KQL|Cafe

Session 2 | Hello again

Your | hosts

Your | hosts

Your | hosts

Alex Verboon

https://twitter.com/alexverboon

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

https://github.com/alexverboon

https://www.verboon.info/

Your | hosts

Alex Verboon

https://twitter.com/alexverboon

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

https://github.com/alexverboon

https://www.verboon.info/

Gianni Castaldi

https://twitter.com/castello_johnny

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

https://github.com/KustoKing

https://www.kustoking.com/

Today's | Guest
Matt Zorich

Principal Cyber Security Specialist

My blog: https://learnsentinel.blog/blog/
#365daysofKQL: https://twitter.com/search?q=%23365daysofkql
KQL Queries: https://github.com/reprise99/Sentinel-Queries
Awesome KQL Resources: https://github.com/reprise99/awesome-kql-sentinel
Twitter: https://twitter.com/reprise_99

Today’s | Agenda

Hello again

KQL Tables | What's new in KQL

Working with IOCs

KQL Tools

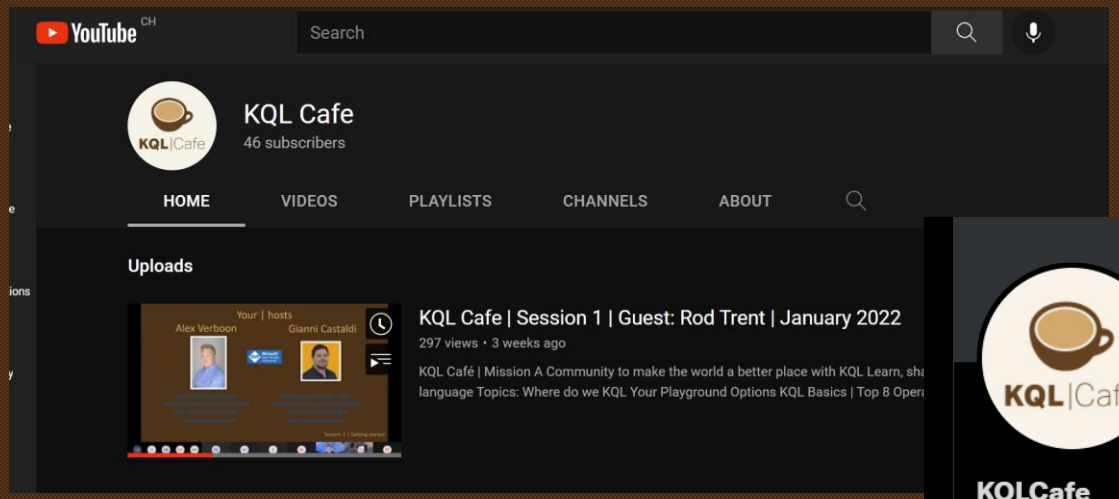
Todays guest speaker: Matt Zorich

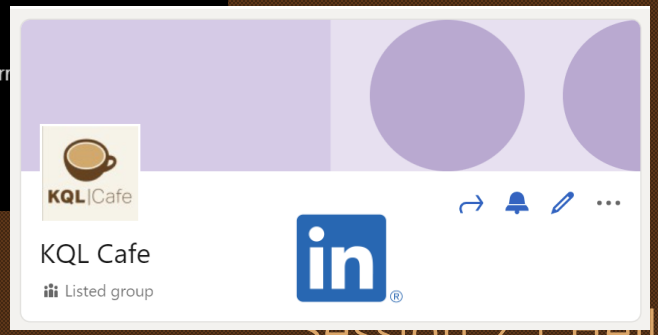
What did you do with KQL this month?

