Detection Engineering for Healthcare IT Infrastructures

CSP011_W_H

CHRISTOS LAZARIDIS
FOCAL POINT

CyberSecPro

CyberSecPro creates cutting-edge education and training materials and courses to advance competencies and professionalism in EU cybersecurity.

Funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or HADEA. Neither the European Union nor the granting authority can be held responsible for them.

Project Agreement no. 101083594

Introduction to Detection Engineering in Healthcare Security

Objective: Exploring the role of detection engineering in safeguarding healthcare information systems.

Why:

- Robust security measures are critical for protecting sensitive patient data in healthcare digital services.
- Increased dynamic cyber threats pose risks to patient safety and privacy.
- Detection engineering identifies, monitors, and mitigates cybersecurity threats effectively and adapts to dynamic threats.

Healthcare services that can be monitored

- **Connected Healthcare Services:**
  - **Doctor-Patient Portals:** Platforms for communication, appointment scheduling, and access to medical records.
  - **Remote Medical Equipment Handling:**
    - Monitoring and control of medical devices used remotely for patient care.
    - Critical for telemedicine and remote diagnostics.
  - **Electronic Health Records (EHR) Systems:** Central storage and access point for patient medical data, interfacing with AD for stringent access controls.

These services are crucial for modern healthcare delivery but are vulnerable to cyber attacks.

Integration of Healthcare Services with Active Directory

- Digital healthcare services use Active Directory (AD) for identity management and access control.
- Digital healthcare Services are deployed within AD environments
- Vulnerabilities in AD can compromise the security of interconnected healthcare services.
- Securing AD environments is challenging but critical for the protection of interconnected healthcare applications.

Inherent AD vulnerabilities that can only be monitored – not mitigated

Steal or Forge Kerberos Tickets

Sub-techniques (4)

ID	Name
T1558.001	Golden Ticket
T1558.002	Silver Ticket
T1558.003	Kerberoasting
T1558.004	AS-REP Roasting

Adversaries may attempt to subvert Kerberos authentication by stealing or forging Kerberos tickets to enable Pass the Ticket. Kerberos is an authentication protocol widely used in modern Windows domain environments. In Kerberos environments, referred to as "realms", there are three basic participants: client, service, and Key Distribution Center (KDC).[1] Clients request access to a service and through the exchange of Kerberos tickets, originating from KDC, they are granted access after having successfully authenticated. The KDC is responsible for both authentication and ticket granting. Adversaries may attempt to abuse Kerberos by stealing tickets or forging tickets to enable unauthorized access.

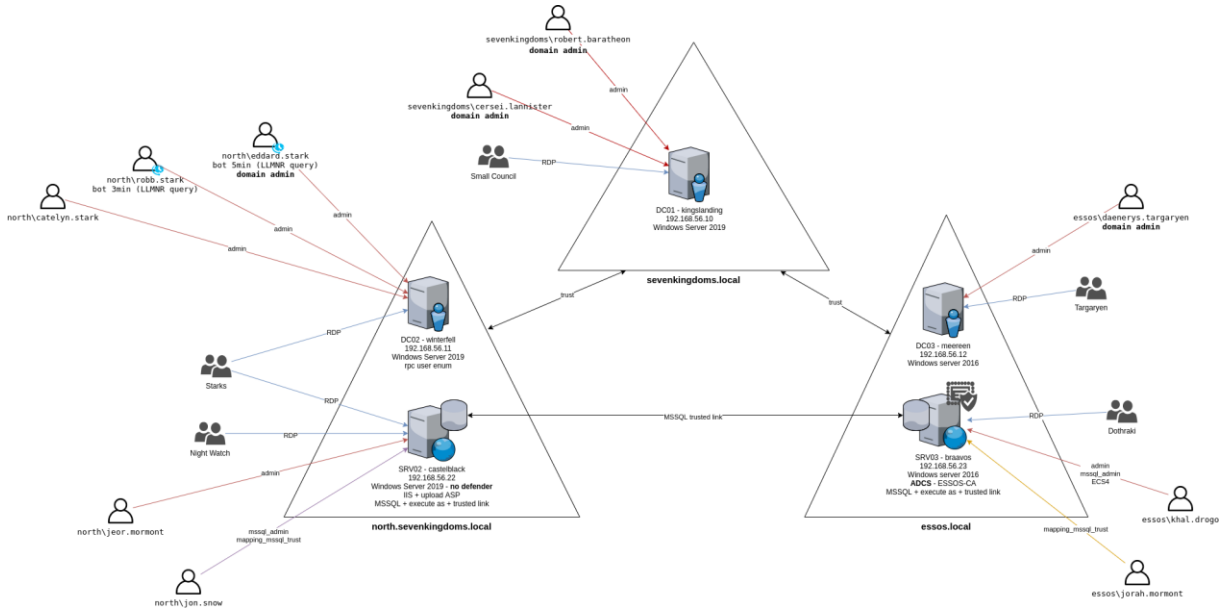
The Role of Detection Engineering in Healthcare Cybersecurity

