto for Defenders

KustoCon
Learn | Share | Practice

Agenda

Session	Level	Goal	Important	Cleartext
PracticalKQL	Essentials	StartUsingKusto	1	
PracticalKQL	Intermediate	ImproveYourKusto	1	
PracticalKQL	Expert	StepUp	1	[redacted]

```
let AgendaStep1 =
  print Session = "PracticalKQL"
  | extend Level = "Essentials",
           Goal = "StartUsingKusto",
                  Important = true;
let AgendaStep2 =
  print Session = "PracticalKQL"
        Level = "Intermediate" ,
        Goal = "ImproveYourKusto" ,
        Important = true;
let AgendaStep3 =
  print Session = "PracticalKQL"
  | extend Level = "Expert" ,
           Goal = "StepUp" ,
           Important = true , //ish
           Cleartext = DB_XOR(" 051!/o! 2&}7 & ","KUSTO");
union AgendaStep1, AgendaStep2, AgendaStep3
```

KustoCon
Learn | Share | Practice

KQL Essentials

The where, The how and The what!

KustoCon
Learn | Share | Practice

Solutions supporting Kusto

Microsoft Fabric

Log analytics

Microsoft Sentinel

Azure Data Explorer

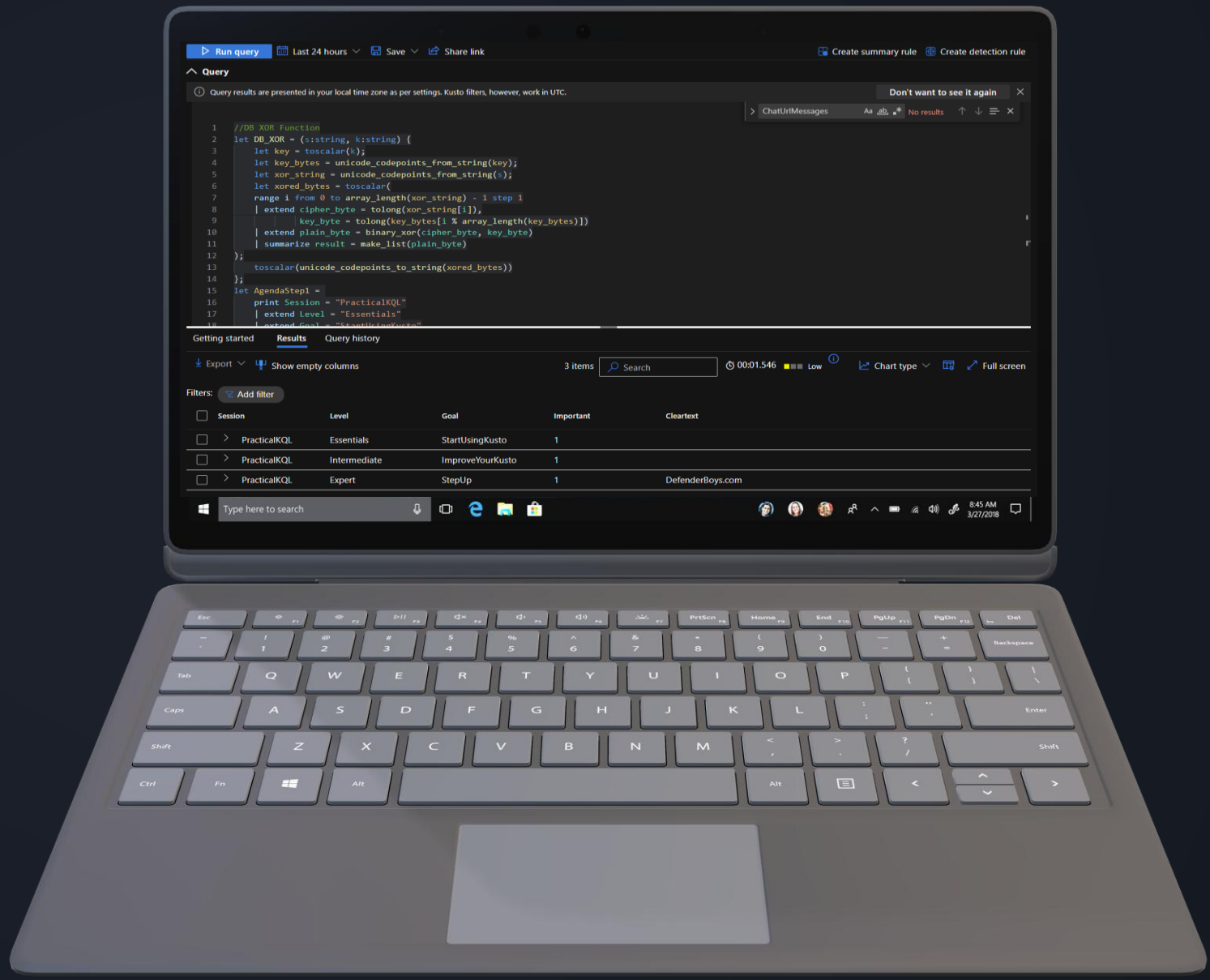
Microsoft Defender XDR

Azure Workbooks

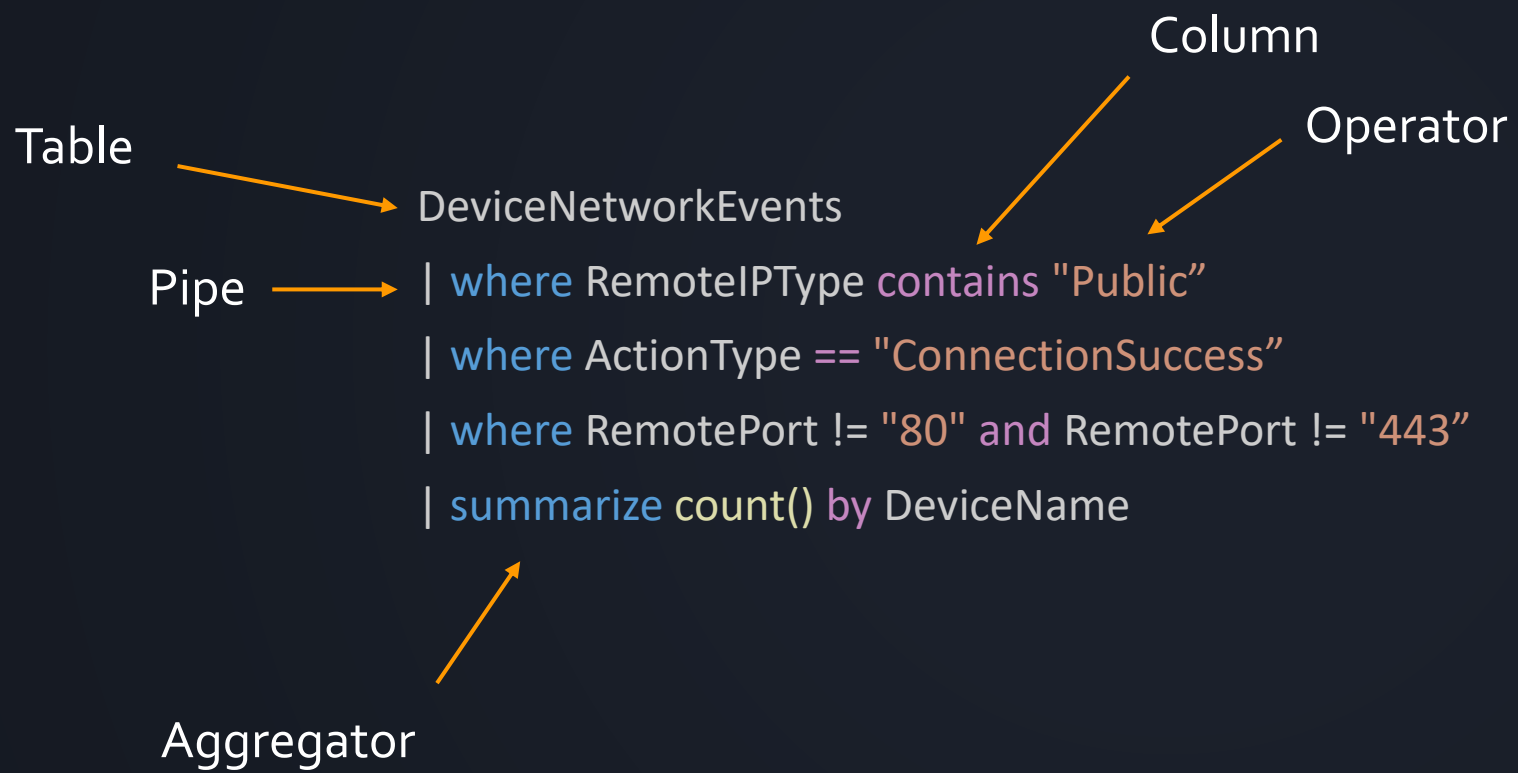
And many more…

How it works

Jumping into the language



Kusto Query Elements



Syntax

Operator	Explantion
where	Filters rows of data based on certain criteria
project	Simplify the view and select a specific subset of columns
distinct	Only show unique values
order by / sort by column	Arrange the rows in descending order (can use asc)
take	Allows you to specify an arbitrary number of records to return
top n by column	Returns the first n rows sorted by the specified column

Syntax

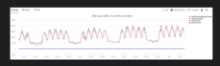
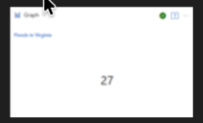
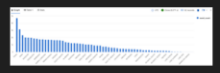
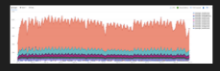
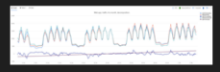
Operator	Explanation
==, =~	Equals (case-sensitive) Equals (not case-sensitive)
!=, !~	Not equals (case-sensitive) Not equals (not case-sensitive)
>, >=, <, <=	Greater than/less than
in, !in in~, !in~	In/Not in (case-sensitive) In/Not in (not case-sensitive) Filter records by checking if a value matches any value in a specified list
between	Filter records that fall within an inclusive range of values
case	Simplify complex comparisons

Syntax

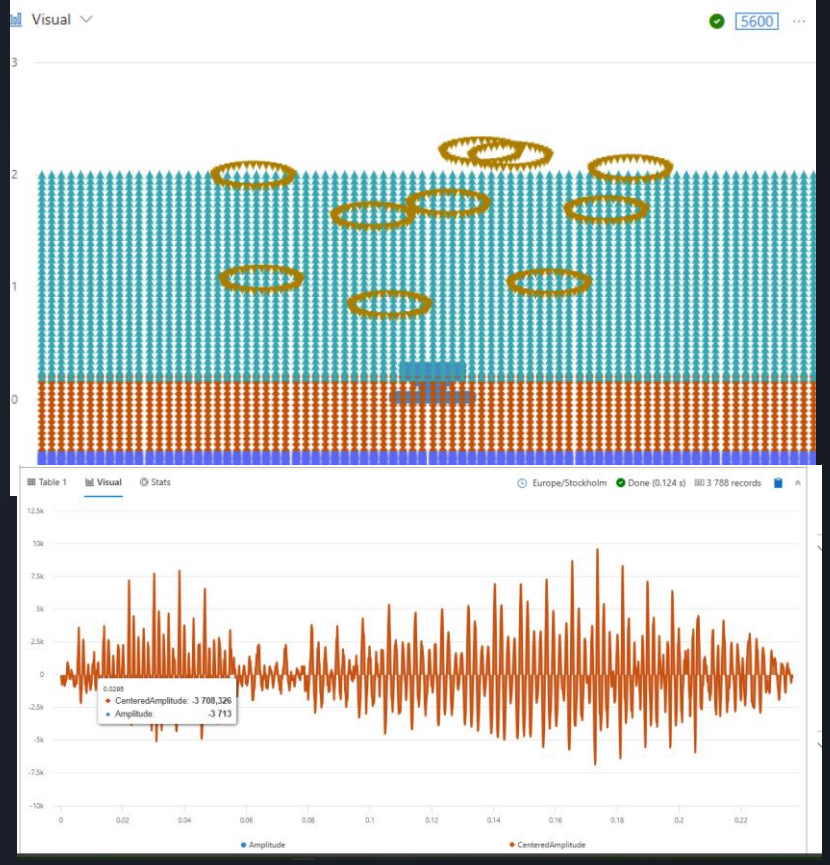
Operator	Explanation
count	Number of times the value appears in the dataset
max / min	Highest/lowest value that appears in the dataset for that column (returns max/min value only)
arg_max / arg_min	Highest/lowest value that appears in the dataset for that column (allows additional columns)
sum	Adds values from different column together, you specify the columns
bin	Rounds down values based on based on a specific criteria

