Unleashing the Forensics Skillset:

Techniques for Extraction and Analysis of the Evidence

Paula Januszkiewicz

Azure Days, Warsaw, 12th of March 2026

CQURE

Unleashing the Forensics Skillset:

Techniques for Extraction and Analysis of the Evidence

Paula Januszkiewicz

CQURE: CEO, Cybersecurity Expert, Penetration Tester

CQURE Academy: Trainer

Microsoft MVP on Cloud and Datacenter Management

Microsoft Regional Director

paula@cqure.us

@PaulaCqure @CQUREAcademy

www.cqureacademy.com

CQURE

AUSTRALIAN CYBER CONFERENCE 2023
MELBOURNE | 17 – 19 OCTOBER

AISA
VICTORIA State Government
Australian Government

FEATURED SPEAKERS

Professor Brian Cox CBE FRS

Arguably the UK's best known experimental physicist, Professor Brian Cox's books and TV programmes have been read and watched around the world and credited with making science engaging and accessible...

READ MORE

Chris Hadfield

ASTRONAUT, ENGINEER AND MILITARY FIGHTER/TEST PILOT

Referred to as "the most famous astronaut since Neil Armstrong," Colonel Chris Hadfield is a worldwide sensation whose video of David Bowie's 'Space Oddity' - seen by over 75 million people - was call...

READ MORE

Mikko Hypponen

Mikko Hypponen is cybersecurity research
He has
Times,
and lect
Oxford,

READ

Paula Januszkiewicz

nic X edition

May 31st - June 2nd
Oslo Spektrum

June 1st | 10:00 CEST
Adventures in the Underland: Uncommon Hacker's Persistency Methods and Countermeasures

June 1st | 16:00 CEST
Hackers' Perspective on Remote Working: Know the Risks, Fight the Tricks

Microsoft CQURE ACADEMY©

We are proud to announce that **Paula Januszkiewicz** was rated as **No 1 Speaker** at Microsoft **Ignite!!!**

May 4-8, 2015
Chicago, IL

HACK IN PARIS CYBERSECURITY EVENT 2022

RSAConference
Where the world talks security

I have been awarded the RSA Top-Rated Session!

RSAC TOP-RATED SESSION

Strong Story to Tell: Top 10 Mistakes by Administrators About Remote Work

Paula Januszkiewicz
CEO, Cybersecurity Expert, CQURE Inc.

No.1 Speaker

Paula Januszkiewicz
CEO CQURE

She received a **"Best of Briefings"** award at her "CQTools: The New Ultimate Hacking Toolkit" Black Hat Asia 2019 briefing session

black hat

LEAP

RIYADH FRONT EXPO CENTRE, SAUDI ARABIA

Keynote: Cyber Attacks & Defense Transformation: The Look to Past, Present & Future.
by Paula Januszkiewicz, CEO of CQURE
February 3rd, 2022 | 16:25 AST (GMT +3)

Dr. Makaziwe...
Founder
House of Mandela...

Eugene Kaspersky
CEO & Co-Founder
Kaspersky

Luís Figo
Ballon d'Or winning...

What does CQURE do?

1. Consulting Services:
   a) Extensive IT Security Audits and Penetration Tests of all kinds,
   b) Configuration Audit and Architecture,
   c) Design Social Engineering Tests,
   d) Advanced Troubleshooting and Debugging,
   e) Emergency Response Services.

2. R&D & CQLabs Tools & Hacks Publications.

3. Trainings & Seminars:
   a) Offline (mainly via our partners worldwide),
   b) Online

Get to know us better!

Scan the QR code or visit
https://cqu.re/adp
to get access to this presentation
and find out more about CQURE!

The Impact of Cybercrime

There is always something to find...

Insightful Stats for 2026

$4.4M

is the global average cost of a data breach in 2025.

286 Days

that's how long, on average, it took to identify and contain breaches involving stolen or compromised credentials.

68%

of data breaches involved human error in 2025.

>50%

of breached organizations are facing high levels of security staffing shortages.

Source: IBM (2025), Verizon (2025), Statista (2026)

Impactful Hacking Stats for 2026

43%

of all cyberattacks are aimed at small businesses.

95%

of data breaches are financially motivated.

83%

of breaches involve external actors.

64%

of organizations do not have a cyber security incident response plan.

Source: Verizon (2023); Accenture (2023), Kroll (2022)

Data Extraction and Analysis Techniques

Demo: Extracting files from the memory dump

Reasons for low detection rates

#1 Understaffing

62% of organizations feel they don’t hire enough cybersecurity professionals.

60% of organizations struggle to keep their qualified cybersecurity staff.

Source: ISACA

#2 Alert fatigue

28% of alerts are never addressed.

Source: IDC

#3 No (or poor) IR plan

27% of all organizations don’t have a plan for IR against data breaches.

Among those that have one, only 63% test it regularly.

Source: Astra

#4 Cutting on cybersecurity

52% of organizations report sacrificing cybersecurity for speed-to-market.

Source: PagerDuty

Threat Hunting

=

Proactive

vs

Forensics

=

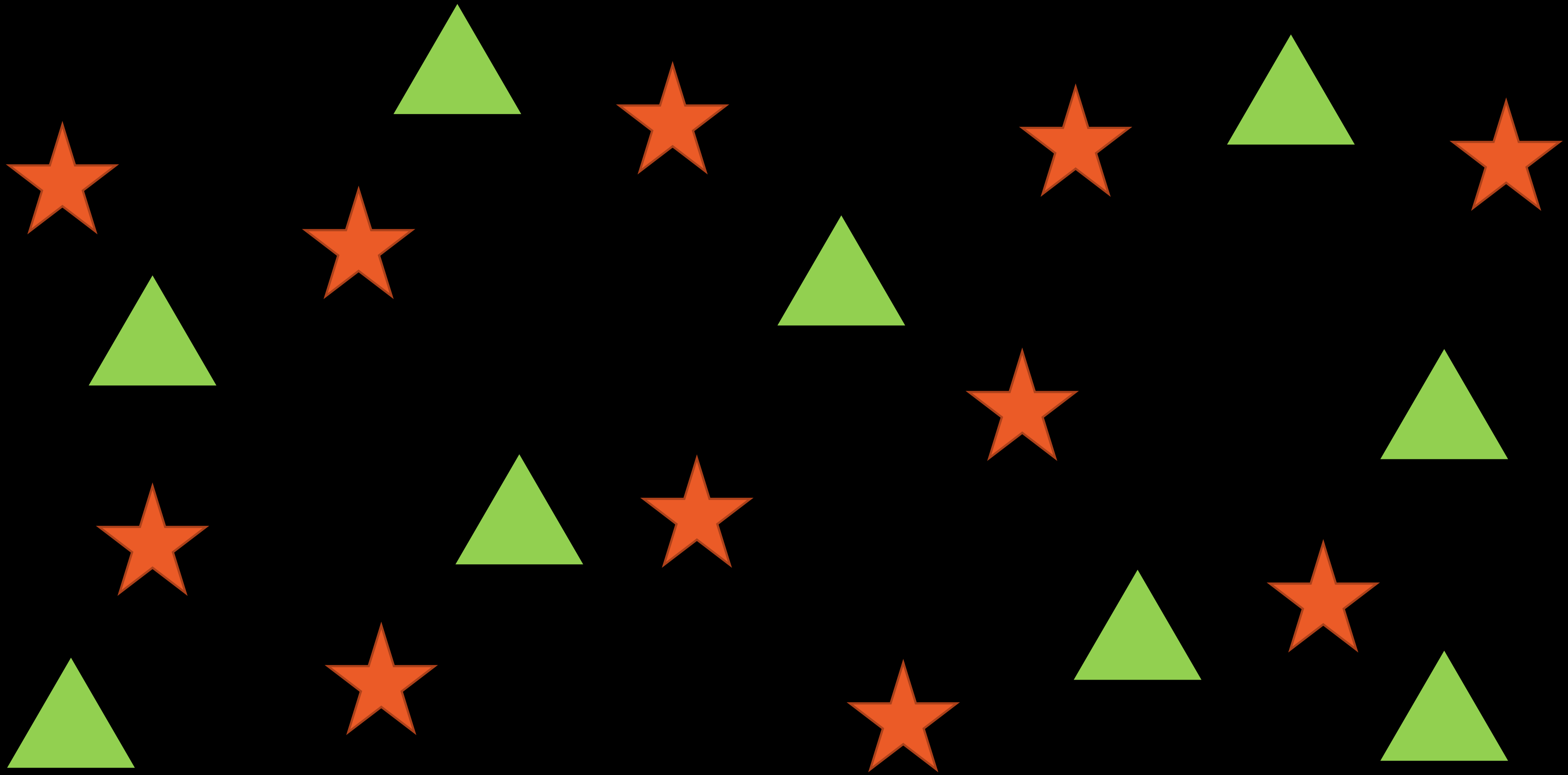
Reactive

Where do we take data from?

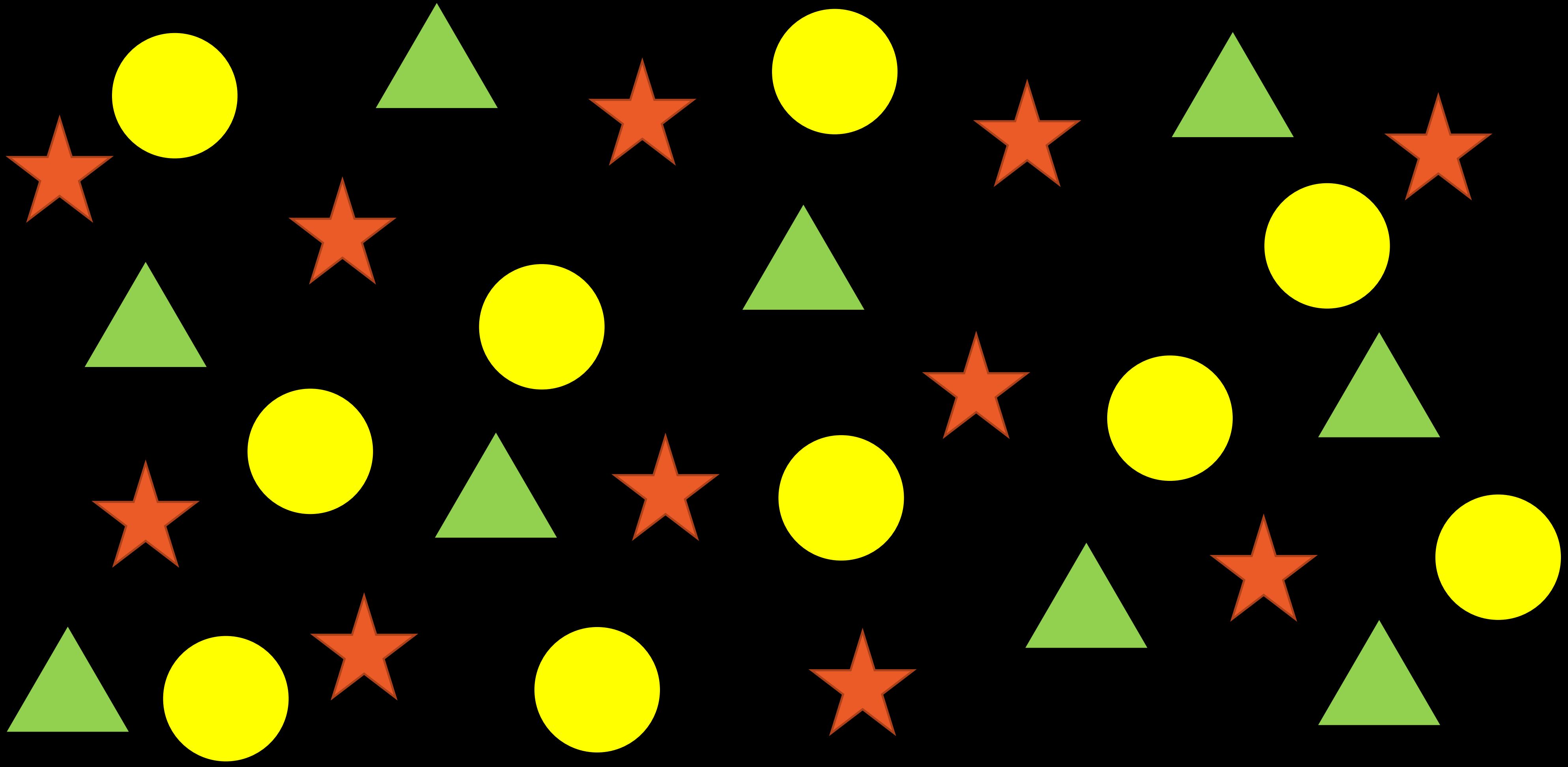
- Network logs
- System alerts
- External threat intelligence feeds
- Other places