- Detection engineering provides visibility into attacks against healthcare digital assets.
- Monitoring interactions between healthcare systems and AD environments detects suspicious activities early.
- Continuous updates to detection strategies are crucial to respond against advanced persistent threats(APTs), with tools like Microsoft Sentinel or other SIEM solutions.

Active directory Detection Workshop

WORKSHOP -Introduction to the lab environment

- The initial setup is based on the open-source project GOAD
  - Game of Active Directory (GOAD)
- 3 Domains:
  - sevenkingdoms – 1 Domain Controller (kingslanding)
  - essos – 1 Domain Controller (meereen), 1 ADCS Server (braavos)
  - north – 1 Domain Controller (winterfell), 1 IIS Server (castelblack)



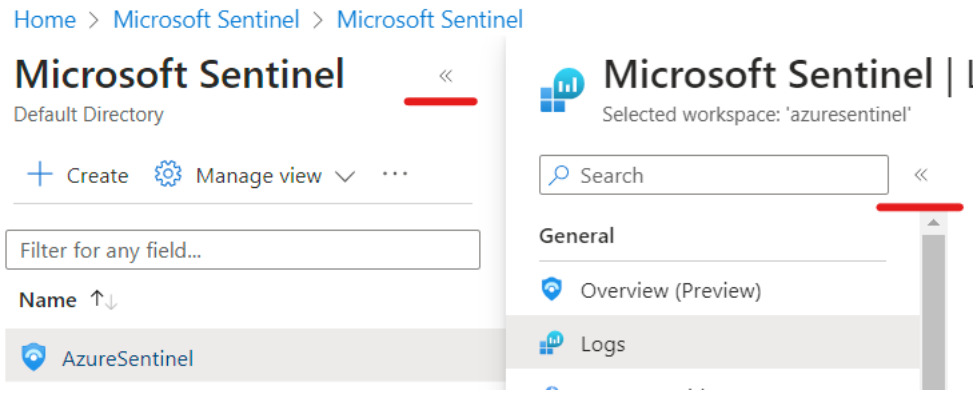
WORKSHOP - Goad Infrastructure

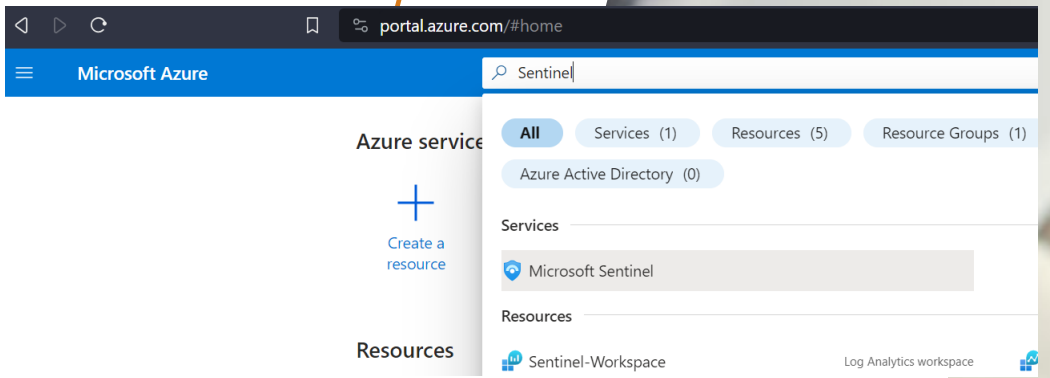
IP Address	Function	OS	Name	Domain
192.168.56.10	Windows Domain Controller	Windows Server 2019	dc01 - kingslanding	sevenkingdoms
192.168.56.11	Windows Domain Controller	Windows Server 2016	dc02 - winterfell	north.sevenkingdoms
192.168.56.12	Windows Domain Controller	Windows Server 2016	dc03 - meereen	essos
192.168.56.22	Web Server	Windows Server 2016	srv01 - castelblack	north.sevenkingdoms
192.168.56.23	ADCS Server	Windows Server 2016	srv02 - bravoos	essos
192.168.56.100	• Firewall • Intrusion Detection System - Suricata • Default Gateway • OpenVPN Server	PfSense	-	-