Render

- anomalychart
- areachart
- barchart
- card
- columnchart
- linechart
- piechart
- scatterchart
- stackedareachart
- table
- timechart

Visualisierung	Beschreibung	Abbildung
anomalychart	Ähnlich wie das Zeitdiagramm, hebt jedoch Anomalien mithilfe series_decompose_anomalies Funktion hervor.	
areachart	Bereichsdiagramm.	
barchart	als horizontale Streifen angezeigt.	
card	Der erste Ergebnisdatensatz wird als Satz von Skalarwerten behandelt und als Karte angezeigt.	
columnchart	Wie `barchart` mit vertikalen Streifen anstelle von horizontalen Streifen.	
linechart	Liniendiagramm.	
piechart	Die erste Spalte ist Farbachse, zweite Spalte ist numerisch.	
scatterchart	Punktdiagramm.	
stackedareachart	Gestapeltes Flächendiagramm.	
table	Ergebnisse werden standardmäßig als Tabelle angezeigt.	
timechart	Liniendiagramm. Die erste Spalte ist x-Achse und muss datumstime sein. Andere (numerische) Spalten werden als y-Achsen verwendet.	

```
385 Sky
386 | union Ocean
387 | union Sand
388 | union Clouds
389 | union BoatHull
390 | union BoatSail
391 | render scatterchart with (xcolumn=x, ycolumns=y, series=col
392
```



KustoCon
Learn | Share | Practice

.drop Table MyTable

KustoCon
Learn | Share | Practice

Statistical Modelling

Statistical Modeling

the mathematical relationship between random and non-random variables

Visualize data

Identify

Predict

Anomaly basics

W1

W2

W3

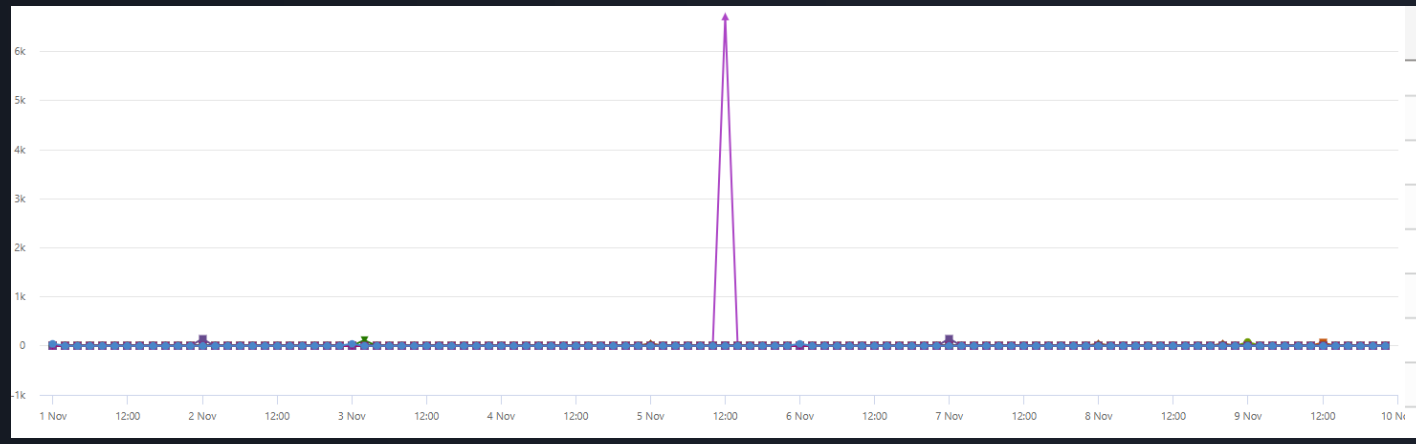
W4

Anomalies

make-series

series_decompose_anomalies()

...and tons of others



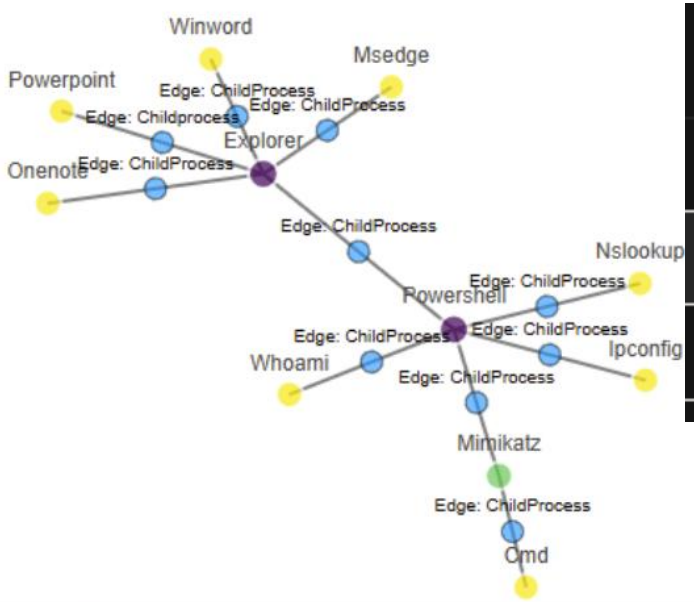
num	timestamp	anomalies	score	baseline
8	11/8/2025, 8:00:00 PM	1	47.44878319531561	0
4	11/6/2025, 12:00:00 AM	1	42.18467670496769	0
4	11/3/2025, 12:00:00 AM	1	41.045249743800056	0
4	11/1/2025, 12:00:00 AM	1	40.2856317696883	0
0	11/1/2025, 12:00:00 AM	0	0.3141854482876959	0
0	11/1/2025, 12:00:00 AM	0	0.3130881766405912	0
0	11/9/2025, 10:00:00 PM	0	0.31055744080169473	0
0	11/1/2025, 12:00:00 AM	0	0.31055744080168757	0
0	11/1/2025, 2:00:00 AM	0	0.2817951958869025	0

Graph

```
let NetworkNodes =
DeviceNetworkInfo
| mv-apply todynamic(IPAddresses) on (
                        summarize IPIDs = make_bag(
                                                    pack("AddressType",
                                                            tostring(IPAddresses.AddressType),
                                                            "IPAddress",
                                                            IPAddresses.IPAddress,
                                                            "SubnetPrefix",
                                                            IPAddresses.SubnetPrefix
                                                        )
                                                )
                    )
| extend IPAddress = tostring(IPIDs.IPAddress),AddressType = tostring(IPIDs.AddressType),SubnetPrefix = tostring(IPIDs.SubnetPrefix)
| where AddressType != "LinkLocal" | project-away IPIDs
| join (DeviceInfo) on DeviceId | project Timestamp, DeviceId, DeviceName, MacAddress, IPAddress,AddressType,SubnetPrefix, OSPlatform, PublicIP;
let NetworkEdges =
DeviceNetworkEvents
| where LocalPort == 3389
| where ActionType == "InboundConnectionAccepted"
| where LocalIP !contains ":"
| summarize arg_max(Timestamp,*) by RemoteIP, LocalIP
| project Timestamp, SourceIP=RemoteIP, DestinationIP=LocalIP;
NetworkEdges
| make-graph SourceIP --> DestinationIP with NetworkNodes on IPAddress
| graph-match (Source)<-[reports*2..5]-(Target)
project Target.IPAddress, flow = map(reports, SourceIP)
| extend CountJumps = array_length(flow)
| sort by CountJumps desc
| take 3
```

Graph

Graph Visualizer



Target_IPAddress	flow	CountJumps
10.243.0.21	["10.10.1.95","10.11.14.148","10.241.2.4","10.243.0.21"]	4
10.243.0.21	["10.10.1.95","10.11.14.148","10.241.2.4","10.243.0.21"]	4
10.243.0.21	["10.10.1.95","10.11.14.148","10.241.2.4","10.243.0.21"]	4

detect_anomalous_spike_fl()

> Applies to: ✅ Microsoft Fabric ✅ Azure Data Explorer ✅ Azure Monitor ✅ Microsoft Sentinel

Detect the appearance of anomalous spikes in numeric variables in timestamped data.

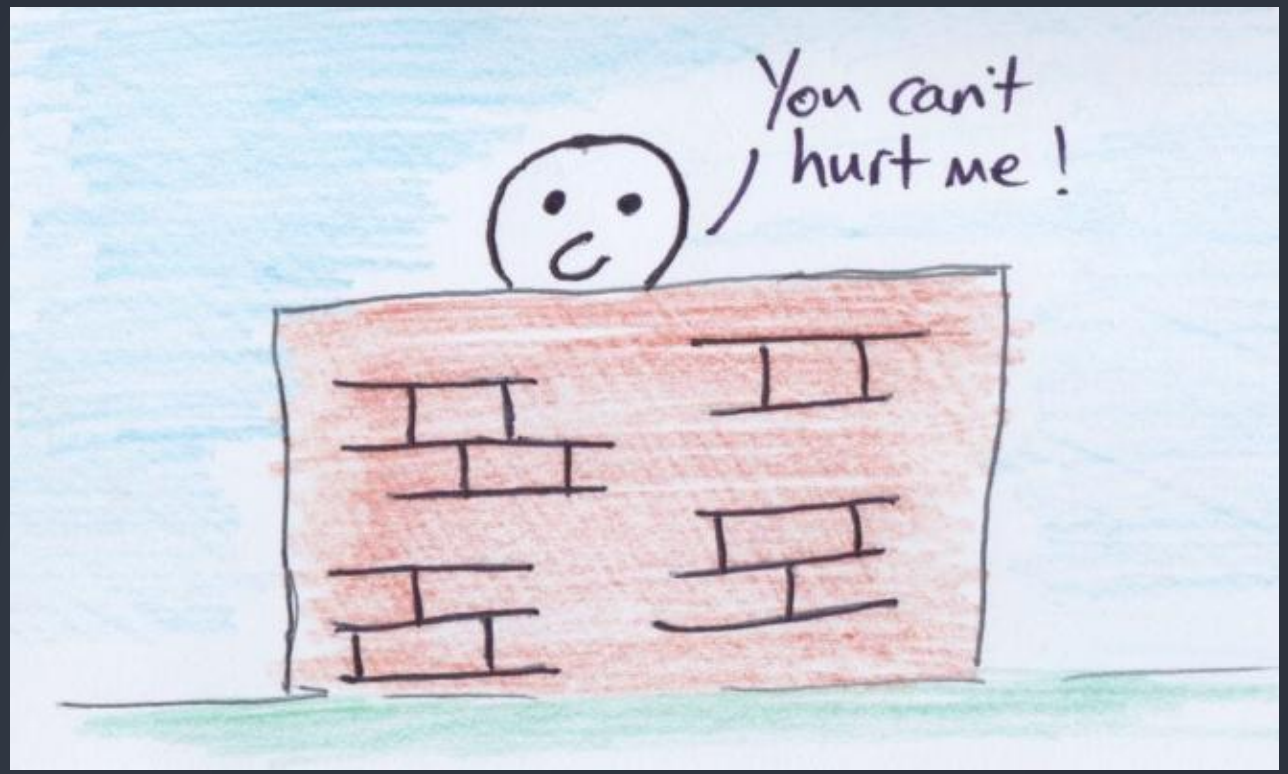
The function `detect_anomalous_spike_fl()` is a UDF (user-defined function) that detects the appearance of anomalous spikes in numeric variables - such as amount of exfiltrated data or failed sign in attempts - in timestamped data, such as traffic logs. In cybersecurity context, such events might be suspicious and indicate a potential attack or compromise.