#1 Indicators of compromise / attack

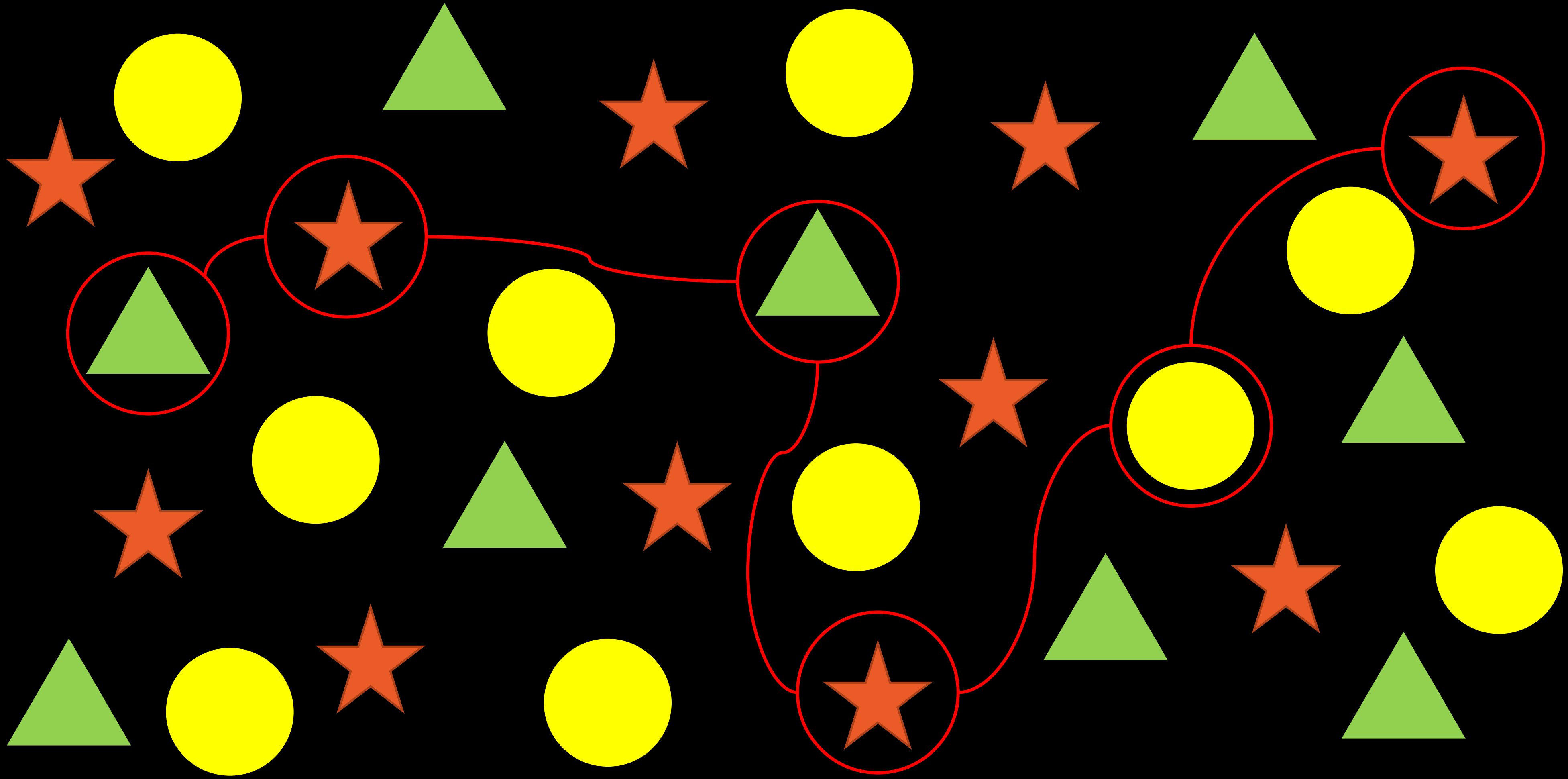
#2 Intel feeds



#3 Vulnerability scan

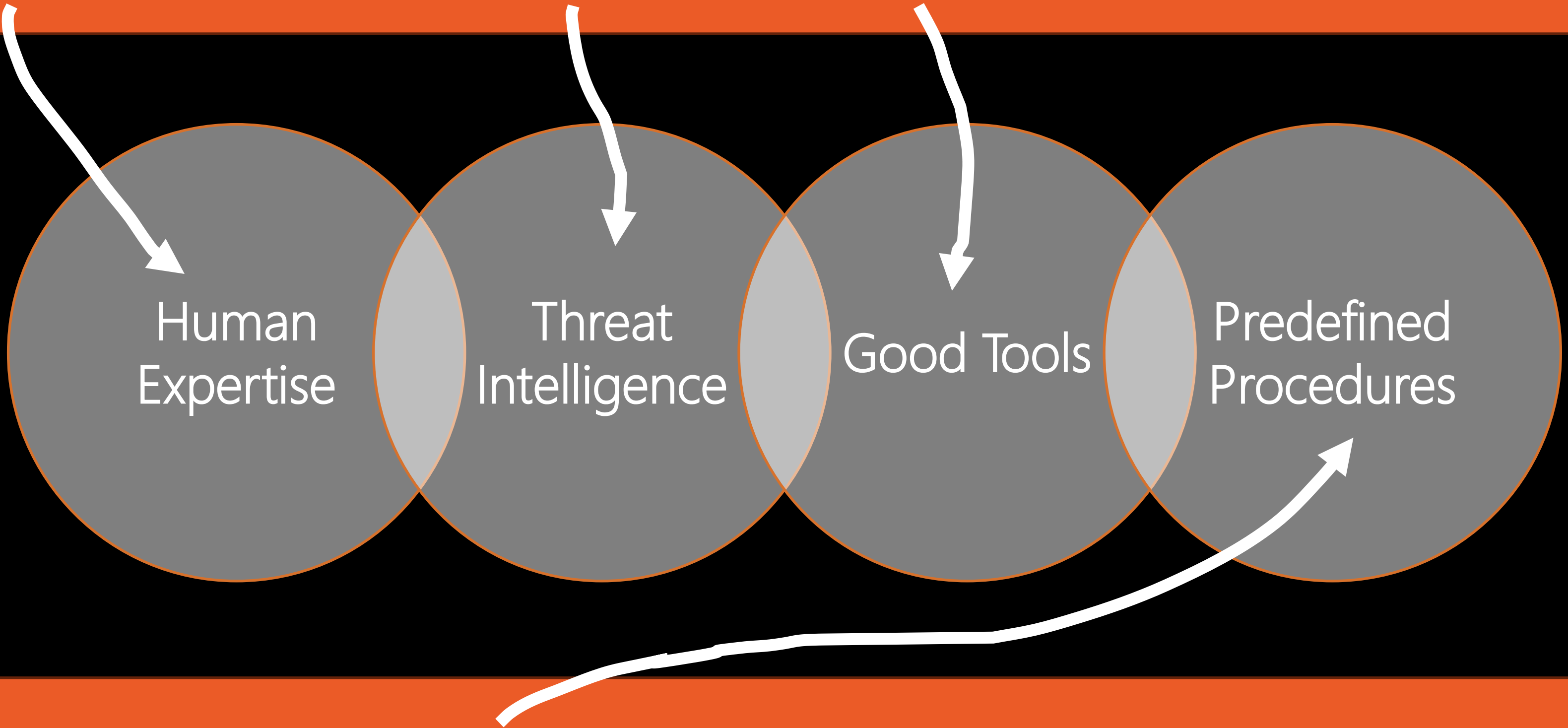


Formulating a hypothesis



Effective Threat Hunting

Threat hunters look for **information** that **indicates** ongoing attacks in the system.



2nd step is to create a **hypothesis** about how the hacker could perform the attack.

Searching for a Trace: Major Focus

- Memory & disk dumps
- Processes
- Sessions
- Network connections
- Drivers / Rootkits
- Modules (DLLs etc.)
- Injections (Shellcode etc.)
- Persistance (Services etc.)
- WMI and PowerShell activity
- Timeline
- Event Logs
- Dedicated logs
- Files that can be extracted

Searching for a Trace: Optional

- Threads
- Handles
- Hidden Processes (ActiveProcessLinks)
- Threads
- Registry
- API Hooks
- UserAssist
- Shellbags
- ShimCache
- Alternate Data Streams
- Profile anomalies (NTUSER)
- Run dialog
- Most Recently Used (MRU)
- Management Console (MMC)
- Remote Desktop connections
- Prefetch files
- Recent documents
- Automatic Destinations (LNK)
- Application Logs
- Temporary Internet Files
- Deleted files – recoverable from the disk