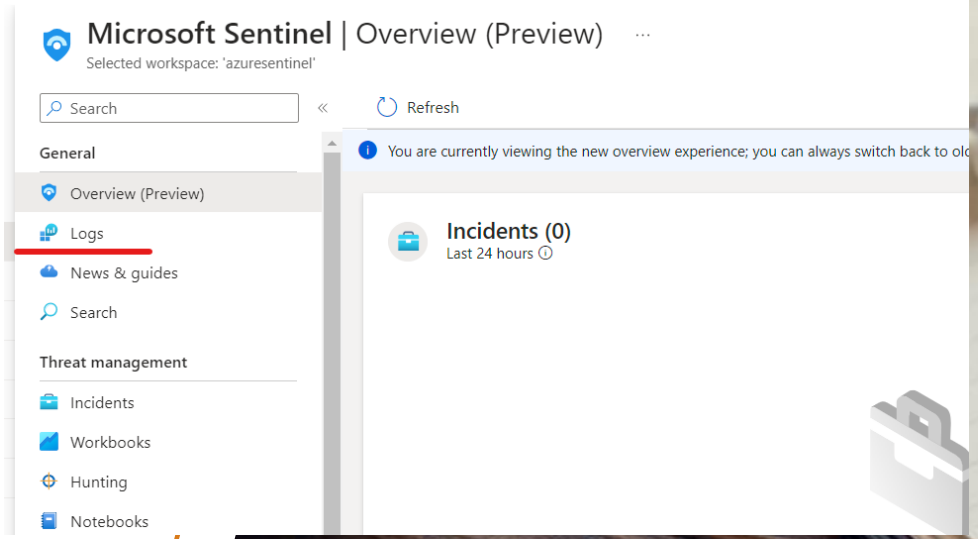
WORKSHOP

Access to Azure Sentinel

- Sentinel Access
  - Open a private tab or a guest profile in your browser
  - https://portal.azure.com
  - Username: [provided] Password: [Provided]
  - Go to search at the top of the page and write Sentinel and click on Microsoft Sentinel
  - In the column name click on the AzureSentinel
  - In the new tab go to Logs
  - Close any pop ups that appear and hide any menu to have full screen experience on the logs