KQL Challenge of the month

Hello Again

Information page: https://www.kqlcafe.com/
LinkedIn group: https://www.linkedin.com/groups/14053778/
Youtube: https://www.youtube.com/channel/UCUJwJO79TYZdnpQ9WWtzQDg
Meetup: https://www.meetup.com/kql-cafe/
Discord Channel: https://discord.gg/V4JWfycSkU





KQL Tables | What’s new

Microsoft 365 Defender

- DeviceTvmSoftwareEvidenceBeta
- AADSignInEventsBeta
- DeviceNetworkEvents – ActionType – NetworkSignatureInspected

KQL Tables | What’s new

DeviceTvmSoftwareEvidenceBeta

The DeviceTvmSoftwareEvidenceBeta table in the advanced hunting schema contains data from Threat & Vulnerability Management related to the software evidence section. This table allows you to view evidence of where a specific software was detected on a device. You can use this table, for example, to identify the file paths of specific software. Use this reference to construct queries that return information from the table.

https://docs.microsoft.com/en-us/microsoft-365/security/defender/advanced-hunting-devicetvmsoftwareevidencebeta-table?view=o365-worldwide

Guidance for preventing, detecting, and hunting for exploitation of the Log4j 2 vulnerability - Microsoft Security Blog

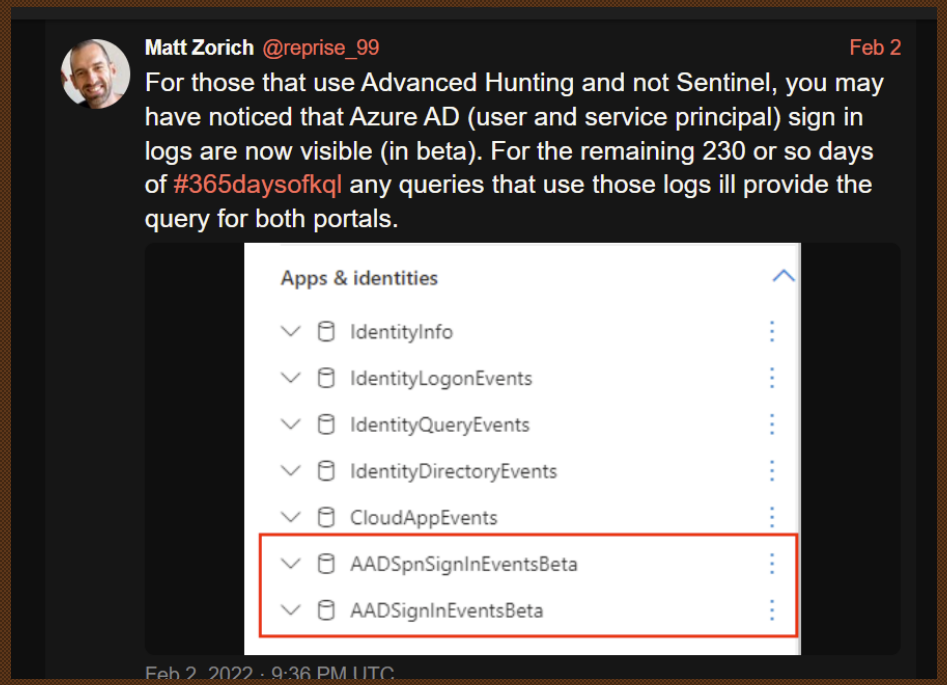
```
DeviceTvmSoftwareEvidenceBeta
| mv-expand DiskPaths, RegistryPaths
| project DeviceId, SoftwareName, SoftwareVendor, SoftwareVersion, DiskPaths, RegistryPaths, LastSeenTime
```