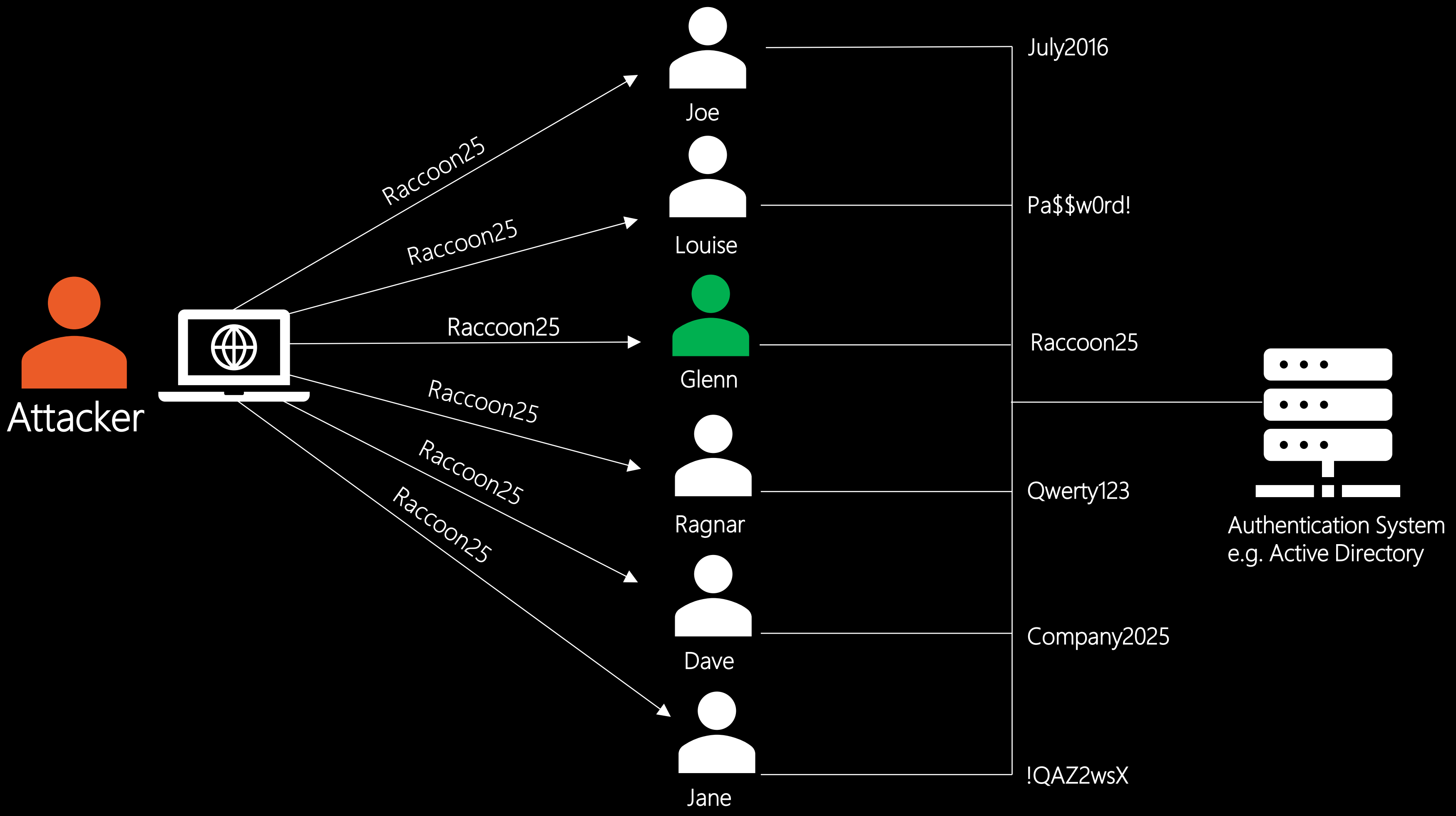
- Hiberfil.sys
- Memory dumps
- NTFS Filesystem Analysis
- Master File Table (MFT) Critical Areas
- NTFS System Files
- NTFS Metadata Attributes
- $Logfile and $UsnJrnl
- $StdInfo and $Filename
- Timestamp Manipulation
- Non-EXE files
- Resident & Nonresident files
- NTFS Structures
- B-Tree Index
- $I30 indexes

CQURE

Demo:
Is Mimikatz deleted?

Demo:
Spotting the Spray

Password Spraying



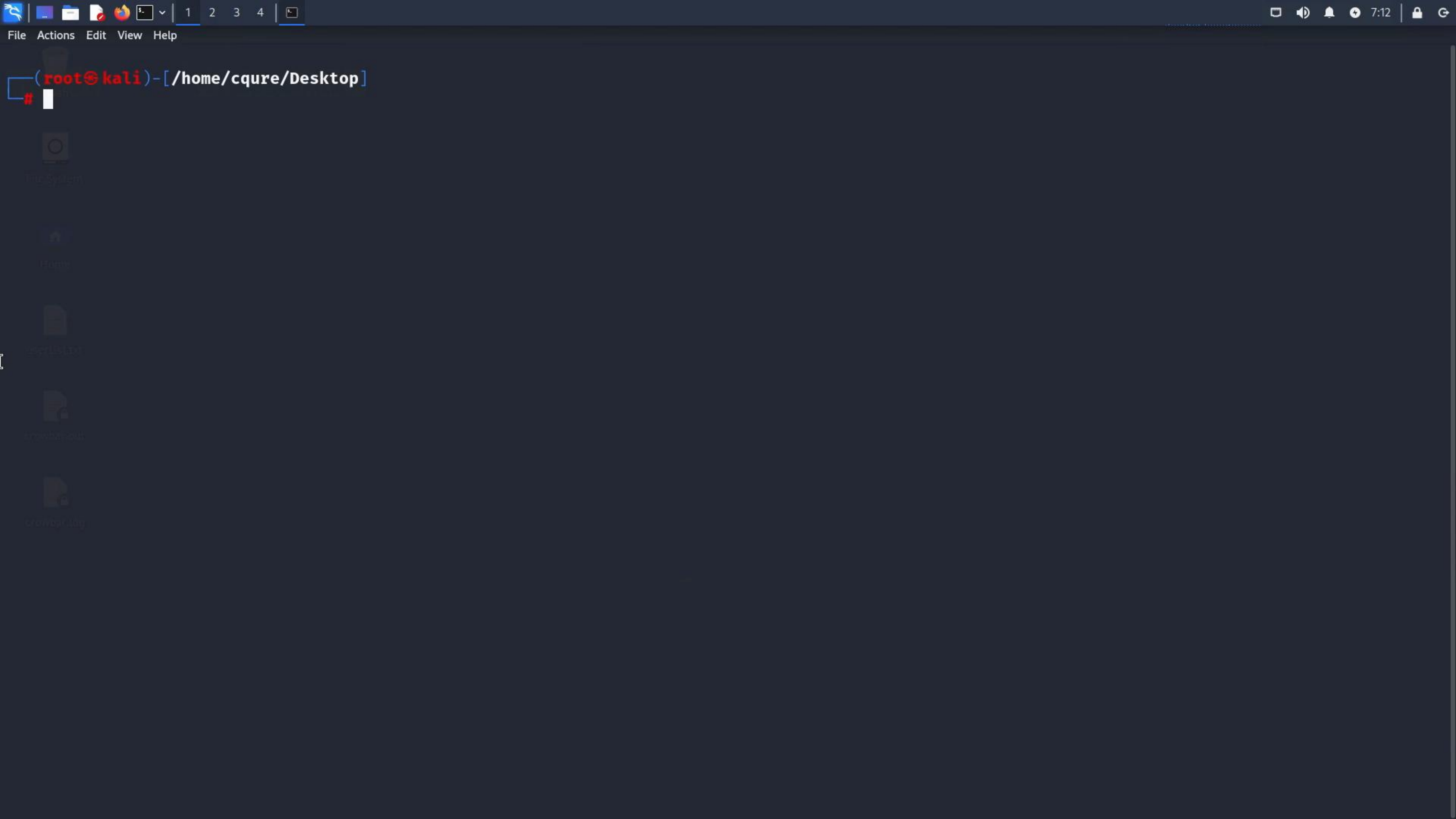
Attacker

Raccoon25
Raccoon25
Raccoon25
Raccoon25
Raccoon25
Raccoon25

Joe
Louise
Glenn
Ragnar
Dave
Jane

July2016
Pa$$w0rd!
Raccoon25
Qwerty123
Company2025
!QAZ2wsX

Authentication System
e.g. Active Directory



```
┌──(root㉿kali)-[/home/cqure/Desktop]
└─# 
```

Password Spraying :: Detection

Failed Login Attempts:

Event ID 4625: Indicates a failed logon attempt.
Key fields to focus on:
Account Name: Look for multiple accounts with single failed logins.
Logon Type: Identifies the authentication type (e.g., 2 = Interactive, 3 = Network, 10 = RemoteInteractive).
Failure Reason: 0x18 (Bad Password) is a common error code in password spraying.

Account Lockouts:

Event ID 4740: Triggers when an account is locked out.
This can indicate repeated password spraying attempts that eventually lock accounts.

Successful Logons:

Event ID 4624: Monitor for successful logons that occur shortly after multiple failed logons.
This could indicate the attacker guessed a valid password.

Password Spraying :: Detection Example

```
 $Events = Get-WinEvent -FilterHashtable @{LogName='Security';ID=4625} -
ErrorAction SilentlyContinue
$Threshold = 10 # Adjust for sensitivity


$Results = $Events | Group-Object {$_.Properties[5].Value} | Where-Object
{$_.Count -ge $Threshold}


foreach ($Result in $Results) {
    Write-Output "Potential Password Spraying Detected for IP: $($Result.Name)
- Failed Attempts: $($Result.Count)"
}
```

USN Journal

USN Journal

Types of records stored:

- Creations
- Modifications
- Deletions
- Renames

Information contained in the USN records

- Timestamp
- Filename
- Attribute
- Reason

USN Journal helps to establish a timeline and analyze it

Demo:
Data extraction and analysis with USN Journal

Windows Prefetch

Windows Prefetch

Prefetch file:

NOTEPAD.EXE-D8513F98.pf

Name of the executable

8-character hash of the location where it was run

Questions Prefetch can answer:

- What was run?
- When was it run?
- How many times?
- What DLLs were loaded?

Demo:

Digital treasure hunt with Windows Prefetch

Services

Services

Store configuration in the registry

- Always need some identity to run the executable!

Local Security Authority (LSA) Secrets

- Must be stored locally, especially when domain credentials are used
- Can be accessed when we impersonate to Local System

Their accounts should be monitored

- If you cannot use gMSA, MSA, use subscription for svc_ accounts (naming convention)

Conclusion: Think twice before using an Administrative account; use gMSA.

Demo:
Service Forensics

Cached Logons

Cached Logons: It used to be like this…

Windows 2003 / XP

The encryption algorithm is RC4.
The hash used to verify authentication is calculated as follows:

```
DCC1 = MD4(MD4(Unicode(password)) .
LowerUnicode(username))
is
DCC1 = MD4(hashNTLM . LowerUnicode(username))
```

Usage in the attack

Before the attacks facilitated by pass-the-hash, we can only rejoice in the "salting" by the username.

There are a number of pre-computed tables for users as administrators facilitating attacks on these hashes.

Cached Logons

Windows Vista / 2008 +

The encryption algorithm is AES128.

The hash used to verify authentication is calculated as follows:

MSDCC2 = PBKDF2(HMAC-SHA1, Iterations, DCC1, LowerUnicode(username))

with DCC 1 calculated in the same way as for 2003 / XP.

There is actually not much of a difference with XP / 2003! No additional salting.

PBKDF2 introduced a new variable: the number of iterations SHA1 with the same salt as before (username).