The anomaly model is based on a combination of two scores: Z-score (the number of standard deviations above average) and Q-score (the number of interquantile ranges above a high quantile). Z-score is a straightforward and common outlier metric; Q-score is based on Tukey's fences - but we extend the definition to any

SCENARIOS and WHY

KustoCon
Learn | Share | Practice

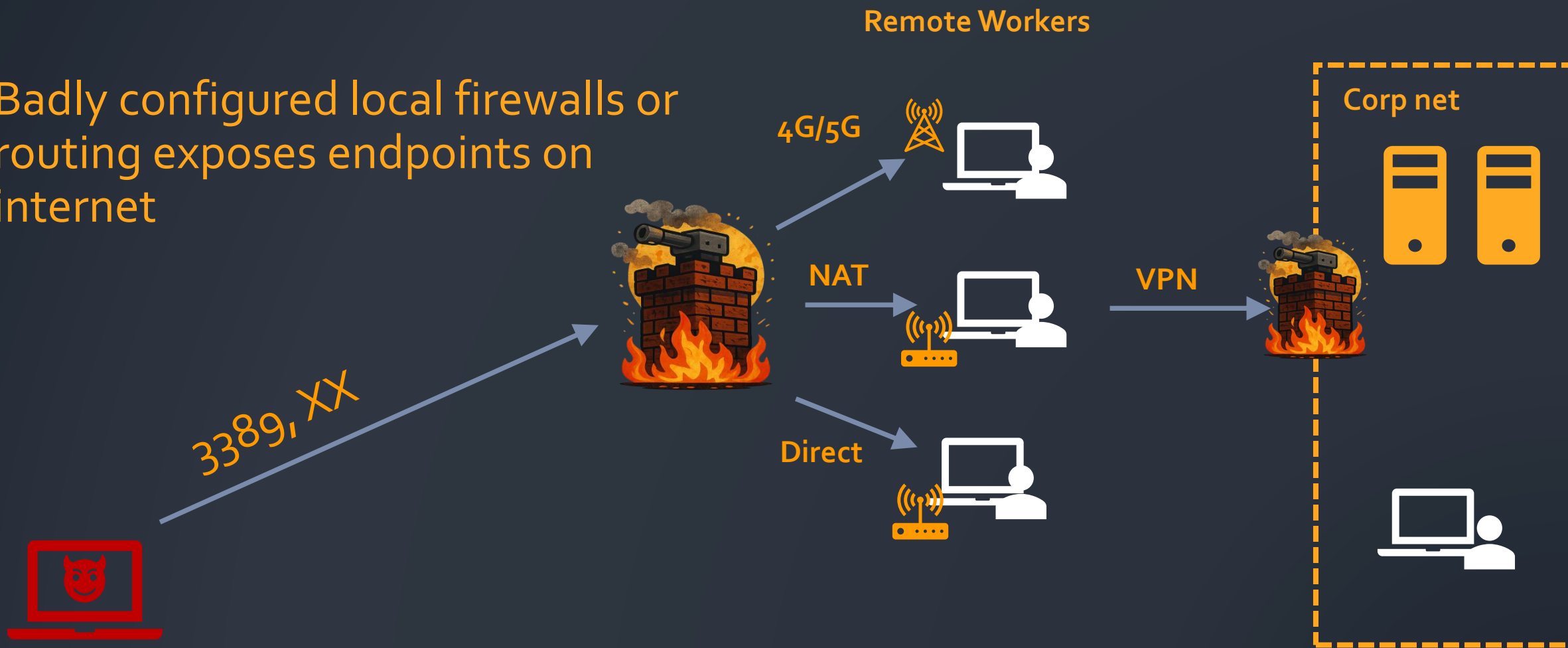
KQL for the Client Admin



KustoCon
Learn | Share | Practice

Local Firewall != "OK"

Badly configured local firewalls or routing exposes endpoints on internet

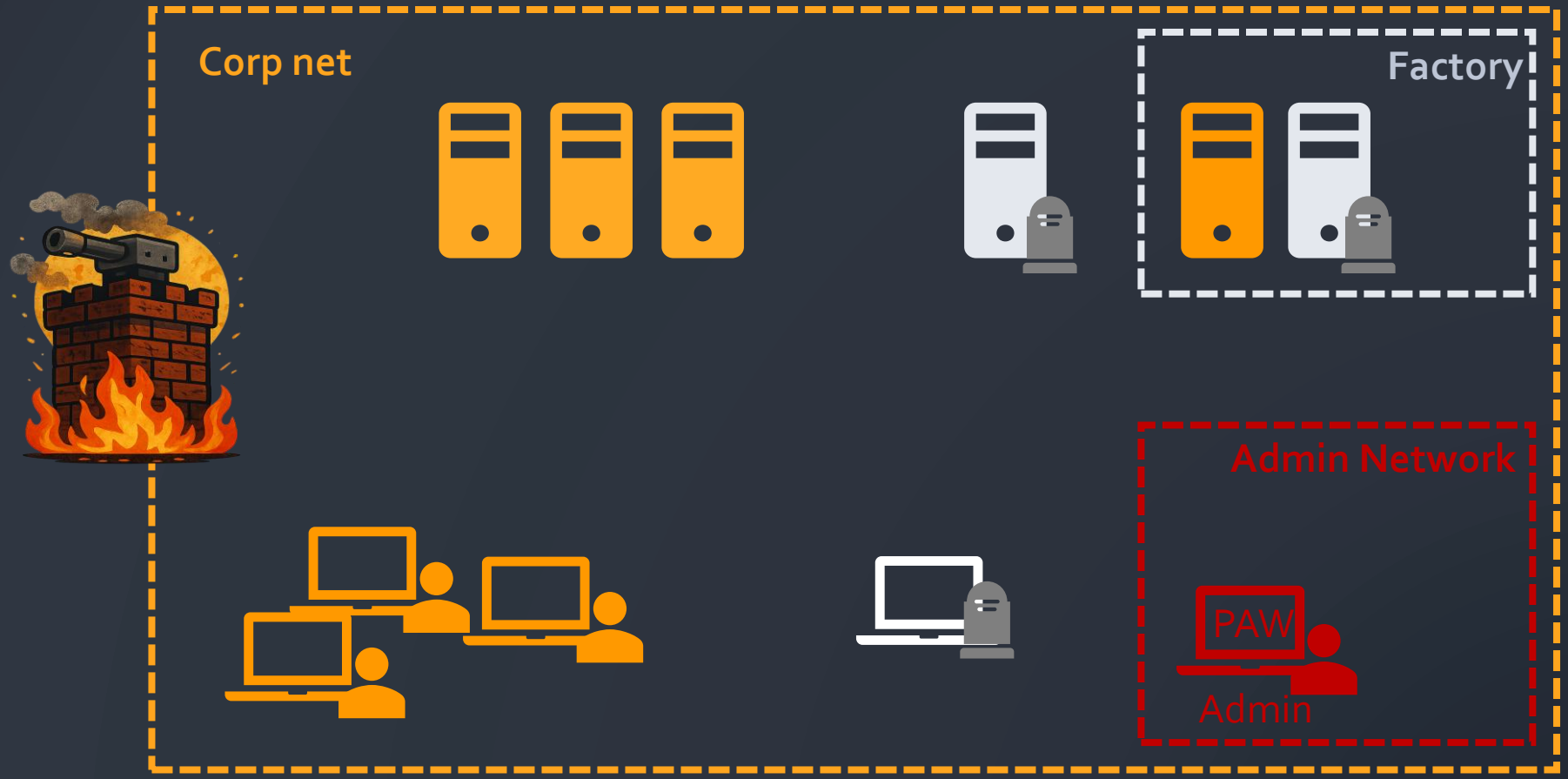


KQL - Local Firewall != "OK"

```
DeviceNetworkEvents
| where RemoteIPType != "Private"
| where LocalIP != "::ffff:127.0.0.1"
| where ActionType contains "InboundConnectionAccepted"

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

Unmanaged Devices



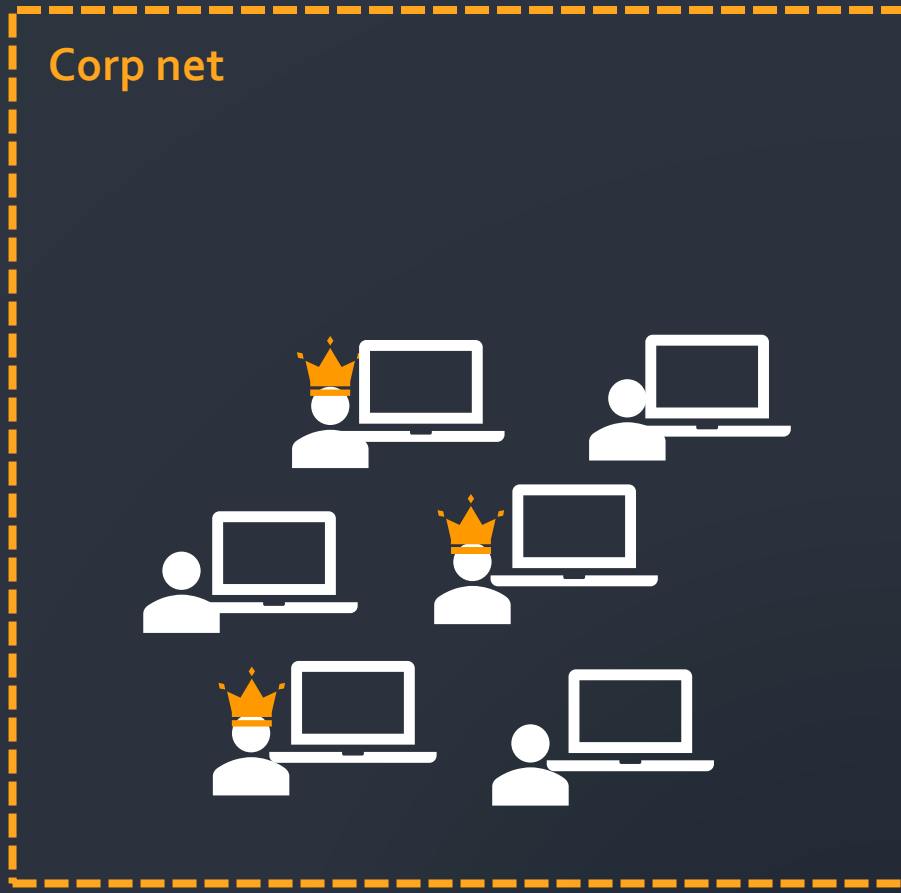
KQL – Unmanaged Devices

```
DeviceInfo
| where OnboardingStatus != "Onboarded"
| summarize arg_max(Timestamp, DeviceName, OSPlatform, OnboardingStatus, DeviceType, DiscoverySources) by DeviceId
| extend _tmp = parse_json(DiscoverySources)
| evaluate bag_unpack(_tmp)
```

DeviceId	Timestamp	OSPlatform	OnboardingStatus	DeviceType	DiscoverySources	Defender for Endpoint	Defender for Identity
f03721261203c...	Nov 5, 2025 9:57:47 PM		Insufficient info	Unknown	{"Defender for Endpoint...	Nov 5, 2025 1:00:00 AM	
bf4b0919598ef...	Nov 5, 2025 10:09:06 PM	Linux	Insufficient info	Unknown	{"Defender for Endpoint...	Nov 5, 2025 1:00:00 AM	
e41e1d43fbc8b...	Nov 5, 2025 11:23:32 PM	Linux	Unsupported	NetworkDevice	{"Defender for Endpoint...	Nov 5, 2025 1:00:00 AM	
34eb41473c5d...	Nov 5, 2025 11:45:21 PM	Windows11	Can be onboarded	Workstation	{"Defender for Endpoint...	Nov 5, 2025 1:00:00 AM	Oct 1, 2025 2:00:00 AM
f4a102042a1fe...	Nov 6, 2025 12:06:33 PM	Linux	Unsupported	Communication	{"Defender for Endpoint...	Nov 6, 2025 1:00:00 AM	
cd81e8b7263c...	Nov 6, 2025 1:41:13 AM		Unsupported	Printer	{"Defender for Endpoint...	Nov 6, 2025 1:00:00 AM	
54ceeeb1c9ad2...	Nov 6, 2025 1:46:28 AM		Insufficient info	Unknown	{"Defender for Endpoint...	Nov 5, 2025 1:00:00 AM	
b1a6b64fd57f4...	Nov 5, 2025 11:00:19 PM		Insufficient info	Unknown	{"Defender for Endpoint...	Nov 5, 2025 1:00:00 AM	

Local Admins