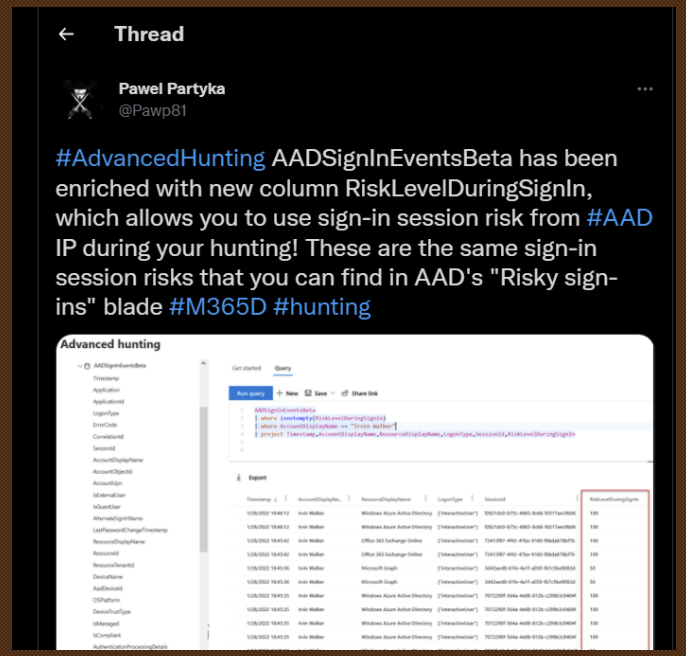
KQL Tables | What’s new

AADSignInEventsBeta

The AADSignInEventsBeta table in the advanced hunting schema contains information about Azure Active Directory interactive and non-interactive sign-ins

AADSignInEventsBeta table in the advanced hunting schema | Microsoft Docs
Hunt for Azure Active Directory sign-in events - Dr. Ware Technology Services - Microsoft Silver Partner (drware.com)





KQL Tables | What’s new

DeviceNetworkEvents – ActionType – NetworkSignatureInspected

Query

```
DeviceNetworkEvents
| where ActionType == 'NetworkSignatureInspected'
| extend signaturename = tostring(parse_json(AdditionalFields).SignatureName)
| distinct signaturename
```

signaturename ↑
DNS_Request
FTP_Client
HTTP_Client
HTTP_RequestBodyPara...
HTTP_Server
HTTPS_Client
NTLM-Challenge
SMB_Client

KQL Tables | What's new

DeviceNetworkEvents – ActionType – NetworkSignatureInspected

```
// Do we have DNS Traffic
DeviceNetworkEvents
| where RemotePort == 53
| where ActionType in ("ConnectionSuccess","ConnectionFound")
```

```
// Which servers receive DNS Traffic
DeviceNetworkEvents
| where RemotePort == 53
| where ActionType in ("ConnectionSuccess","ConnectionFound")
| summarize Total = count(), Devices = dcount(DeviceId)  by RemoteIP
```

KQL Tables | What's new

DeviceNetworkEvents – ActionType – NetworkSignatureInspected

```
// Introducing Network Signatures
DeviceNetworkEvents
| where ActionType == "NetworkSignatureInspected"
| extend AF = parse_json(AdditionalFields)
| extend SignatureName = AF.SignatureName
```

```
// hunting for DNS servers
DeviceNetworkEvents
| where ActionType == "NetworkSignatureInspected"
| extend AF = parse_json(AdditionalFields)
| extend SignatureName = AF.SignatureName
| where SignatureName == "DNS_Request"
| summarize Total = count(), Servers = dcount(DeviceId) by RemoteIP
```

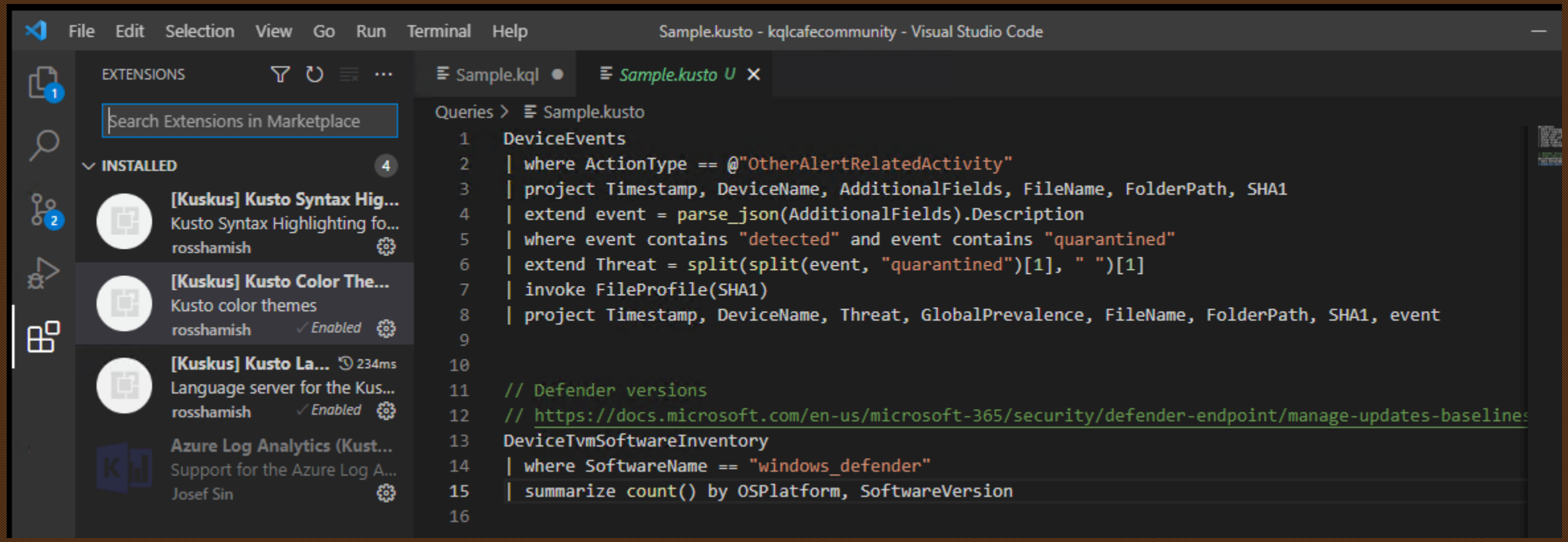
```
// hunting for DNS on different ports
let DNSPorts = dynamic([53]);
DeviceNetworkEvents
| where ActionType == "NetworkSignatureInspected"
| extend AF = parse_json(AdditionalFields)
| extend SignatureName = AF.SignatureName
| where SignatureName == "DNS_Request"
| where RemotePort !in(DNSPorts)
```