Usage in the attack

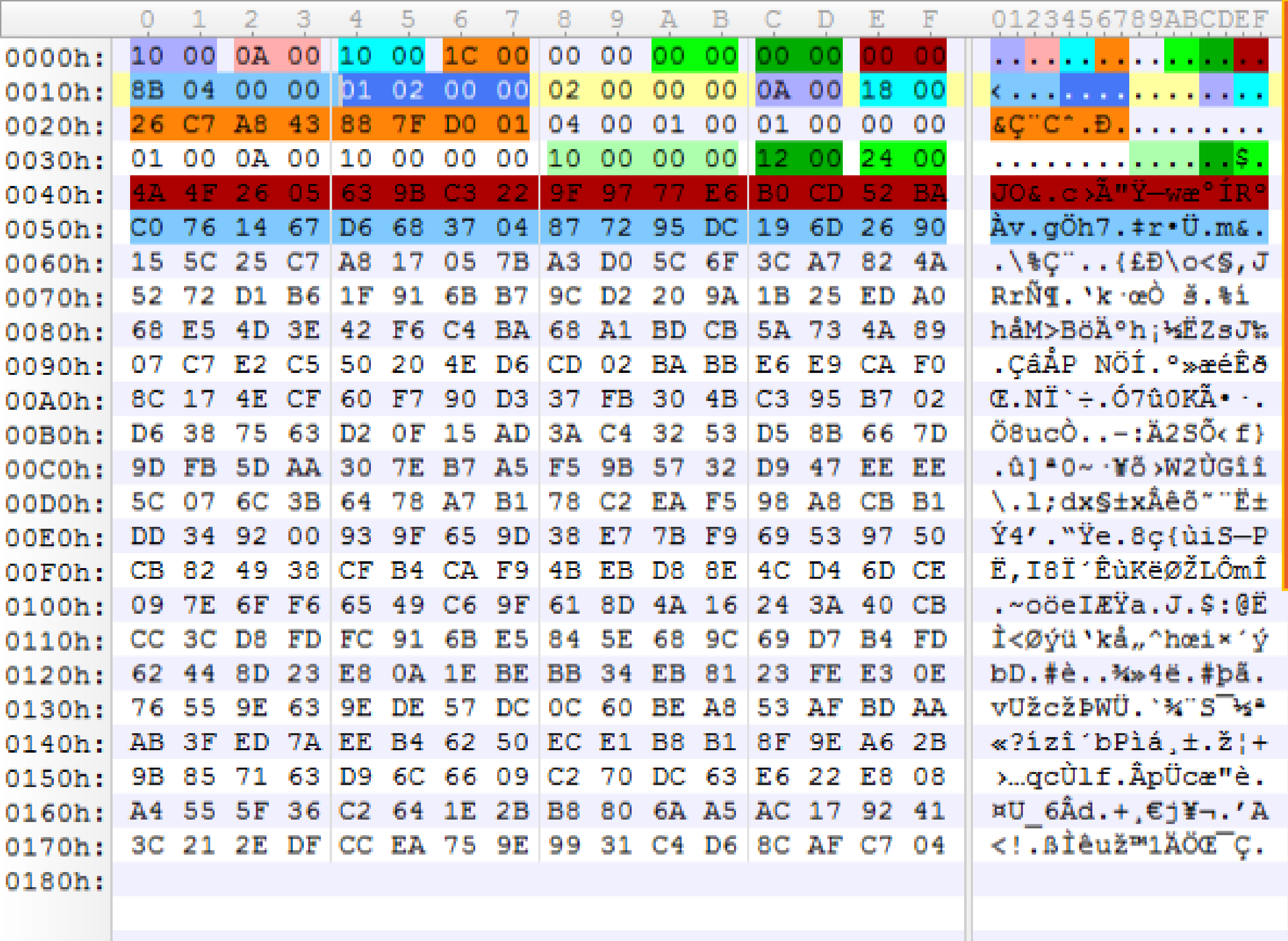
Sysmon stores a hash base

Getting the: Cached data

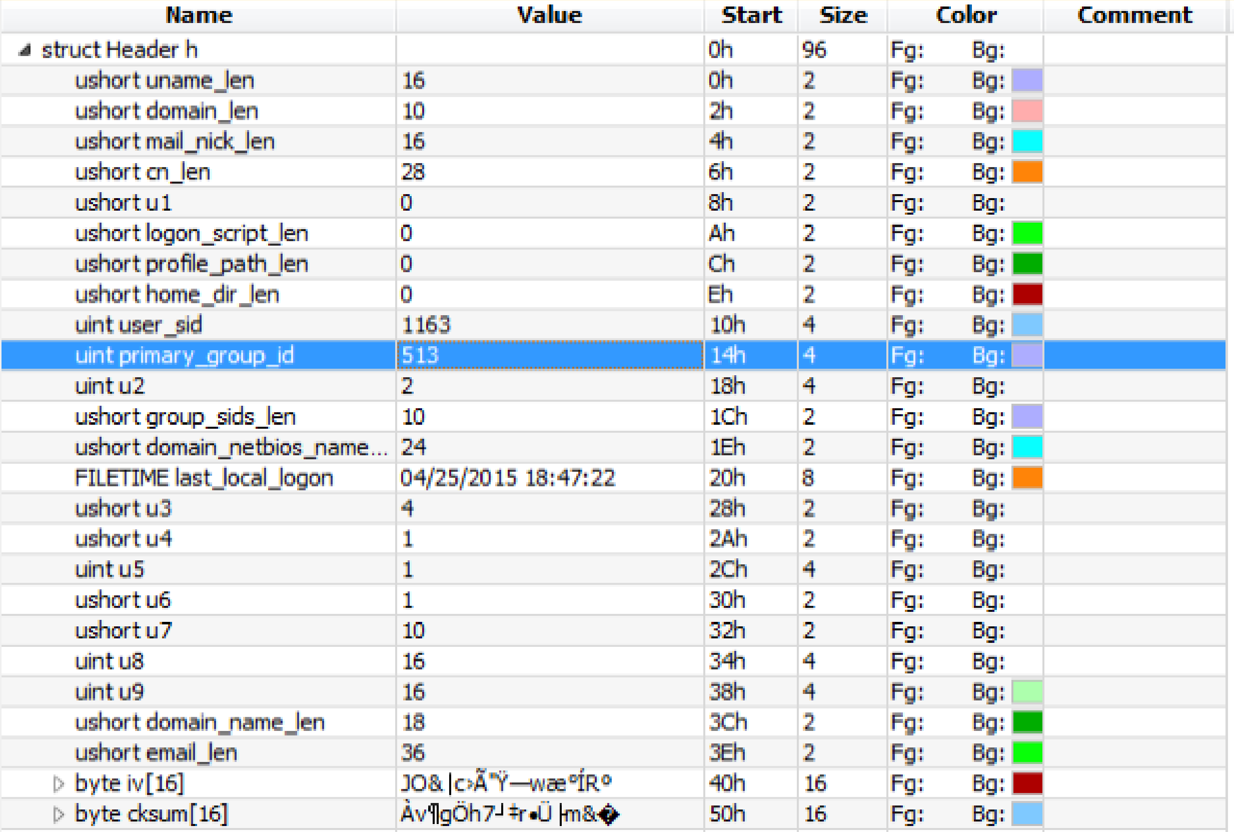
MSDCC2

1. bootkey: classes from HKLM\SYSTEM\CCS\Control\Lsa + [class names for: Data, GBG, JD, Skew1] (+arrays’ permutations) int[] permutationBootKey = new int[] { 0x8, 0x5, 0x4, 0x2, 0xb, 0x9, 0xd, 0x3, 0x0, 0x6, 0x1, 0xc, 0xe, 0xa, 0xf, 0x7 };
2. PolEKList: HKLM\SECURITY\Policy\PolEKList [default value]
3. lsakey: AES_DECRYPT(key, data) -> AES(bootkey, PolEKList)
4. NL$KM secret: HKLM\SECURITY\Policy\Secrets\NL$KM
5. nlkm_decrypted: AES_DECRYPT(lsakey, NL$KM secret)
6. Cache_Entry{id} -> HKLM\SECURITY\Cache\NL${id}
7. cache_entry_decrypted -> AES_DECRYPT(nlkm_decrypted, Cache_Entry{id})

Encrypted Cached Credentials: Legend



```
        0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F   0123456789ABCDEF
0000h: 10 00 0A 00 10 00 1C 00 00 00 00 00 00 00 00 00  ................
0010h: 8B 04 00 00 01 02 00 00 02 00 00 00 0A 00 18 00  ‹...............
0020h: 26 C7 A8 43 88 7F D0 01 04 00 01 00 01 00 00 00  &Ç¨Cˆ.Ð.........
0030h: 01 00 0A 00 10 00 00 00 10 00 00 00 12 00 24 00  ..............$.
0040h: 4A 4F 26 05 63 9B C3 22 9F 97 77 E6 B0 CD 52 BA  JO&.c›Ã"Ÿ—wæ°ÍRº
0050h: C0 76 14 67 D6 68 37 04 87 72 95 DC 19 6D 26 90  Àv.gÖh7.‡r•Ü.m&.
0060h: 15 5C 25 C7 A8 17 05 7B A3 D0 5C 6F 3C A7 82 4A  .\%Ç¨..{£Ð\o<§‚J
0070h: 52 72 D1 B6 1F 91 6B B7 9C D2 20 9A 1B 25 ED A0  RrÑ¶.‘k·œÒ š.%í 
0080h: 68 E5 4D 3E 42 F6 C4 BA 68 A1 BD CB 5A 73 4A 89  håM>BöÄºh¡½ËZsJ‰
0090h: 07 C7 E2 C5 50 20 4E D6 CD 02 BA BB E6 E9 CA F0  .ÇâÅP NÖÍ.º»æéÊð
00A0h: 8C 17 4E CF 60 F7 90 D3 37 FB 30 4B C3 95 B7 02  Œ.NÏ`÷.Ó7û0KÕ·.
00B0h: D6 38 75 63 D2 0F 15 AD 3A C4 32 53 D5 8B 66 7D  Ö8ucÒ..-:Ä2SÕ‹f}
00C0h: 9D FB 5D AA 30 7E B7 A5 F5 9B 57 32 D9 47 EE EE  .û]ª0~·¥õ›W2ÙGîî
00D0h: 5C 07 6C 3B 64 78 A7 B1 78 C2 EA F5 98 A8 CB B1  \.l;dx§±xÂêõ˜¨Ë±
00E0h: DD 34 92 00 93 9F 65 9D 38 E7 7B F9 69 53 97 50  Ý4’.“Ÿe.8ç{ùiS—P
00F0h: CB 82 49 38 CF B4 CA F9 4B EB D8 8E 4C D4 6D CE  Ë‚I8Ï´ÊùKëØŽLÔmÎ
0100h: 09 7E 6F F6 65 49 C6 9F 61 8D 4A 16 24 3A 40 CB  .~oöeIÆŸa.J.$:@Ë
0110h: CC 3C D8 FD FC 91 6B E5 84 5E 68 9C 69 D7 B4 FD  Ì<Øýü‘kå„^hœi×´ý
0120h: 62 44 8D 23 E8 0A 1E BE BB 34 EB 81 23 FE E3 0E  bD.#è..¾»4ë.#þã.
0130h: 76 55 9E 63 9E DE 57 DC 0C 60 BE A8 53 AF BD AA  vUžcžÞWÜ.`¾¨S¯½ª
0140h: AB 3F ED 7A EE B4 62 50 EC E1 B8 B1 8F 9E A6 2B  «?ízî´bPìá¸±.ž¦+
0150h: 9B 85 71 63 D9 6C 66 09 C2 70 DC 63 E6 22 E8 08  ›…qcÙlf.ÂpÜcæ"è.
0160h: A4 55 5F 36 C2 64 1E 2B B8 80 6A A5 AC 17 92 41  ¤U_6Âd.+¸€j¥¬.’A
0170h: 3C 21 2E DF CC EA 75 9E 99 31 C4 D6 8C AF C7 04  <!.ßÌêuž™1ÄÖŒ¯Ç.
0180h:
```