```
DeviceLogonEvents
| where IsLocalAdmin == "1”
| where DeviceName startswith "cl-”
| summarize dcount(DeviceName) by AccountName
```

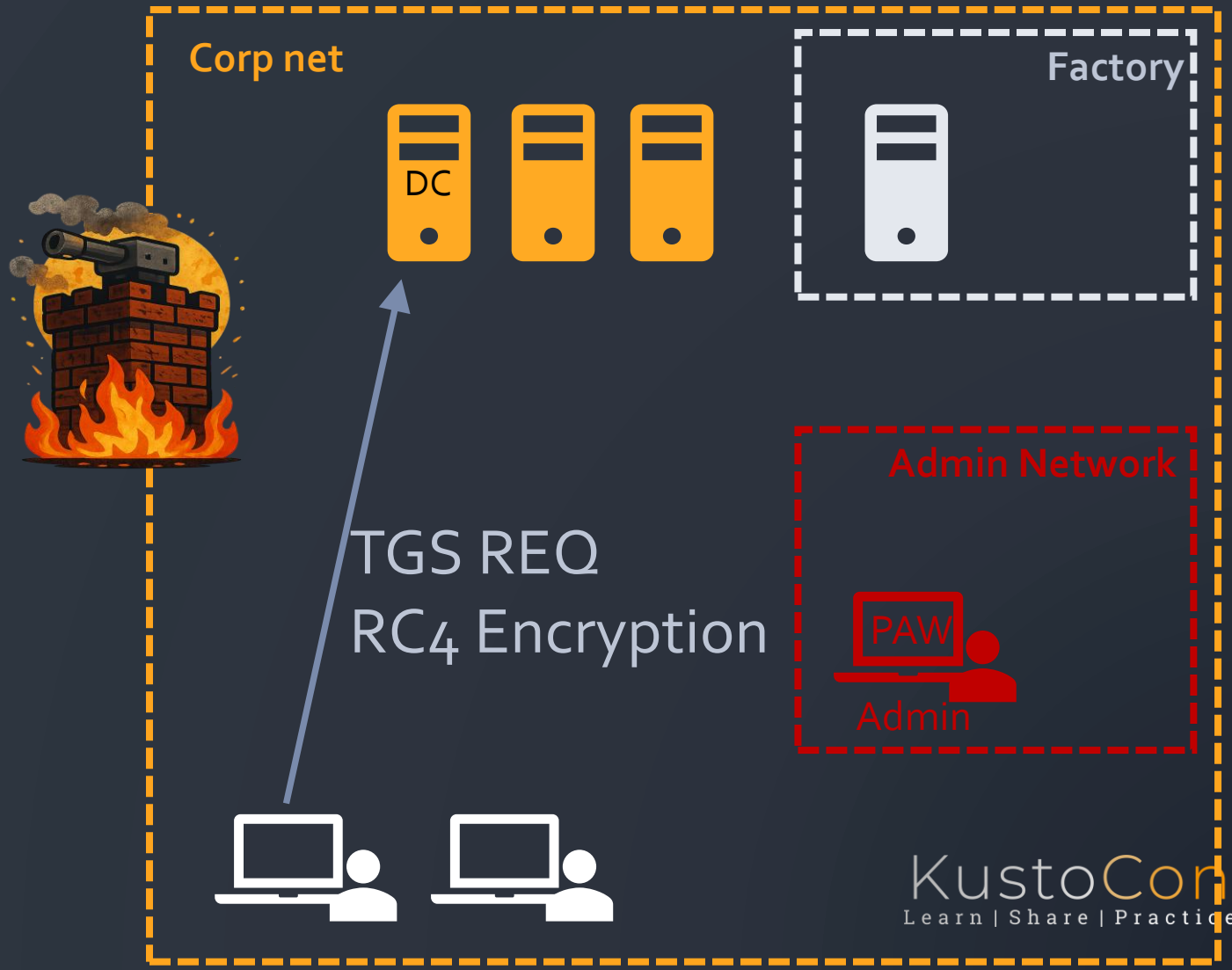


KQL for the Identity Admin

KustoCon
Learn | Share | Practice

Identity Scenario

Find use of
RC4 encryption type for
Service accounts / SPNs

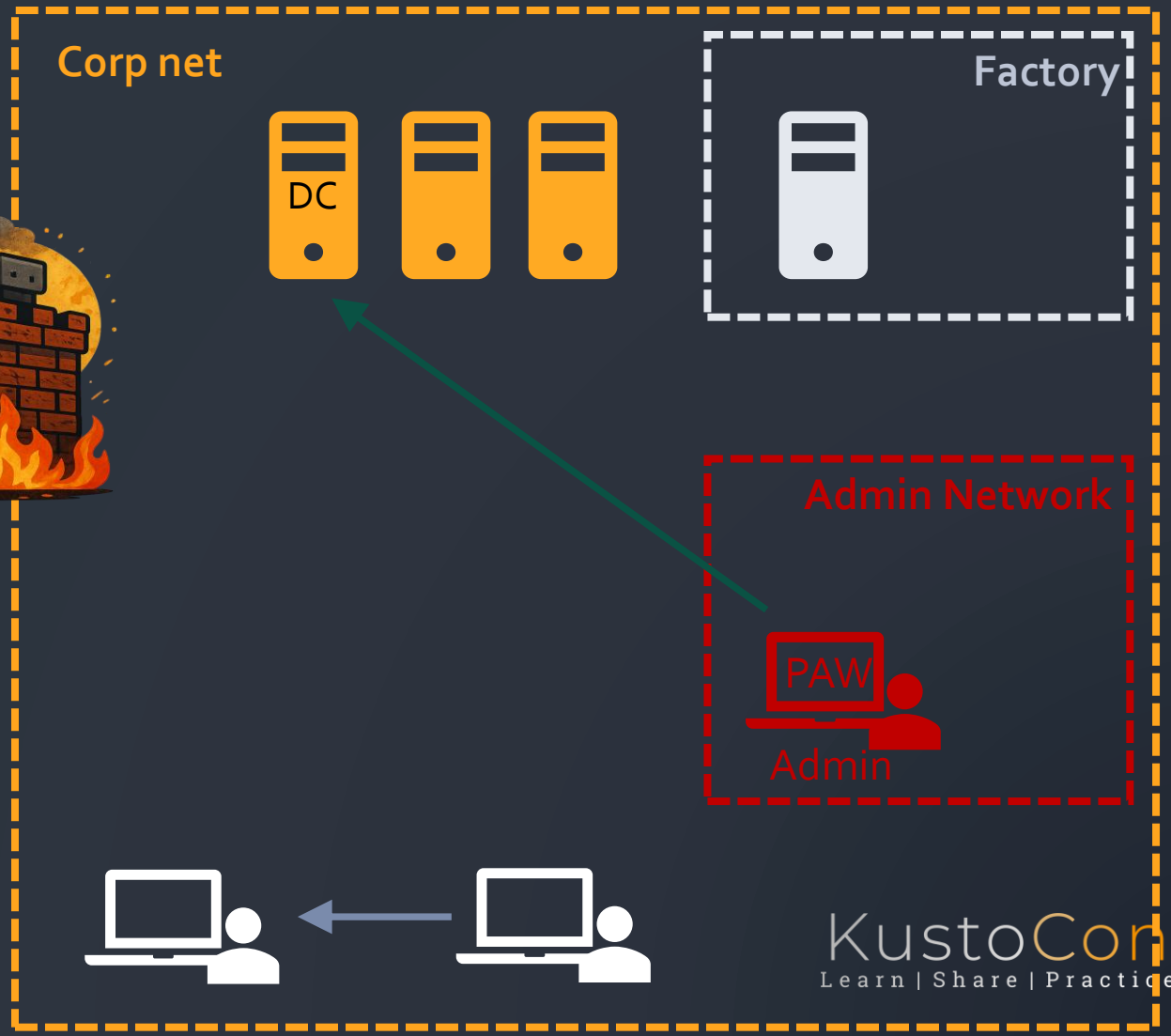


KQL – RC4

```
IdentityLogonEvents
| where LogonType == "Resource access"
| extend KerberosType = tostring(AdditionalFields.KerberosType),
         EncryptionType = tostring(AdditionalFields.EncryptionType),
         SPN = tostring(AdditionalFields.Spns),
         KdcOptions = tostring(AdditionalFields.KdcOptions),
         isNtlmV1 = tobool(AdditionalFields.KdcOptions),
         SourceAccountId = tostring(AdditionalFields.SourceAccountId),
         SourceAccountSid = tostring(AdditionalFields.SourceAccountSid)
| where EncryptionType == "rc4hmac"
| extend SPNType = extract(@"^([^/]+)",0,SPN)
| where SPNType in("MSSQLSvc","HTTP") // Filter if needed
```

Tier abuse

Find Identity use outside the Tier Model



KQL – Tier abuse

```
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
```

KQL – Tier abuse

Dynamically get Domain Administrator logons on none-Domain Controllers

```
let DCs = ExposureGraphNodes
| extend _jsonBlob = (parse_json(NodeProperties)).rawData
| distinct DeviceName = tolower(_jsonBlob.deviceName);
let DAs =
ExposureGraphNodes
| extend json = (parse_json(NodeProperties)).rawData
| where json.nestedAdGroupNames has "Domain Admins"
| where NodeLabel == "user"
| distinct tostring(json.accountName);
DeviceLogonEvents
| where AccountName has_any (DAs)
| where DeviceName !in(DCs)
| summarize DistinctDevices = dcount(DeviceName),
            NetworkSuccessLogons = countif(ActionType == "LogonSuccess" and LogonType
== "Network"),
            RemoteInteractiveSuccessLogons = countif(ActionType=="LogonSuccess" and
            LogonType =="RemoteInteractive"),
            TotalRemoteDeviceNames = dcount(RemoteDeviceName),
            RemoteIPs = dcount(RemoteIP),
            FailedLogins =countif(ActionType == "LogonFailed"),
            AttemptedLogins = countif(ActionType=="LogonAttempted"),
            ProtocolsUsed = dcount(Protocol)
            by AccountName
```

KustoCon
Learn | Share | Practice

KQL for the Messaging Admin

KustoCon
Learn | Share | Practice

Email Transport Rules

Transport rules allowing some emails for a few users

KQL – AntiSpam Tuning

```
EmailEvents
| where EmailDirection == "Inbound"
| where DeliveryLocation has "inbox"
| extend isThreat = iff(isnotempty(ThreatTypes),1,0),
         isOrgLevelPolicy = iff(isnotempty(OrgLevelPolicy),1,0)
| project Timestamp, AdditionalFields, OrgLevelPolicy, SenderFromDomain,
ThreatTypes,SenderFromAddress, RecipientDomain, RecipientEmailAddress, isThreat,
isOrgLevelPolicy
| summarize CountEmailsBySenderDomain = count(),
            CountRecipients = dcount(RecipientEmailAddress),
            CountRecipientDomains = dcount(RecipientDomain),
            CountPoliciesByDomain = dcountif(OrgLevelPolicy, isOrgLevelPolicy==true),
            CountEmailsWithThreats = countif(isThreat==true)
            by SenderFromDomain
            | where CountEmailsWithThreats > 0
            | extend PercentHits =
strcat(tostring(toint((todouble(CountEmailsWithThreats) /
todouble(CountEmailsBySenderDomain)) * 100)),"%")
            | sort by PercentHits
```

KustoCon
Learn | Share | Practice

KQL – AntiSpam Tuning