Working with IOCs

External data is the way to go. *[Session 3 will be indepth!]*

```
let OSINT =  externaldata(IP:string) [@"https://raw.githubusercontent.com/stamparm/ipsum/master/levels/6.txt"]
with (format="txt",ignoreFirstRecord=false)
| where IP !startswith "#"
| project IP;
CommonSecurityLog
| where DestinationIP in(OSINT)
```

KQL Tools | Visual Studio Code Extensions



What did you do with KQL this month?

Microsoft Defender for Identity and Npcap - Microsoft Tech Community

https://techcommunity.microsoft.com/t5/microsoft-defender-for-identity/microsoft-defender-for-identity-and-npcap/m-p/2584151

```
DeviceNetworkEvents
| where LocalPort == "88"
| distinct DeviceId
| join kind=inner (
    DeviceInfo
    | where OSPlatform hasprefix "windowsserver"
    | summarize  arg_max(Timestamp,*) by DeviceId
) on DeviceId
| project Timestamp, DeviceId, OSPlatform, OSVersionInfo
| join kind=leftouter (
    DeviceProcessEvents
    | where FileName =~ "Microsoft.Tri.Sensor.exe"
    | summarize arg_max(Timestamp,*) by DeviceId
    | distinct DeviceId, ProcessVersionInfoProductName, ProcessVersionInfoProductVersion
) on DeviceId
| project-away DeviceId1
| join kind=inner (
    DeviceTvmSoftwareInventory
    | where SoftwareName contains "pcap"
    | distinct DeviceId, SoftwareVendor, SoftwareName, SoftwareVersion
) on DeviceId
| project-away DeviceId1
```

Use the above query to identify MDI Agents where winpcap or npcap is installed

Today's | Guest
Matt Zorich

Principal Cyber Security Specialist

My blog: https://learnsentinel.blog/blog/
#365daysofKQL: https://twitter.com/search?q=%23365daysofkql
KQL Queries: https://github.com/reprise99/Sentinel-Queries
Awesome KQL Resources: https://github.com/reprise99/awesome-kql-sentinel
Twitter: https://twitter.com/reprise_99

Challenge of the month

The winner of lasts months challenge is

@shviammalaviya

With his submission:

https://github.com/KQLCafe/kqlcafecommunity/issues/1

Our next challenge will be one without submissions, but one where you can test your knowledge. The questions will be about data in Microsoft Defender or Endpoint.

https://github.com/KQLCafe/kqlcafecommunity/blob/main/Challenge%20of%20the%20Month/February%202022/Challenge.txt

On the 8th of March we will the answer file in the same repository