Name	Value	Start	Size	Color	Comment
◢ struct Header h		0h	96	Fg: Bg:	
ushort uname_len	16	0h	2	Fg: Bg: 🟪	
ushort domain_len	10	2h	2	Fg: Bg: 🟥	
ushort mail_nick_len	16	4h	2	Fg: Bg: 🟦	
ushort cn_len	28	6h	2	Fg: Bg: 🟧	
ushort u1	0	8h	2	Fg: Bg:	
ushort logon_script_len	0	Ah	2	Fg: Bg: 🟩	
ushort profile_path_len	0	Ch	2	Fg: Bg: 🟩	
ushort home_dir_len	0	Eh	2	Fg: Bg: 🟥	
uint user_sid	1163	10h	4	Fg: Bg: 🟦	
uint primary_group_id	513	14h	4	Fg: Bg: 🟪	
uint u2	2	18h	4	Fg: Bg:	
ushort group_sids_len	10	1Ch	2	Fg: Bg: 🟪	
ushort domain_netbios_name...	24	1Eh	2	Fg: Bg: 🟦	
FILETIME last_local_logon	04/25/2015 18:47:22	20h	8	Fg: Bg: 🟧	
ushort u3	4	28h	2	Fg: Bg:	
ushort u4	1	2Ah	2	Fg: Bg:	
uint u5	1	2Ch	4	Fg: Bg:	
ushort u6	1	30h	2	Fg: Bg:	
ushort u7	10	32h	2	Fg: Bg:	
uint u8	16	34h	4	Fg: Bg:	
uint u9	16	38h	4	Fg: Bg: 🟩	
ushort domain_name_len	18	3Ch	2	Fg: Bg: 🟩	
ushort email_len	36	3Eh	2	Fg: Bg: 🟩	
▷ byte iv[16]	JO& |c›Ã"Ÿ—wæ°ÍR°	40h	16	Fg: Bg: 🟥	
▷ byte cksum[16]	Àv¶gÖh7┘‡r•Ü |m&�	50h	16	Fg: Bg: 🟦	

Encrypted Cached Credentials

DK = PBKDF2(PRF, Password, Salt, c, dkLen)

Microsoft's implementation: MSDCC2= PBKDF2(HMAC-SHA1, DCC1, username, 10240, 16)

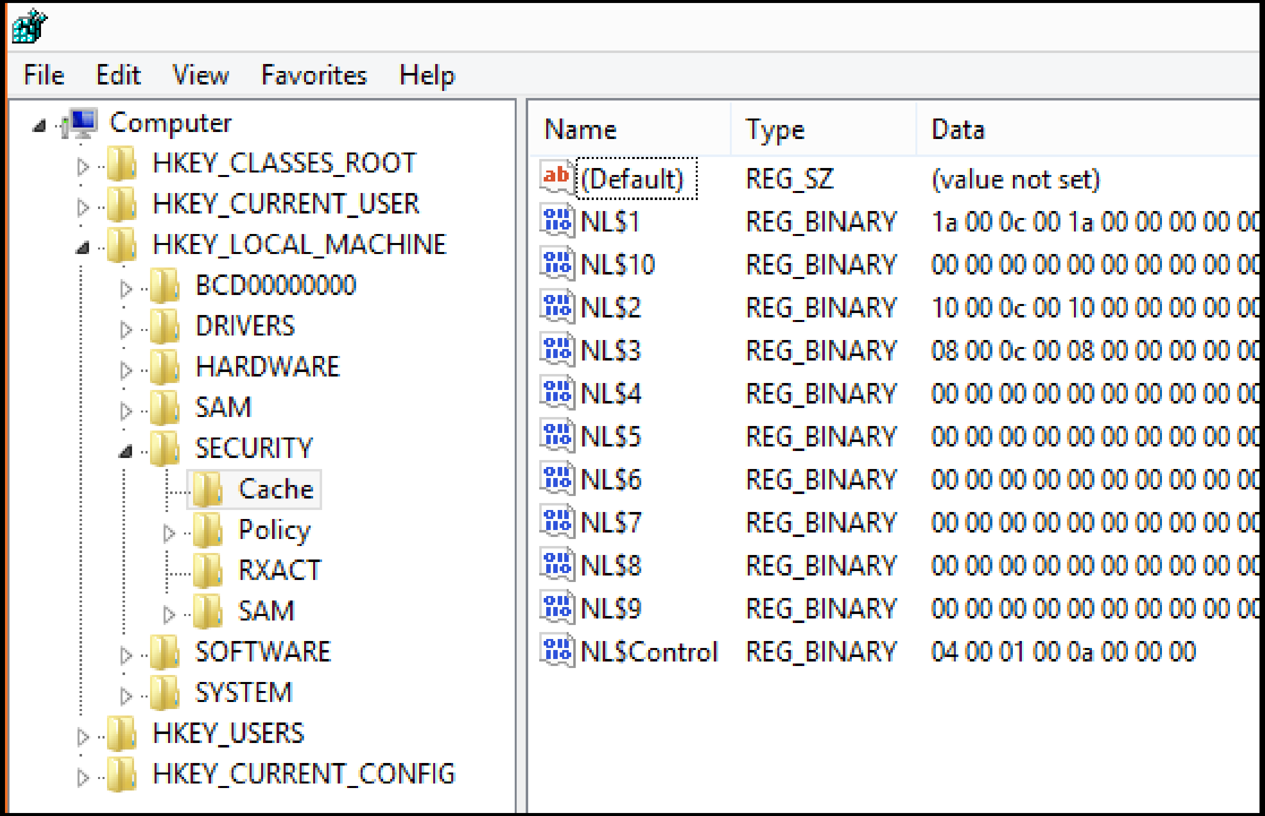
Cached Logons: Iterations

The number of iterations in PBKDF2 is configurable through the registry:

HKEY_LOCAL_MACHINE\SECURITY\Cache DWORD (32) NL$IterationCount

If the number is smaller than 10240, it is a multiplier by 1024 (20, therefore, gives 20480 iterations)

If the number is greater than 10240, it is the number of iterations (rounded to 1024)



Name	Type	Data
(Default)	REG_SZ	(value not set)
NL$1	REG_BINARY	1a 00 0c 00 1a 00 00 00 00 0
NL$10	REG_BINARY	00 00 00 00 00 00 00 00 00 0
NL$2	REG_BINARY	10 00 0c 00 10 00 00 00 00 0
NL$3	REG_BINARY	08 00 0c 00 08 00 00 00 00 0
NL$4	REG_BINARY	00 00 00 00 00 00 00 00 00 0
NL$5	REG_BINARY	00 00 00 00 00 00 00 00 00 0
NL$6	REG_BINARY	00 00 00 00 00 00 00 00 00 0
NL$7	REG_BINARY	00 00 00 00 00 00 00 00 00 0
NL$8	REG_BINARY	00 00 00 00 00 00 00 00 00 0
NL$9	REG_BINARY	00 00 00 00 00 00 00 00 00 0
NL$Control	REG_BINARY	04 00 01 00 0a 00 00 00

Demo: Extracting Cached Logons

DPAPI Blobs

Classic Data Protection API

- Based on the following components:

Password, data blob, entropy

- Is not prone to password resets!

Protects from outsiders when being in offline access
Effectively protects users data

- Stores the password history

You need to be able to get access to some of your passwords from the past

Conclusion: OS greatly helps us to protect secrets

Getting the: Classic DPAPI Secrets

DPAPI (classic)

A. MasterKey
   1. pwdhash = MD4(password) or SHA1(password)
   2. pwdhash_key = HMACSHA1(pwdhash, user_sid)
   3. PBKDF2(…, pwdhash_key,…), another elements from the file. Windows 10 no domain: HA512, AES-256, 8000 rounds
   4. Control – HMACSHA512

B. CREDHIST
   1. pwdhash = MD4(password) or SHA1(password)
   2. pwdhash_key = HMACSHA1(pwdhash, user_sid)
   3. PBKDF2(…, pwdhash_key,…), another elements from the file. Windows 10 no domain: SHA512, AES-256, 8000 rounds
   4. Control – HMACSHA512

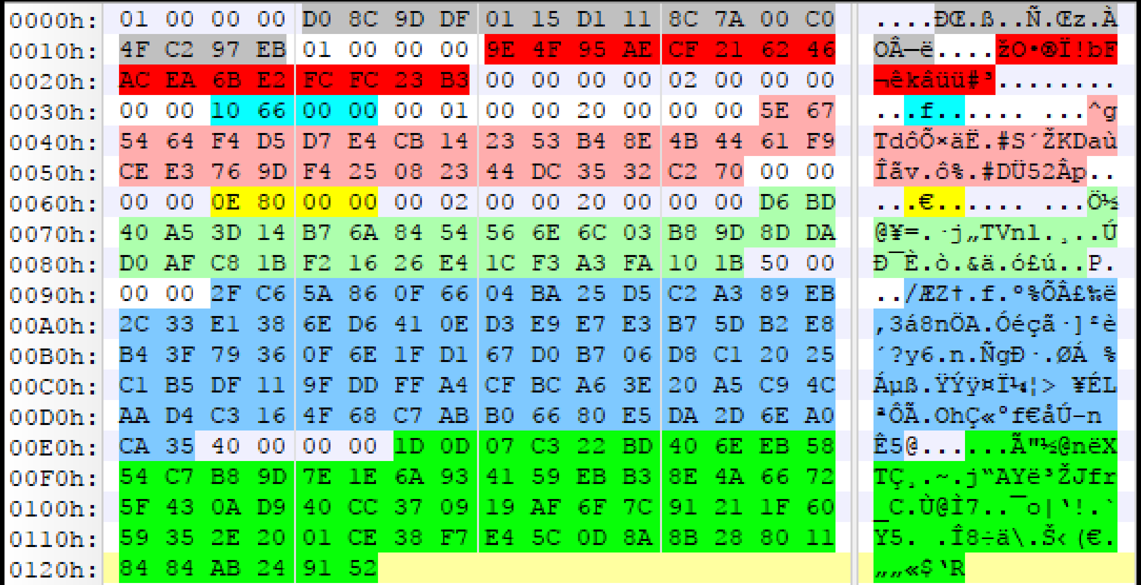
C. DPAPI blob Algorithms are written in the blob itself.

Demo:
Extracting DPAPI Secrets

Demo:
What about KeePass?