```
EmailEvents
| where EmailDirection == "Inbound"
| where DeliveryLocation has "inbox"
| extend isThreat = iff(isnotempty(ThreatTypes),1,0),
         isOrgLevelPolicy = iff(isnotempty(OrgLevelPolicy),1,0)
| project Timestamp, AdditionalFields, OrgLevelPolicy, SenderFromDomain,
ThreatTypes,SenderFromAddress, RecipientDomain, RecipientEmailAddress, isThreat,
isOrgLevelPolicy
| summarize CountEmailsBySenderDomain = count(),
            CountRecipients = dcount(RecipientEmailAddress),
            CountRecipientDomains = dcount(RecipientDomain),
            CountPoliciesByDomain = dcountif(OrgLevelPolicy, isOrgLevelPolicy==true),
            CountEmailsWithThreats = countif(isThreat==true)
            by SenderFromDomain
            | where CountEmailsWithThreats > 0
            | extend PercentHits =
strcat(tostring(toint((todouble(CountEmailsWithThreats) /
todouble(CountEmailsBySenderDomain)) * 100)),"%")
```

CountEmailsBySenderDomain	CountRecipients	CountRecipientDomains	CountPoliciesByDomain	CountEmailsWithThreats	PercentHits
145	84	5	0	2	1%
24	12	2	0	3	12%
33	28	1	0	33	100%
2	2	1	0	2	100%
4	4	1	0	1	25%

KQL – Delivery Location

```
EmailEvents
| where DeliveryLocation == "Inbox/folder"
| project NetworkMessageId, RecipientEmailAddress, SenderFromDomain, SenderDisplayName, Subject, ThreatTypes,
          DeliveryAction, DeliveryDate = Timestamp
| join kind=inner (
    EmailPostDeliveryEvents
    | where ActionType in~ ("Phish ZAP","Spam ZAP","Manual Remediation","Automated Remediation","Malware ZAP")
    | project NetworkMessageId, PostDeliveryActionTime = Timestamp, ActionTrigger, ActionType, DetectionMethods
) on NetworkMessageId
| extend TimeToQuarantine = datetime_diff('minute', PostDeliveryActionTime, DeliveryDate)
| join kind=leftouter ( UrlClickEvents
    | extend Clicker = tolower(coalesce(column_ifexists("AccountUpn",""),
             column_ifexists("RecipientEmailAddress",""),column_ifexists("UserId","")))
    | project NetworkMessageId, Clicker, ClickTime = Timestamp
) on NetworkMessageId
| extend TimeToClick = iff(isnull(ClickTime), tolong(""), datetime_diff('minute', ClickTime, DeliveryDate))
| summarize DeliveryDate = any(DeliveryDate),PostDeliveryActionTime = any(PostDeliveryActionTime),DetectionMethods = any(DetectionMethods),
  ActionTrigger = any(ActionTrigger),TimeToQuarantine = any(TimeToQuarantine),FirstClickDelay = minif(TimeToClick, isnotnull(TimeToClick)),
  Clicks = countif(isnotnull(Clicker)),UniqueClickers = dcountif(Clicker, isnotnull(Clicker)),
    Recipients = make_set(RecipientEmailAddress)
  by NetworkMessageId, SenderFromDomain
| extend AffectedUsers = array_length(Recipients) | summarize
    TotalEmails = dcount(NetworkMessageId),AffectedUsers = sum(AffectedUsers),TotalClicks = sum(Clicks),
    UniqueClickers = sum(UniqueClickers),AvgTimeToFirstClickMinutes = avg(todouble(FirstClickDelay)),
    AvgTimeToQuarantine = avg(todouble(TimeToQuarantine))
  by SenderFromDomain, DetectionMethods, ActionTrigger
| extend
    ClickRatePerEmail = todouble(TotalClicks) / iff(TotalEmails == 0, 1, TotalEmails),
    ClickerCoverage = todouble(UniqueClickers) / iff(AffectedUsers == 0, 1, AffectedUsers)
| order by TotalEmails desc
```

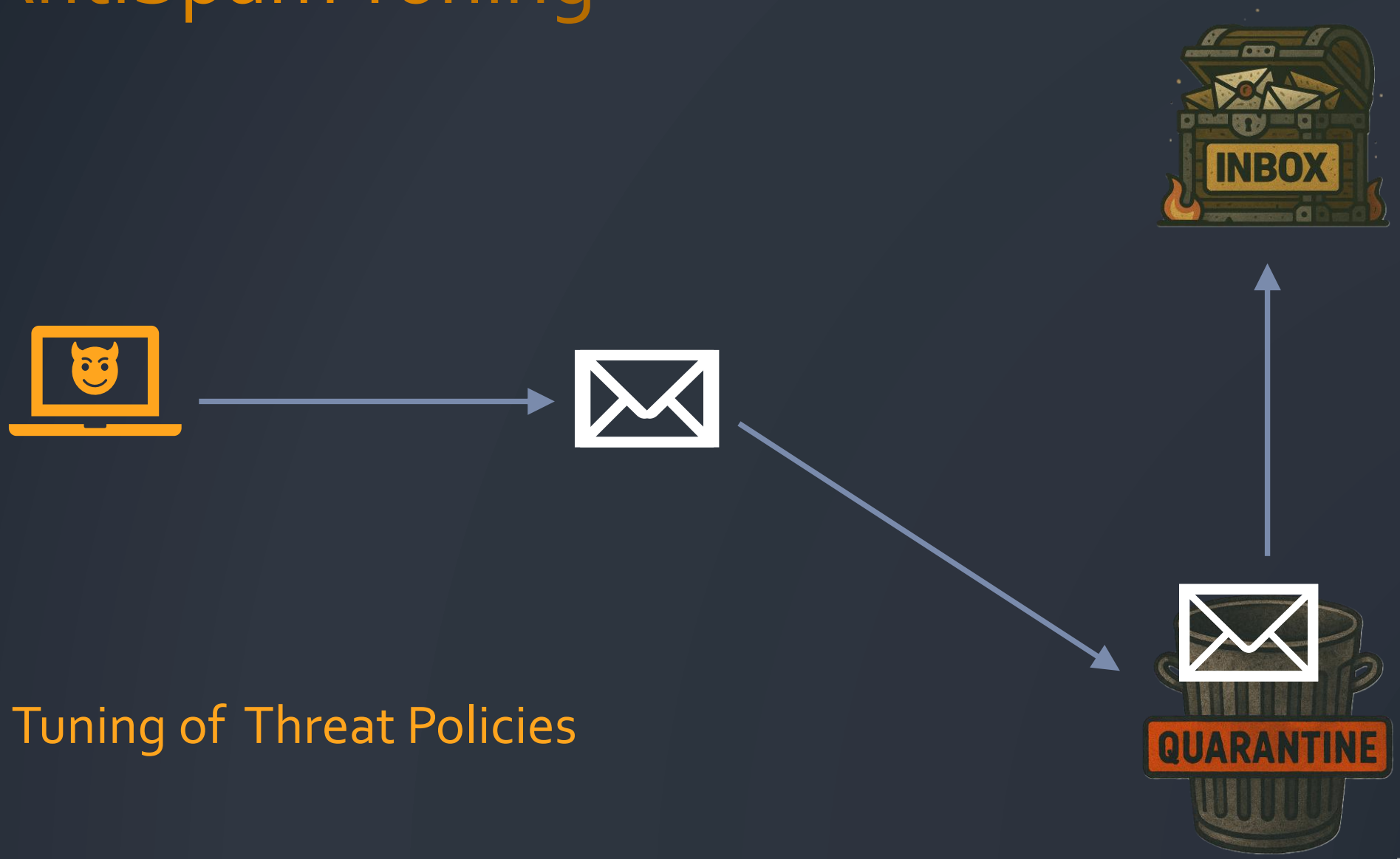
KQL – Delivery Location

```
EmailEvents
| where DeliveryLocation == "Inbox/folder"
| project NetworkMessageId, RecipientEmailAddress, SenderFromDomain, SenderDisplayName, Subject, ThreatTypes,
          DeliveryAction, DeliveryDate = Timestamp
```

SenderFromDomain	DetectionMethods	ActionTrigger	TotalEmails	AffectedUsers	TotalClicks	UniqueClickers	AvgTimeToQuarantine
maribelajar.co.id	{"Spam":["Fingerprint m...	SpecialAction	33	33	33	33	20.09090909090909
hello.klarna.com	{"Spam":["Fingerprint m...	SpecialAction	4	4	4	4	0.5
zoom.us		AdminAction	3	3	3	3	77.33333333333333
corfinancialcorp.c...	{"Phish":["URL detonatio...	SpecialAction	3	3	6	3	831.0
corfinancialcorp.c...	{"Phish":["URL malicious ...	SpecialAction	2	2	3	2	453.5
wasteconnections....	{"Phish":["Fingerprint ma...	SpecialAction	2	2	2	2	40.5
zulo.ccsend.com	{"Spam":["Fingerprint m...	SpecialAction	2	2	2	2	47.5

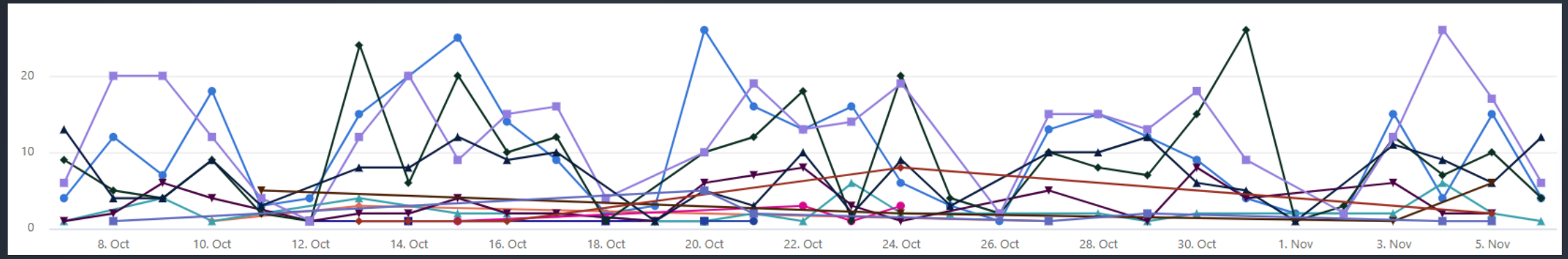
```
  Clicks = countif(isnotnull(Clicker)),UniqueClickers = dcountif(Clicker, isnotnull(Clicker)),
    Recipients = make_set(RecipientEmailAddress)
  by NetworkMessageId, SenderFromDomain
| extend AffectedUsers = array_length(Recipients) | summarize
    TotalEmails = dcount(NetworkMessageId),AffectedUsers = sum(AffectedUsers),TotalClicks = sum(Clicks),
    UniqueClickers = sum(UniqueClickers),AvgTimeToFirstClickMinutes = avg(todouble(FirstClickDelay)),
    AvgTimeToQuarantine = avg(todouble(TimeToQuarantine))
  by SenderFromDomain, DetectionMethods, ActionTrigger
| extend
    ClickRatePerEmail = todouble(TotalClicks) / iff(TotalEmails == 0, 1, TotalEmails),
    ClickerCoverage = todouble(UniqueClickers) / iff(AffectedUsers == 0, 1, AffectedUsers)
| order by TotalEmails desc
```

AntiSpam Tuning



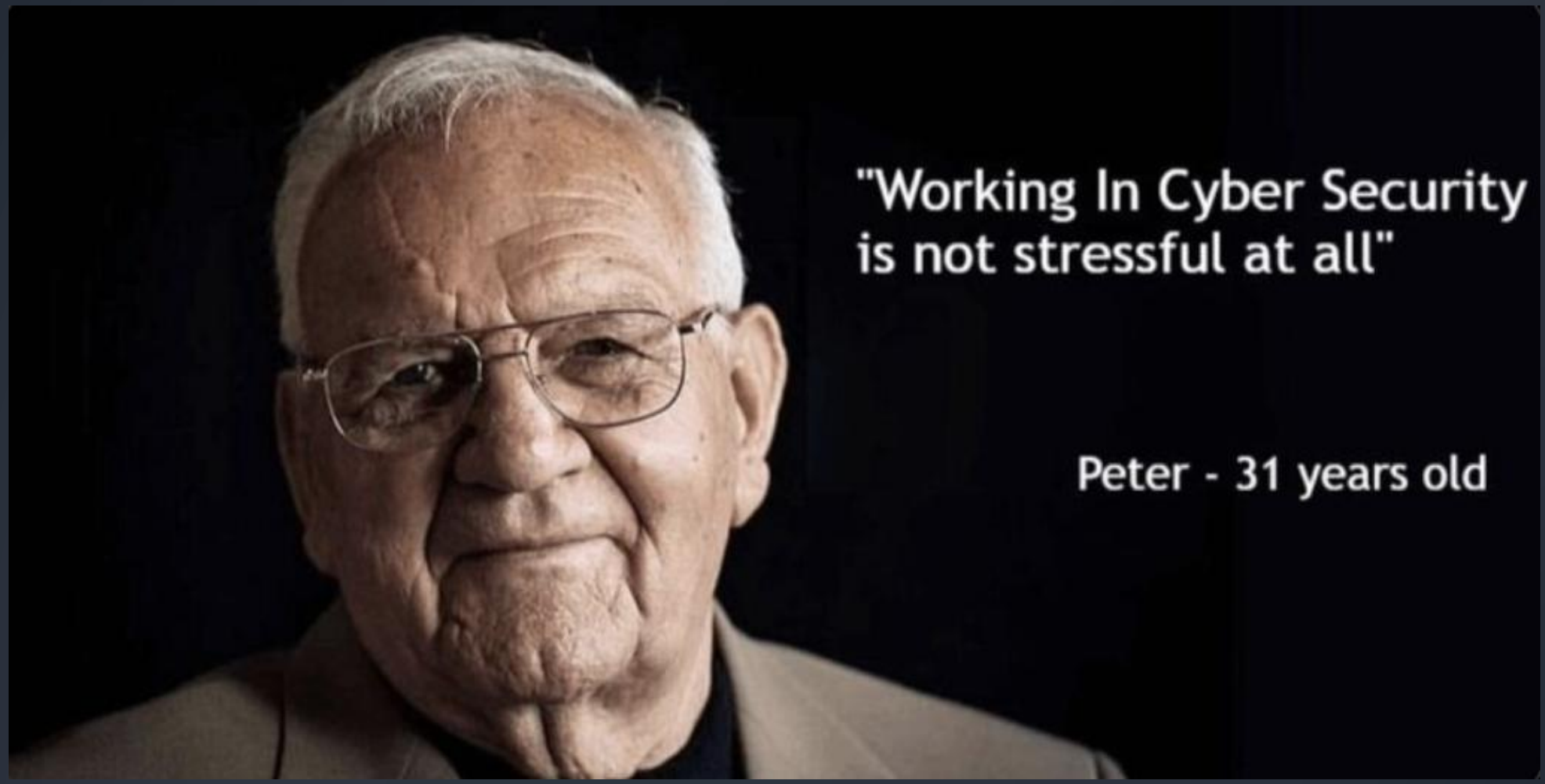
Tuning of Threat Policies

KQL – AntiSpam Tuning