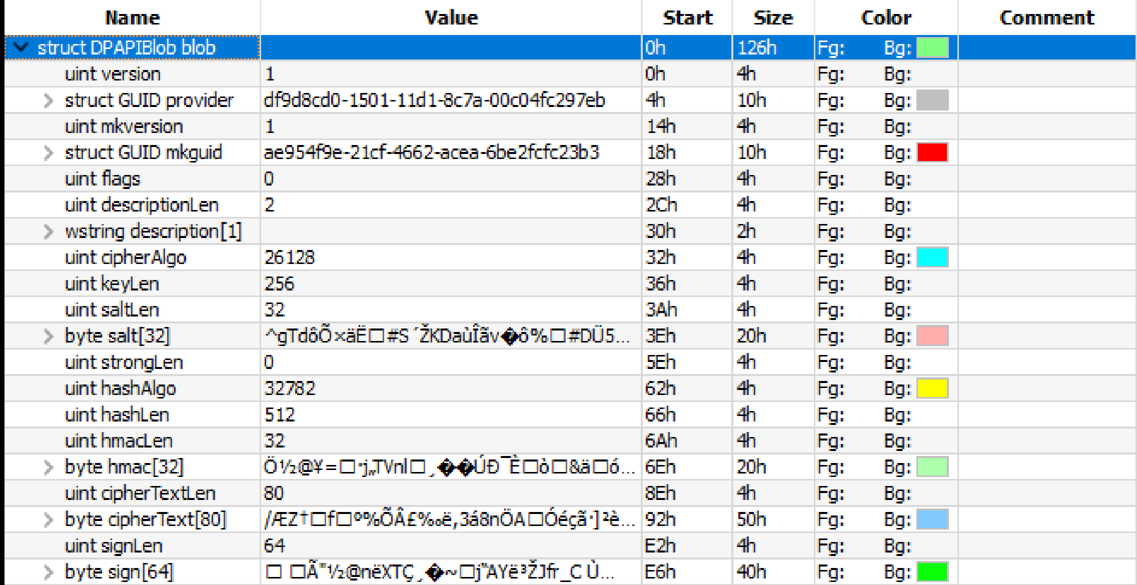
DPAPI in pictures. Example: KeePass ProtectedUserKey.bin

The master password for KeePass files encrypted & stored as cipherText (80 bytes)



```
0000h:  01 00 00 00 D0 8C 9D DF 01 15 D1 11 8C 7A 00 C0   ....ÐŒ.ß..Ñ.Œz.À
0010h:  4F C2 97 EB 01 00 00 00 9E 4F 95 AE CF 21 62 46   OÂ—ë....žO•®Ï!bF
0020h:  AC EA 6B E2 FC FC 23 B3 00 00 00 00 02 00 00 00   ¬êkâüü#³........
0030h:  00 00 10 66 00 00 00 01 00 00 20 00 00 00 5E 67   ...f...... ...^g
0040h:  54 64 F4 D5 D7 E4 CB 14 23 53 B4 8E 4B 44 61 F9   TdôÕ×äË.#S´ŽKDaù
0050h:  CE E3 76 9D F4 25 08 23 44 DC 35 32 C2 70 00 00   Îãv.ô%.#DÜ52Âp..
0060h:  00 00 0E 80 00 00 00 02 00 00 20 00 00 00 D6 BD   ...€...... ...Ö½
0070h:  40 A5 3D 14 B7 6A 84 54 56 6E 6C 03 B8 9D 8D DA   @¥=.·j„TVnl.¸..Ú
0080h:  D0 AF C8 1B F2 16 26 E4 1C F3 A3 FA 10 1B 50 00   Ð¯È.ò.&ä.ó£ú..P.
0090h:  00 00 2F C6 5A 86 0F 66 04 BA 25 D5 C2 A3 89 EB   ../ÆZ†.f.º%ÕÂ£‰ë
00A0h:  2C 33 E1 38 6E D6 41 0E D3 E9 E7 E3 B7 5D B2 E8   ,3á8nÖA.Óéçã·]²è
00B0h:  B4 3F 79 36 0F 6E 1F D1 67 D0 B7 06 D8 C1 20 25   ´?y6.n.ÑgÐ·.ØÁ %
00C0h:  C1 B5 DF 11 9F DD FF A4 CF BC A6 3E 20 A5 C9 4C   Áµß.ŸÝÿ¤Ï¼¦> ¥ÉL
00D0h:  AA D4 C3 16 4F 68 C7 AB B0 66 80 E5 DA 2D 6E A0   ªÔÃ.OhÇ«°f€åÚ-n 
00E0h:  CA 35 40 00 00 00 1D 0D 07 C3 22 BD 40 6E EB 58   Ê5@......Ã"½@nëX
00F0h:  54 C7 B8 9D 7E 1E 6A 93 41 59 EB B3 8E 4A 66 72   TÇ¸.~.j“AYë³ŽJfr
0100h:  5F 43 0A D9 40 CC 37 09 19 AF 6F 7C 91 21 1F 60   _C.Ù@Ì7..¯o|‘!.`
0110h:  59 35 2E 20 01 CE 38 F7 E4 5C 0D 8A 8B 28 80 11   Y5. .Î8÷ä\.Š‹(€.
0120h:  84 84 AB 24 91 52                                 „„«$‘R
```

DPAPI blob: Legend



Name	Value	Start	Size	Color	Comment
struct DPAPIBlob blob		0h	126h	Fg: Bg: 🟩	
uint version	1	0h	4h	Fg: Bg:	
struct GUID provider	df9d8cd0-1501-11d1-8c7a-00c04fc297eb	4h	10h	Fg: Bg: ▨	
uint mkversion	1	14h	4h	Fg: Bg:	
struct GUID mkguid	ae954f9e-21cf-4662-acea-6be2fcfc23b3	18h	10h	Fg: Bg: 🟥	
uint flags	0	28h	4h	Fg: Bg:	
uint descriptionLen	2	2Ch	4h	Fg: Bg:	
wstring description[1]		30h	2h	Fg: Bg:	
uint cipherAlgo	26128	32h	4h	Fg: Bg: 🟦	
uint keyLen	256	36h	4h	Fg: Bg:	
uint saltLen	32	3Ah	4h	Fg: Bg:	
byte salt[32]	^gTdôÕ×äË□#S´ŽKDaùÎãv�ô%□#DÜ5...	3Eh	20h	Fg: Bg: 🟥	
uint strongLen	0	5Eh	4h	Fg: Bg:	
uint hashAlgo	32782	62h	4h	Fg: Bg: 🟨	
uint hashLen	512	66h	4h	Fg: Bg:	
uint hmacLen	32	6Ah	4h	Fg: Bg:	
byte hmac[32]	Ö½@¥=□·j„TVnl□¸��ÚÐ¯È□ò□&ä□ó...	6Eh	20h	Fg: Bg: 🟩	
uint cipherTextLen	80	8Eh	4h	Fg: Bg:	
byte cipherText[80]	/ÆZ†□f□º%ÕÂ£‰ë,3á8nÖA□Óéçã·]²è...	92h	50h	Fg: Bg: 🟦	
uint signLen	64	E2h	4h	Fg: Bg:	
byte sign[64]	□ □Ã"½@nëXTÇ¸�~□j"AYë³ŽJfr_C Ù...	E6h	40h	Fg: Bg: 🟩	

CQURE

Demo:
What about RDP Connections?

Summary: what to learn?

- First and foremost, prevent unwanted programs from running.
- Block file writes to unusual places, if possible, such as public user profile in Windows.
- Reduce privileges so more advanced persistence techniques would fail due to missing privileges.
- File system permissions should be checked regularly and be as restricted as possible.
- Lock down configuration files (read-only) and put ACLs on specific registry keys.

Get to know us better!

Scan the QR code or visit
https://cqu.re/adp
to get access to this presentation
and find out more about CQURE!

DOWNLOAD THE TOOLS

https://resources.cqureacademy.com/tools/

Username: student
Password: CQUREAcademy#123!

Visit our BLOG and discover more about cybersecurity solutions & tools:

https://cqureacademy.com/blog

Thank you!

CQURE

Unleashing the Forensics Skillset:

Techniques for Extraction and Analysis of the Evidence

Paula Januszkiewicz

CQURE: CEO, Cybersecurity Expert, Penetration Tester

CQURE Academy: Trainer

Microsoft MVP on Cloud and Datacenter Management

Microsoft Regional Director

paula@cqure.us

@PaulaCqure @CQUREAcademy

www.cqureacademy.com

Unleashing the Forensics Skillset:

Techniques for Extraction and Analysis of the Evidence

Paula Januszkiewicz

Azure Days, Warsaw, 12th of March 2026

CQURE

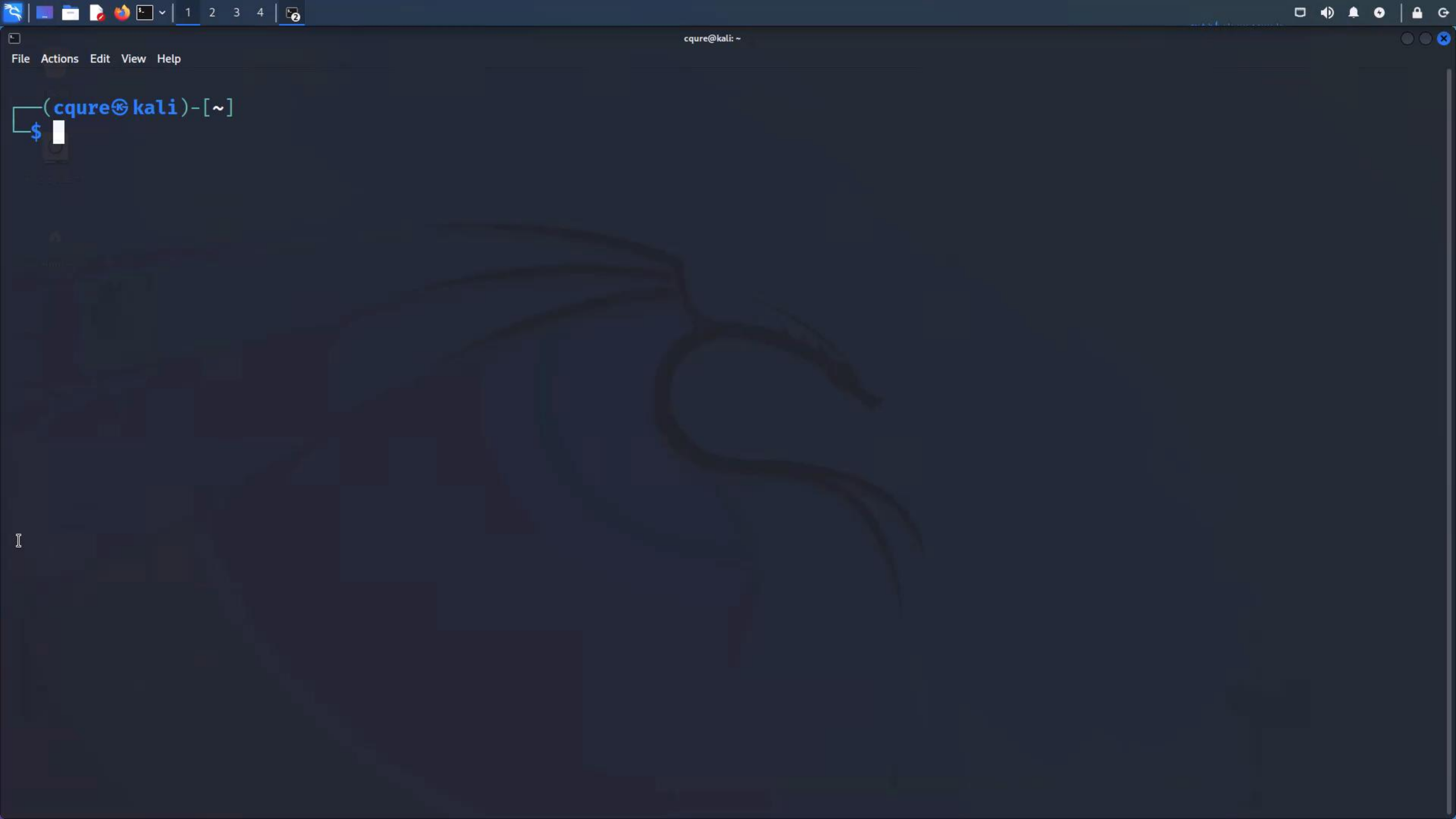
#2

Check for Legacy Configuration or Misconfiguration

Decommission of old solutions or their default settings

- Key learning points:
  - Certificate Services are often misconfigured:
    - Default certificate request website should be removed
    - Review of templates and template permissions
  - Regular audits of identity services (Active Directory etc.) are necessary
  - SQL issues – TDS provides by default a lack of encryption
  - ODBC Driver – check if it has a secure networking layer built into it

Demo:
Password Spray Escalates



```
┌──(cqure㉿kali)-[~]
└─$ 
```

#3

Absence of Insight: Look for Persistence

Persistence Through Misconfiguration

- Key learning points:
  - The best approach is to avoid using a solution until we know when it fails (time matters too)
  - Solution may have some 'backdoor weakness'
  - Some antivirus solutions can be stopped by SDDL modification for their services
  - Passwords are stored in various configuration files/places
  - Lack of monitoring adds up -> Golden Ticket etc.
  - Privilege Access Management not in place
  - Example: How to get access to the password management portal?

Persistence Used by Malware

- Key learning points:
  - Persistence through Windows mechanisms (services, task scheduler, etc.)
  - Autoruns / FRST are very helpful
  - Malware payload (e.g. PowerShell script) may be stored in the registry
  - Malware as a debugger for other apps
  - Extension hijacking (e.g. *.txt files are opened with malware by default)
  - DLL hijacking attacks
  - Callback on shutdown
  - Adding entries to Group Policy
  - WMI repository
  - Domain persistence, like AdminSDHolder, etc.

Demo:

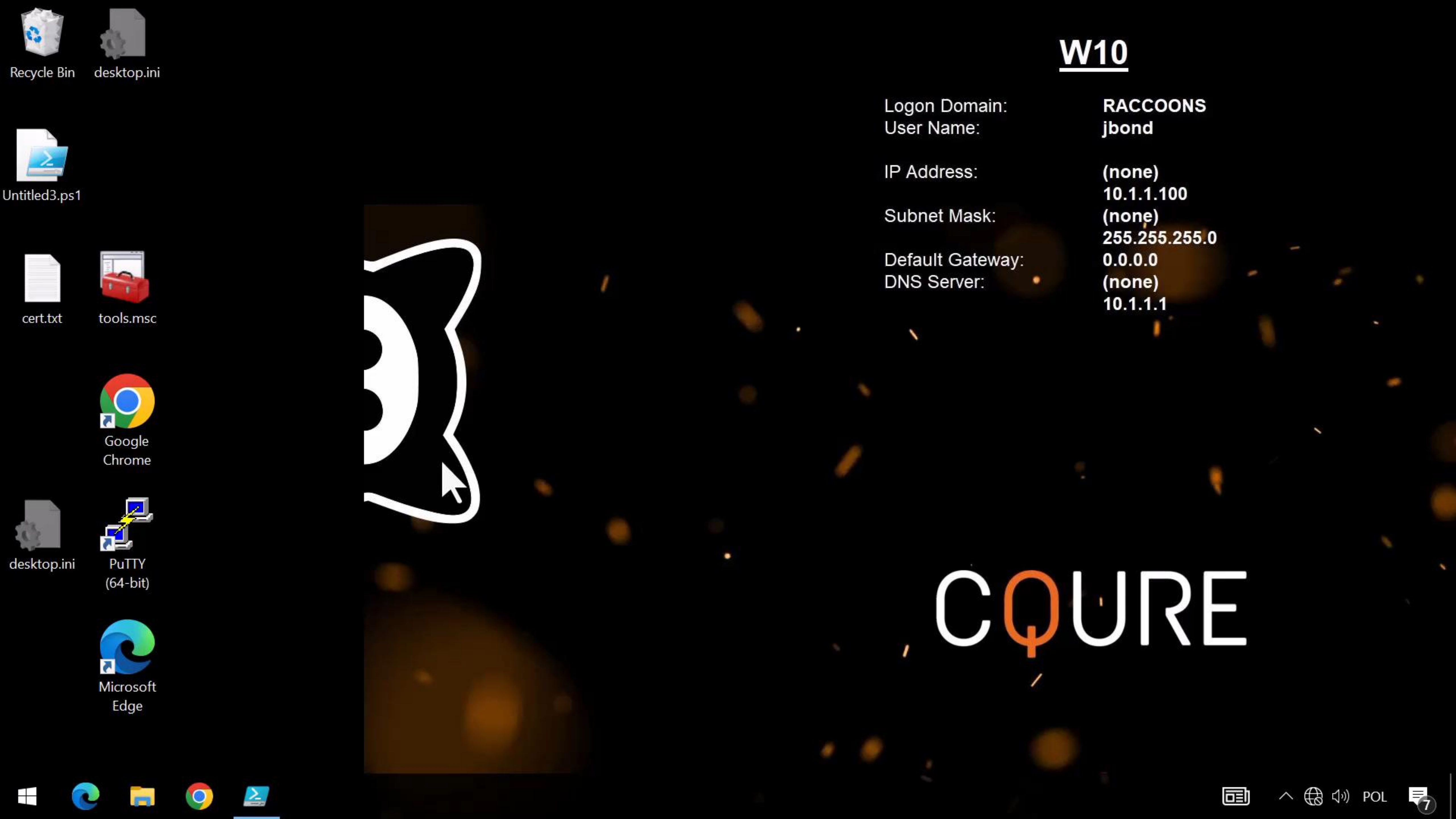
Persistence

#4

The Lack of Threat Hunting Skills

Demo:

dfsCoerce



W10

Logon Domain: RACCOONS
User Name: jbond

IP Address: (none)
10.1.1.100
Subnet Mask: (none)
255.255.255.0
Default Gateway: 0.0.0.0
DNS Server: (none)
10.1.1.1

#5

Define Incident Response Readiness

77% →	More than 77% of organizations do not test their incident response plan.
54 days →	Organizations with both an IR team and IR plan testing identified breaches 54 days faster than those with neither.
26.1 days →	The average time to investigate an incident is approximately 26.1 days, and the time to resolve incidents is an additional 17.1 days.
19% →	19% of companies highlight managing the complexities of having too many tools as a significant challenge in their digital forensics and incident response activities.
32% →	On average, SOC Team Members spend one-third of their typical workday investigating/validating incidents that are not a real threat.

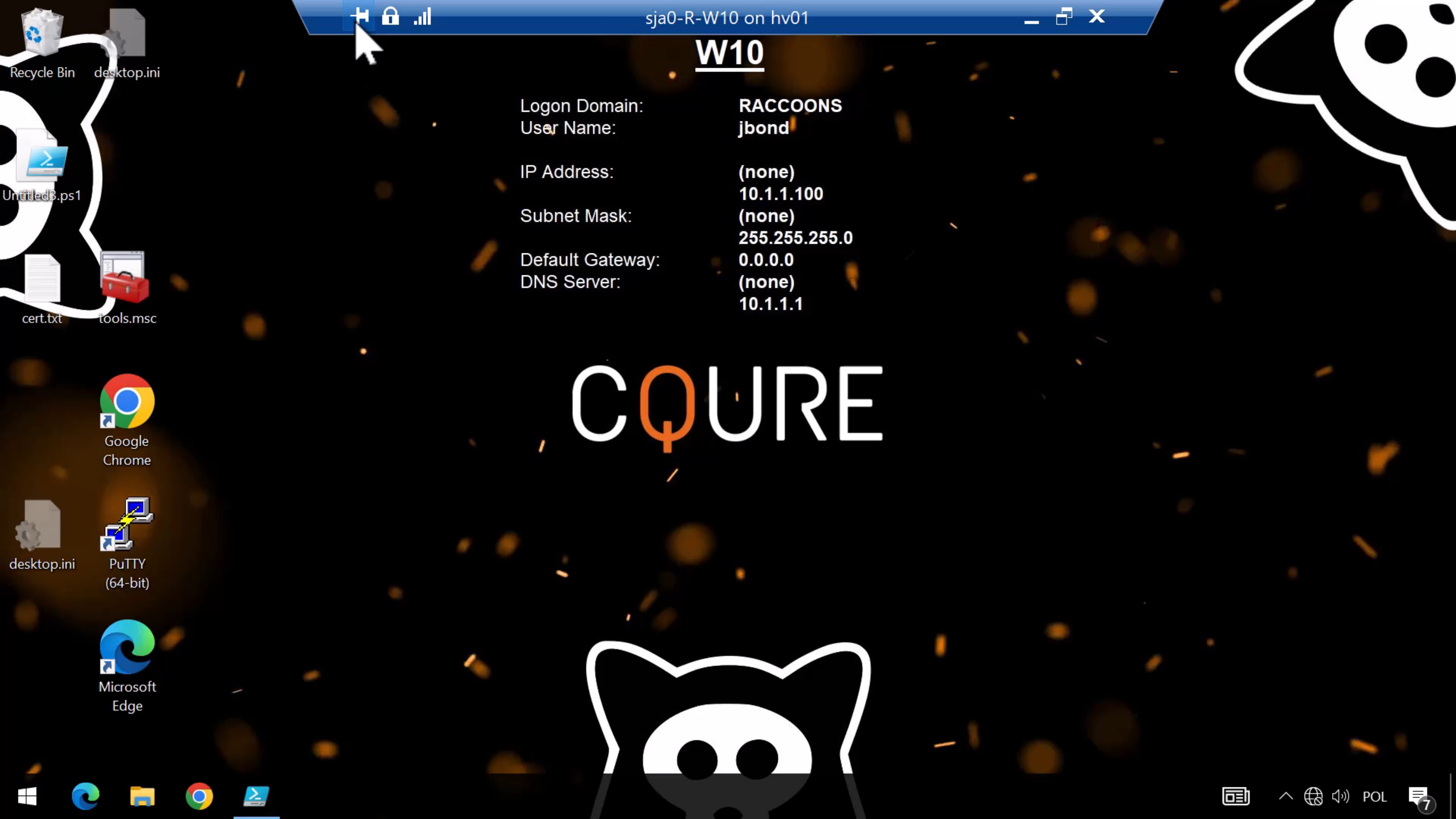
Sources: IBM Security (2023), ThriveDX (2022), IDC (2023), Morning Consult (2023)

Demo:

Incident Ready?

Demo:

NGC



W10

Logon Domain:	RACCOONS
User Name:	jbond
IP Address:	(none) 10.1.1.100
Subnet Mask:	(none) 255.255.255.0
Default Gateway:	0.0.0.0
DNS Server:	(none) 10.1.1.1

#6

Support Yourself with AI

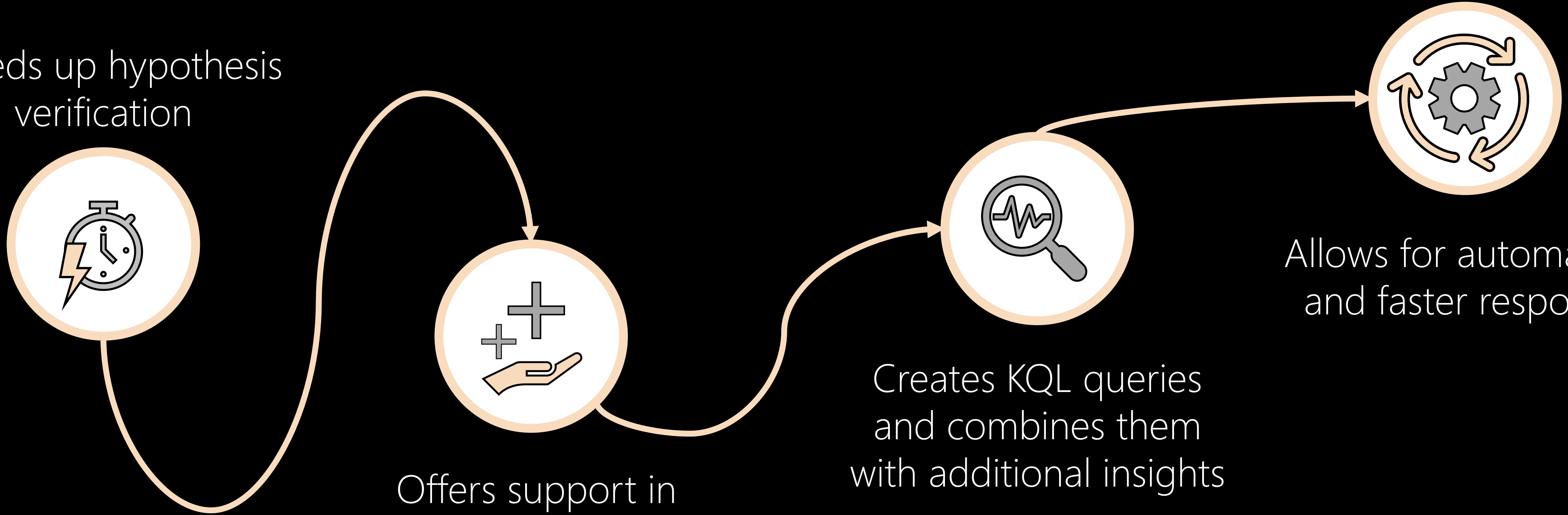
How can you use a KQL Query to prove or refute a hypothesis?