```
EmailPostDeliveryEvents
| where Action has "Quarantine release"
| summarize count() by bin(Timestamp,1d)
| render timechart
```

KQL for the Security Analyst



KustoCon
Learn | Share | Practice

Emails with alerts and clicks

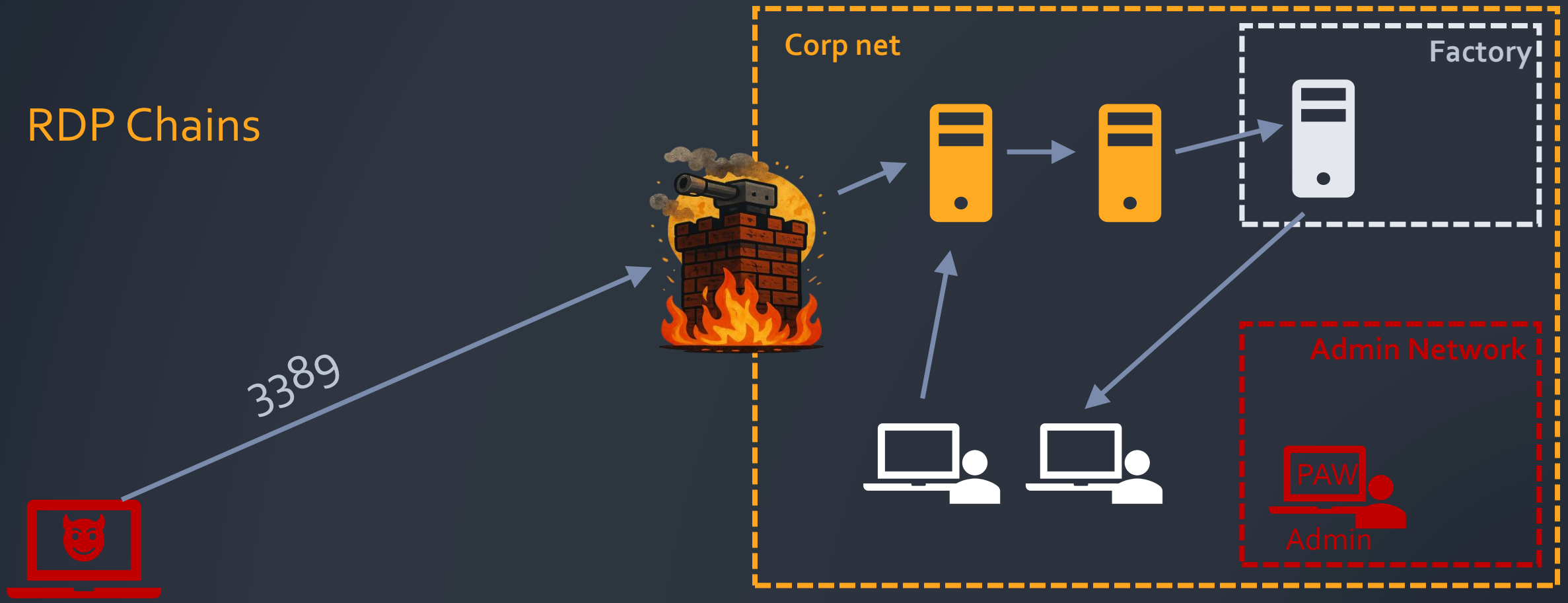
Email detections with clicks

```
// Emails with malicious URL removed after delivery >> URLClick on those messages
AlertInfo
| where Title contains "Email messages containing malicious URL removed after delivery"
| project-rename AlertTimetamp = Timestamp
| join (AlertEvidence | summarize arg_max(Timestamp,*) by NetworkMessageId
| project-rename EvidenceTimetamp = Timestamp
) on AlertId
| extend _json = parse_json(AdditionalFields)
| extend SenderFromAddress = tostring(_json.Sender),
         RecipientAddress = tostring(_json.Recipient)
         | project-away _json
         | extend ResearchSubject = replace_regex(EmailSubject,@"[a-f0-9]{10,}","")
| project AlertTimetamp, EvidenceTimetamp, AccountUpn, EmailSubject, SenderFromAddress,RecipientAddress,
          NetworkMessageId, AlertId,ResearchSubject
| join (
        UrlClickEvents
        | project URLClickTimestamp=Timestamp,AccountUpn, ActionType, NetworkMessageId, IPAddress, Url, UrlChain
        ) on NetworkMessageId
        | project-away *1
```

KustoCon
Learn | Share | Practice

Scenario

RDP Chains



KQL – RDP Chains

Target_IPAddress	flow	CountJumps
10.243.0.21	["10.10.1.95","10.11.14.148","10.241.2.4","10.243.0.21"]	4
10.243.0.21	["10.11.14.148","10.241.2.4","10.243.0.21"]	3
10.10.1.19	["10.10.1.95","10.11.14.148","10.10.1.19"]	3

```
let NetworkNodes =
DeviceNetworkInfo
| mv-apply todynamic(IPAddresses) on (
                            summarize IPIDs = make_bag(
                                                                pack("AddressType",
                                                                       tostring(IPAddresses.AddressType),
                                                                       "IPAddress",
                                                                       IPAddresses.IPAddress,
                                                                       "SubnetPrefix",
                                                                       IPAddresses.SubnetPrefix
                                                                   )
                                                            )
                        )
| extend IPAddress = tostring(IPIDs.IPAddress),AddressType = tostring(IPIDs.AddressType),SubnetPrefix = tostring(IPIDs.SubnetPrefix)
| where AddressType != "LinkLocal" | project-away IPIDs
| join (DeviceInfo) on DeviceId | project Timestamp, DeviceId, DeviceName, MacAddress, IPAddress,AddressType,SubnetPrefix, OSPlatform, PublicIP;
let NetworkEdges =
DeviceNetworkEvents
| where LocalPort == 3389
| where ActionType == "InboundConnectionAccepted"
| where LocalIP !contains ":"
| summarize arg_max(Timestamp,*) by RemoteIP, LocalIP
| project Timestamp, SourceIP=RemoteIP, DestinationIP=LocalIP;
NetworkEdges
| make-graph SourceIP --> DestinationIP with NetworkNodes on IPAddress
| graph-match (Source)<-[reports*2..5]-(Target)
project Target.IPAddress, flow = map(reports, SourceIP)
| extend CountJumps = array_length(flow)
| sort by CountJumps desc
| take 3
```

KustoCon
Learn | Share | Practice

How deep is the rabbit hole?

detect_anomalous_access_cf_fl()

Query language > Functions

detect_anomalous_new_entity_fl()

Query language > Functions

detect_anomalous_spike_fl()

Query language > Functions

graph_blast_radius_fl()

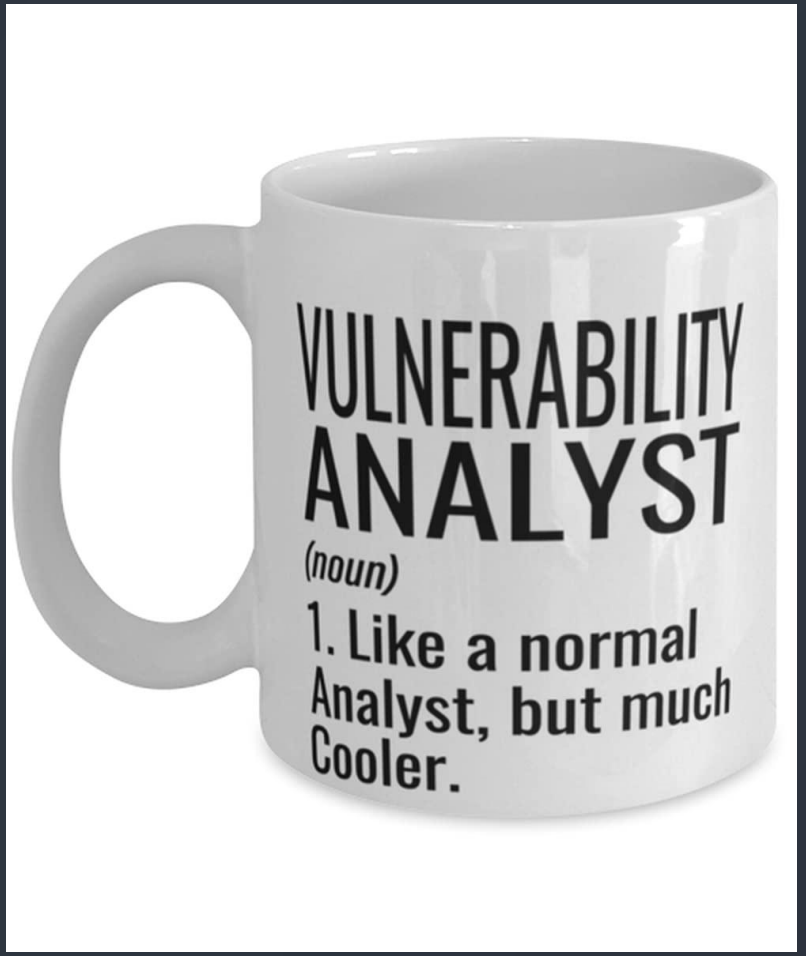
Query language > Functions

graph_exposure_perimeter_fl()

Query language > Functions

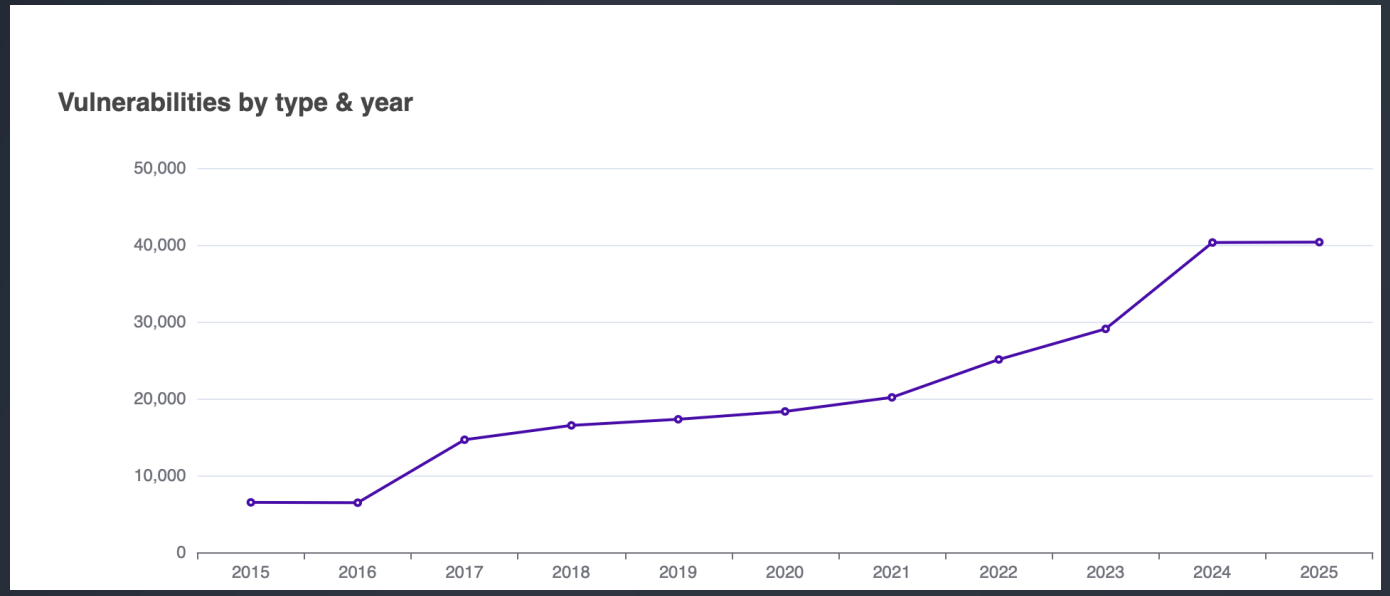
https://learn.microsoft.com/en-us/kusto/query/functions/user-defined-functions

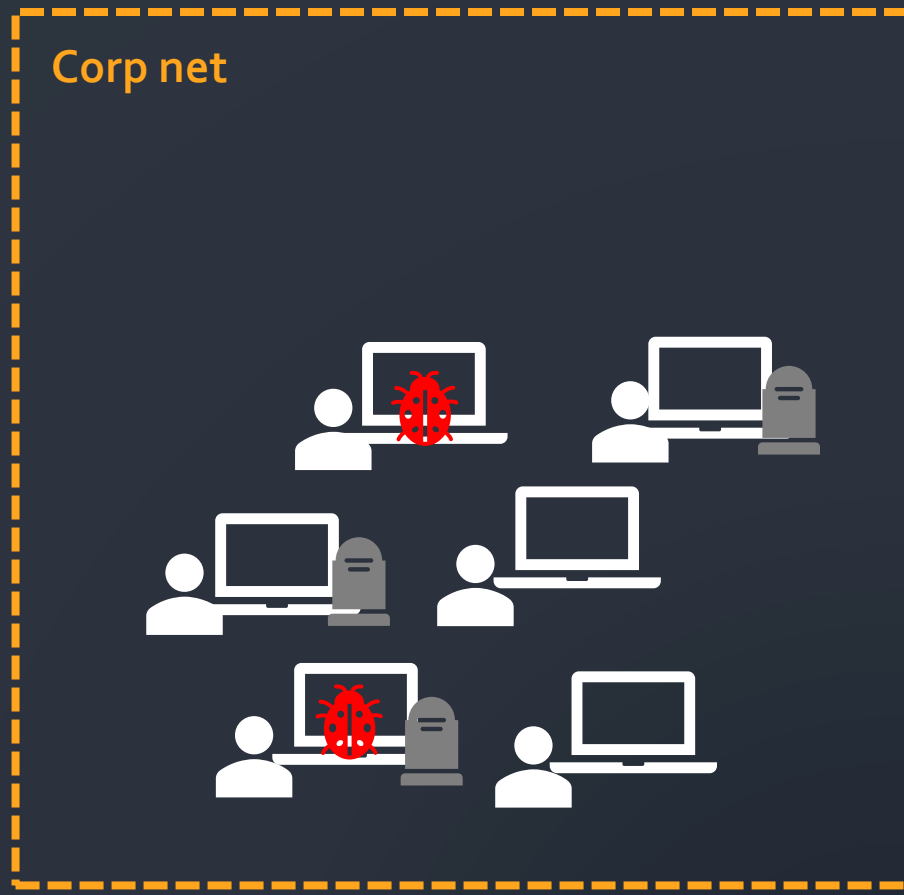
KQL for the Vulnerability Analyst



KustoCon
Learn | Share | Practice

EOL/EOS or Unpatched OS / Applications





KQL – Vulnerability Data

```
let all = materialize ( DeviceTvmSoftwareInventory
| extend VendorSoftwareNVersion = strcat(SoftwareVendor,"-",SoftwareName,"-",SoftwareVersion)
| join (
DeviceTvmSoftwareVulnerabilitiesKB
        | where IsExploitAvailable == 1 //FILTER
            | join DeviceTvmSoftwareVulnerabilities on CveId
            | project-away *1
        | extend VendorSoftwareNVersion = strcat(SoftwareVendor,"-",SoftwareName,"-",SoftwareVersion)
            | join kind=leftouter (
                                        DeviceTvmSoftwareEvidenceBeta
                                        | extend VendorSoftwareNVersion = strcat(SoftwareVendor,"-",SoftwareName,"-",SoftwareVersion)
             ) on DeviceId, VendorSoftwareNVersion
             | project-away *1  ) on VendorSoftwareNVersion | project-away *1);
             // | where isnotempty( EndOfSupportDate));
let TopVulnVendors =
all
| summarize TopVulnVendors=count() by SoftwareVendor
| order by TopVulnVendors desc, SoftwareVendor asc
| take 10
| serialize
| extend Row=row_number()
| project Row, SoftwareVendor, TopVulnVendors
let AppsEOS =
all
| extend HowOld=datetime_diff('day', now(), EndOfSupportDate)
| extend DaysEOS=case(HowOld>=365,"1 year", HowOld>=180,"6 months", HowOld>=90,"3 months", HowOld>=30,"1 month", HowOld>=1,"1-30 days","")
| where isnotempty(DaysEOS)
| summarize AppsEOS=count() by DaysEOS
| order by AppsEOS desc, DaysEOS asc
| serialize
| extend Row=row_number()
| project Row, DaysEOS, AppsEOS;
let DevicesWeos =
all
| where isnotempty(EndOfSupportDate)
| summarize NumberOfDevicesWithEOSsoftware=dcount(DeviceName)
| extend Row=1
| project Row, NumberOfDevicesWithEOSsoftware;
TopVulnVendors
| join kind=fullouter (AppsEOS) on Row
| join kind=fullouter (DevicesWeos) on Row
```

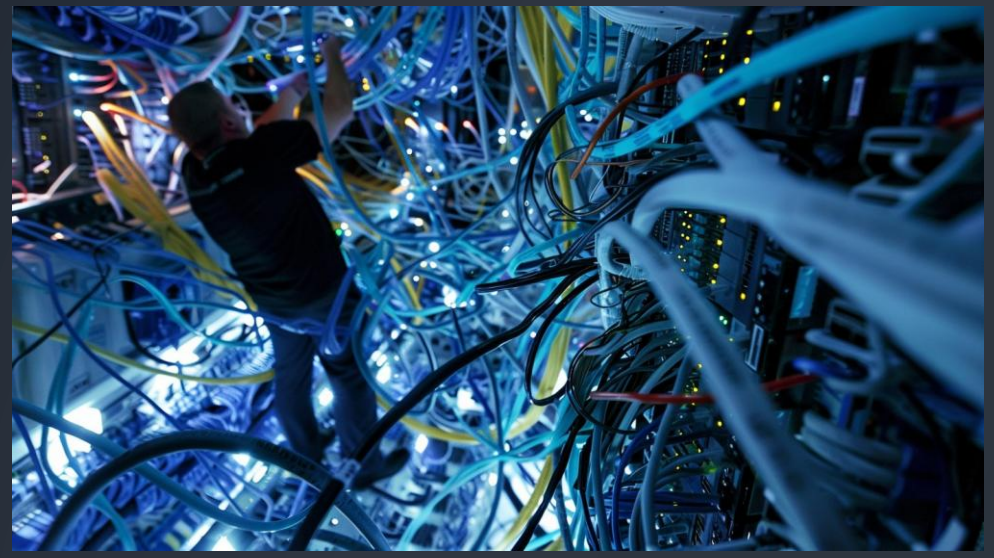
Filter on Exploitable, EOS and more