By writing custom KQL queries, cybersecurity professionals can create targeted searches to identify specific threat indicators and potentially stop attacks before they can cause significant damage.

How to create good KQL Queries?

- → Use time filters first
- → Use ‘has’ instead of ‘contains’
- → Don't filter on a calculated column
- → Select the table with the fewest rows to be the first one (left-most in query)

What does AI bring to threat hunting?



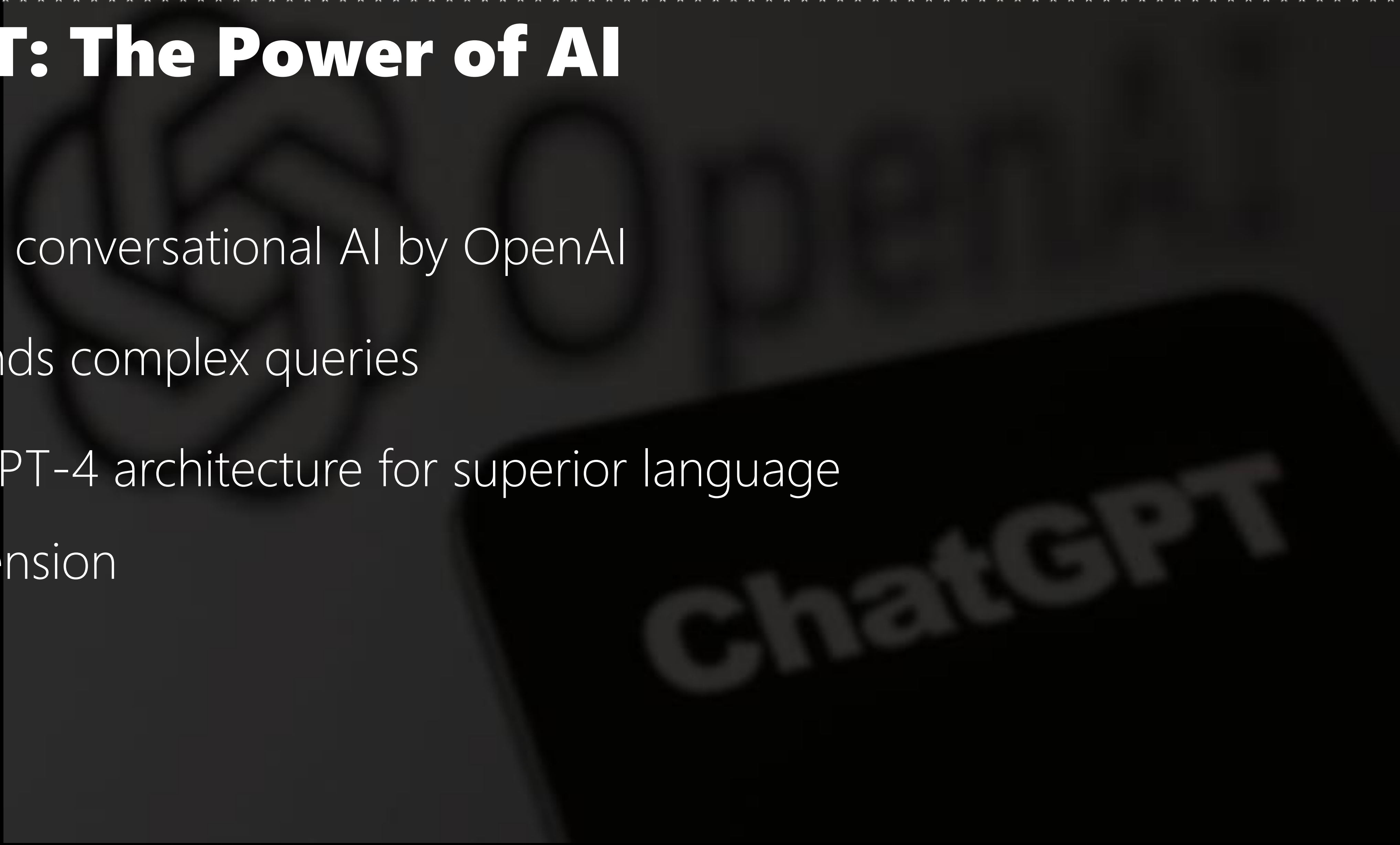
Speeds up hypothesis verification

Offers support in information enrichment

Creates KQL queries and combines them with additional insights

Allows for automation and faster response

ChatGPT: The Power of AI



- → Advanced conversational AI by OpenAI
- → Understands complex queries
- → Built on GPT-4 architecture for superior language comprehension

How ChatGPT and KQL can be used?

→ ChatGPT helps refine queries for non-experts

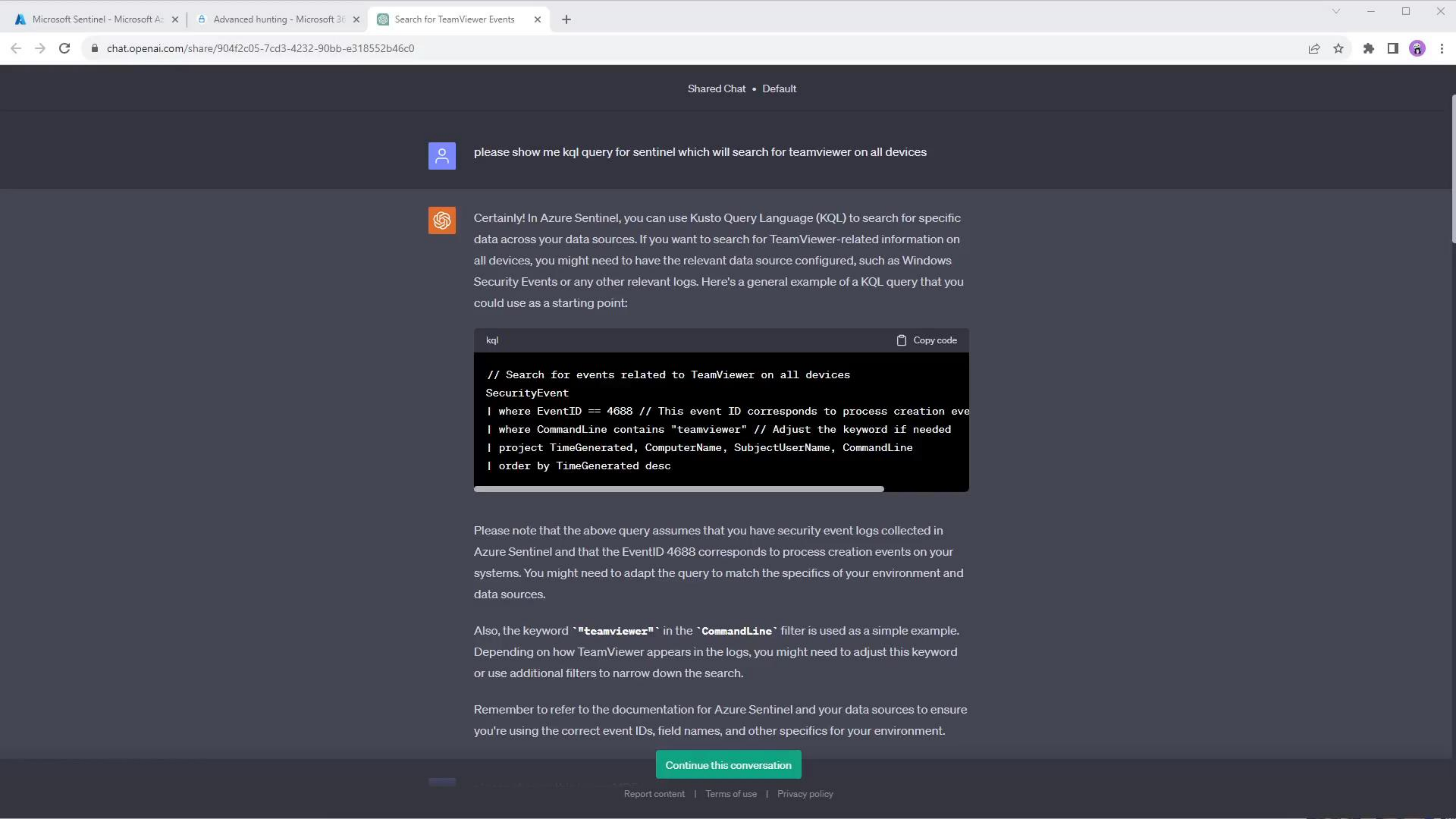
→ KQL extracts precise data from large datasets

→ Their combined power enables accurate and efficient threat detection

→ Automated threat detection using ChatGPT-guided KQL queries

Demo:

ChatGPT + KQL


Microsoft Sentinel - Microsoft Az
Advanced hunting - Microsoft 36
Search for TeamViewer Events
chat.openai.com/share/904f2c05-7cd3-4232-90bb-e318552b46c0
Shared Chat • Default
kql
Copy code
Continue this conversation
Report content | Terms of use | Privacy policy

please show me kql query for sentinel which will search for teamviewer on all devices

Certainly! In Azure Sentinel, you can use Kusto Query Language (KQL) to search for specific data across your data sources. If you want to search for TeamViewer-related information on all devices, you might need to have the relevant data source configured, such as Windows Security Events or any other relevant logs. Here's a general example of a KQL query that you could use as a starting point:

```kql
// Search for events related to TeamViewer on all devices
SecurityEvent
| where EventID == 4688 // This event ID corresponds to process creation eve
| where CommandLine contains "teamviewer" // Adjust the keyword if needed
| project TimeGenerated, ComputerName, SubjectUserName, CommandLine
| order by TimeGenerated desc
```

Please note that the above query assumes that you have security event logs collected in Azure Sentinel and that the EventID 4688 corresponds to process creation events on your systems. You might need to adapt the query to match the specifics of your environment and data sources.

Also, the keyword `"teamviewer"` in the `CommandLine` filter is used as a simple example. Depending on how TeamViewer appears in the logs, you might need to adjust this keyword or use additional filters to narrow down the search.

Remember to refer to the documentation for Azure Sentinel and your data sources to ensure you're using the correct event IDs, field names, and other specifics for your environment.