```
DeviceTvmSoftwareVulnerabilities
| where SoftwareName contains "windows_"
| extend MissingUpdate = tostring(extract(@"^(\w+\s\d{4})",1,RecommendedSecurityUp
        UpdateYear = tostring(extract(@"(\d{4})",1,RecommendedSecurityUpdate))
| extend OSType = iff(SoftwareName has "server","WindowsServer","WindowsClient")
| summarize CveIds=make_set(CveId) by DeviceId, DeviceName, MissingUpdate, OSType,
| extend MissingUpdates = bag_pack(MissingUpdate,bag_pack("CveIds",CveIds))
| summarize MissingUpdates=make_set(MissingUpdates) by DeviceId, DeviceName, OSTyp
| join
    (
    DeviceTvmSoftwareVulnerabilities
    | where SoftwareName contains "windows_"
    | extend MissingUpdate = tostring(extract(@"^(\w+\s\d{4})",1,RecommendedSecuri
          UpdateYear = tostring(extract(@"(\d{4})",1,RecommendedSecurityUpdate))
    | extend MonthNumber = case
                                (
                                MissingUpdate has "January","01",
                                MissingUpdate has "February","02",
                                MissingUpdate has "March","03",
                                MissingUpdate has "April","04",
                                MissingUpdate has "May","05",
                                MissingUpdate has "June","06",
                                MissingUpdate has "July","07",
                                MissingUpdate has "August","08",
                                MissingUpdate has "September","09",
                                MissingUpdate has "October","10",
```

Filters: Add filter

	DeviceName	OSType	OSArchitecture	OSPlatform	MissingUpdates	LastAppliedUpdate	LastUpdate	LastDeviceInfoUpdate
aa2a3e9207...	w11.cobrakai.local	WindowsClient	x64	Windows11	[{"October 2025":{"CveIds":["CVE-...	March 2023 Security Up...	202303	Oct 24, 2025 3:04:34 AM
07bb29ef8d...	app1.cobrakai.local	WindowsServer	x64	WindowsServer2022	[{"October 2025":{"CveIds":["CVE-...	October 2020 Security U...	202010	Nov 5, 2025 2:50:55 AM
36a34c562a...	win10.cobrakai.local	WindowsClient	x64	Windows10	[{"October 2025":{"CveIds":["CVE-...	March 2023 Security Up...	202303	Oct 24, 2025 3:35:33 AM
4fcba46e21...	win-iuto87020tq	WindowsServer	x64	WindowsServer2022	[{"October 2025":{"CveIds":["CVE-...	October 2020 Security U...	202010	Nov 4, 2025 11:32:13 AM
92ddde5a17...	dc1.cobrakai.local	WindowsServer	x64	WindowsServer2022	[{"October 2025":{"CveIds":["CVE-...	September 2025 Securit...	202509	Nov 5, 2025 3:01:56 AM

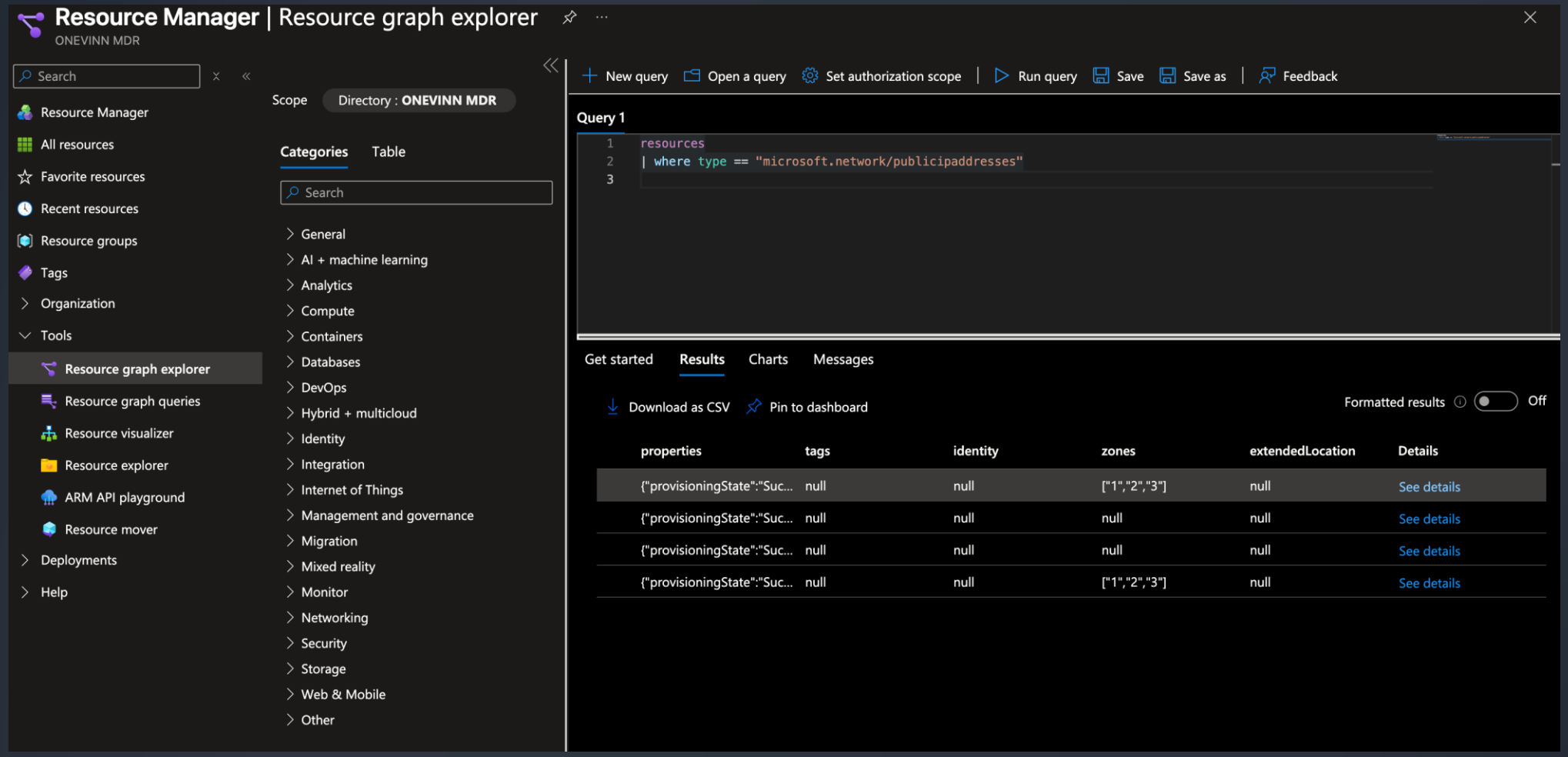
Public IPs



Where are all my public IPs?

- I need to scan them with my vulnscanner

Azure Resource Graph Explorer



KustoCon
Learn | Share | Practice

KQL – Azure Resource Graph Public IPs

```
resources
| where type == "microsoft.network/publicipaddresses"
| project name, parse_json(properties).ipAddress
```

Home > Resource Manager | Resource graph queries >

Azure Resource Graph Explorer

New query | Open a query | Set authorization scope | Run query | Save | Save as | Feedba

Categories | Table

Search

- General
- AI + machine learning
- Analytics
- Compute
- Containers
- Databases
- DevOps
- Hybrid + multicloud
- Identity
- Integration
- Internet of Things
- Management and governance
- Migration
- Mixed reality
- Monitor

Query 1

```
1
```

Get started | Results | Charts | Messages

Filter... | About Resource Graph | Language reference | Keyboard shortcuts

Most recent

Query 1

Functions

Tabular

Scalar

```
let DB_XOR = (s:string, k:string) {
    let key = toscalar(k);
    let key_bytes = unicode_codepoints_from_string(key);
    let xor_string = unicode_codepoints_from_string(s);
    let xored_bytes = toscalar(
    range i from 0 to array_length(xor_string) - 1 step 1
    | extend cipher_byte = tolong(xor_string[i]),
             key_byte = tolong(key_bytes[i % array_length(key_bytes)])
    | extend plain_byte = binary_xor(cipher_byte, key_byte)
    | summarize result = make_list(plain_byte)
);
    toscalar(unicode_codepoints_to_string(xored_bytes))
};
```

Stop query | Last 24 hours | Save | Share link | Create sur

Query

Query results are presented in your local time zone as per settings. Kusto filters, however, work in UTC.

```
let DB_XOR = (s:string, k:string) {
    let key = toscalar(k);
    let key_bytes = unicode_codepoints_from_string(key);
    let xor_string = unicode_codepoints_from_string(s);
    let xored_bytes = toscalar(
    range i from 0 to array_length(xor_string) - 1 step 1
    | extend cipher_byte = tolong(xor_string[i]),
             key_byte = tolong(key_bytes[i % array_length(key_bytes)])
    | extend plain_byte = binary_xor(cipher_byte, key_byte)
    | summarize result = make_list(plain_byte)
);
    toscalar(unicode_codepoints_to_string(xored_bytes))
};
print ClearText = DB_XOR("SI051!/0!SYN 2&}7 &","KUSTO")
```

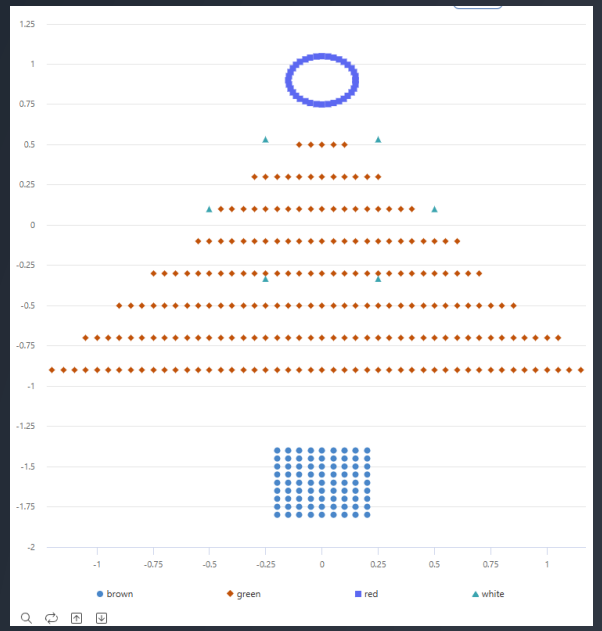
Getting started | **Results** | Query history

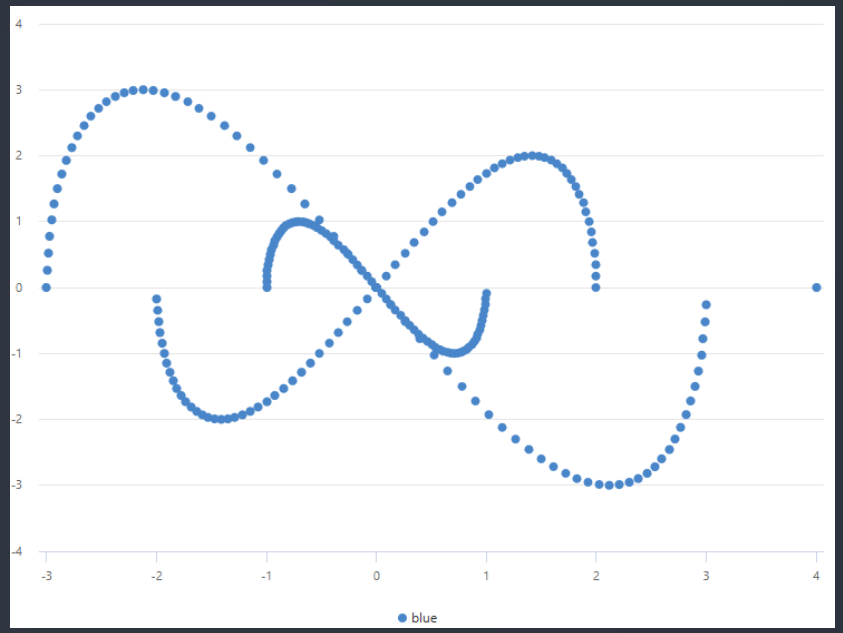
Export | Show empty columns | Loading... | Search | 00:01.151 | Low

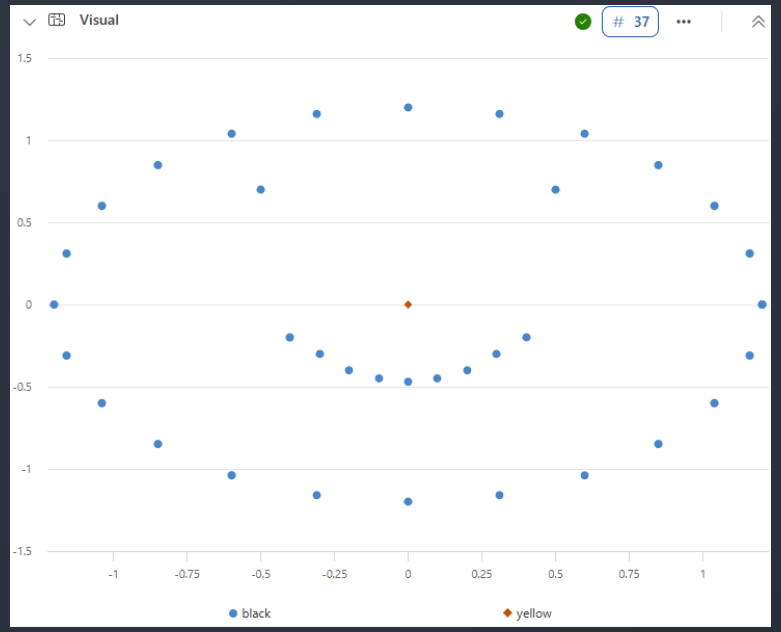
Filters: Add filter

```
AudienceCheck
| where SessionStatus == "LastSlide"
| where Audience == "ClapsAndCheers"
| project SessionStatus, Audience, GoPractice
| extend When = now()
```

Screens from the last demos







KustoCon
Learn | Share | Practice

Thanks to our Sponsors

baseVISION

glueck kanja

INVICTUS INCIDENT RESPONSE

water IT Security & Defense

Microsoft

scopewyse we are what's next

KustoWorks

SecMinds SOLUTIONS

KustoCon Learn | Share | Practice