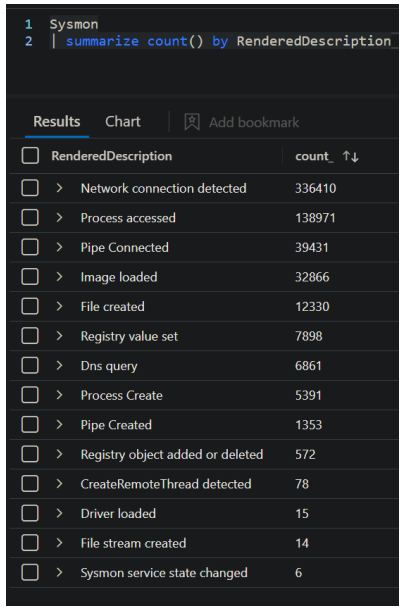


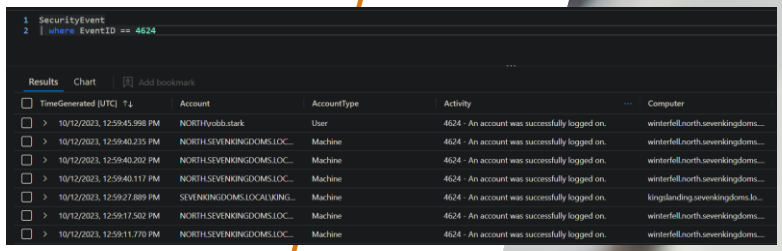


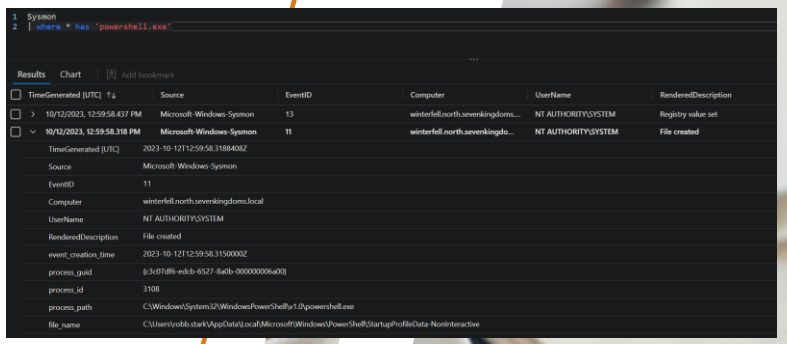
WORKSHOP

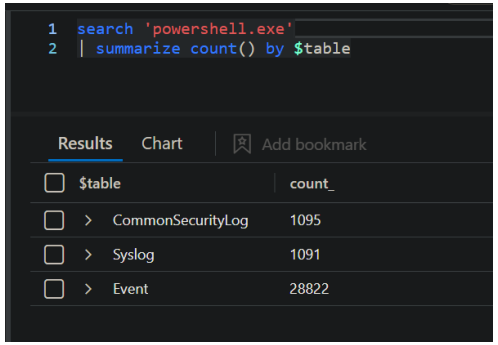
Log analysis with Kusto Query Language

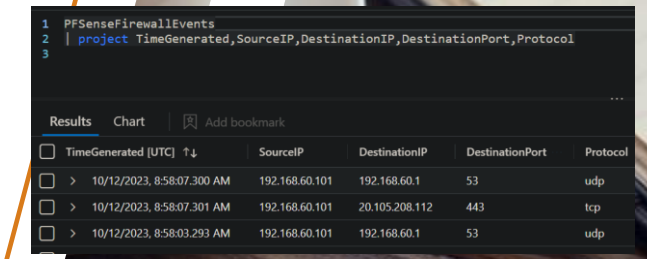
- Event tables in Lab environment
  - SecurityEvent
  - Sysmon (based on Event)
  - Suricata
  - PFSenseFirewallEvents
- Event Filtering
  - Filter by condition (where)
  - Check if any column contains string (where * has)
  - Select a subset of columns (project)
  - List unique values (distinct)
  - search operator
  - Operators on strings (==, != , has, contains)
- Event Aggregation
  - Use the summarize operator (summarize)
  - Conditionally count rows (countif())
  - Distinct Values (dcount())
  - Group data into bins (bin())
  - Time series analysis (make-series)
- Visualization
  - Visualize query results (render)











WORKSHOP

Specific log characteristics - SecurityEvent

- Provide audit information related to activities occurring on Windows Operating Systems
- Types of Events:
  - KQL:

    SecurityEvent | distinct Activity
- More information about each log type can be found in the below links
  - Windows Security Log Events

Logon Type:

This is a valuable piece of information as it tells you HOW the user just logged on:

Logon Type	Description
2	Interactive (logon at keyboard and screen of system)
3	Network (i.e. connection to shared folder on this computer from elsewhere on network)
4	Batch (i.e. scheduled task)
5	Service (Service startup)
7	Unlock (i.e. unnattended workstation with password protected screen saver)
8	NetworkCleartext (Logon with credentials sent in the clear text. Most often indicates a logon to IIS with "basic authentication") See this article for more information.
9	NewCredentials such as with RunAs or mapping a network drive with alternate credentials. This logon type does not seem to show up in any events. If you want to track users attempting to logon with alternate credentials see 4648. MS says "A caller cloned its current token and specified new credentials for outbound connections. The new logon session has the same local identity, but uses different credentials for other network connections."
10	RemoteInteractive (Terminal Services, Remote Desktop or Remote Assistance)
11	CachedInteractive (logon with cached domain credentials such as when logging on to a laptop when away from the network)

4776: The domain controller attempted to validate the credentials for an account

On this page

- Description of this event
- Field level details
- Examples

Operating Systems	Windows 2008 R2 and 7 Windows 2012 R2 and 8.1 Windows 2016 and 10 Windows Server 2019 and 2022
Category • Subcategory	Account Logon • Credential Validation
Type	Success Failure
Corresponding events in Windows 2003 and before	680 , 681

Despite what this event says, the computer is not necessarily a domain controller; member servers and workstations also log this event for logon attempts with local SAM accounts.

When a domain controller successfully authenticates a user via NTLM (instead of Kerberos), the DC logs this event. This specifies which user account who logged on (Account Name) as well as the client computer's name from which the user initiated the logon in the Workstation field.

For Kerberos authentication see event 4768, 4769 and 4771.

This event is also logged on member servers and workstations when someone attempts to logon with a local account.

Authentication Package: Always "MICROSOFT_AUTHENTICATION_PACKAGE_V1_0"

Logon Account: name of the account

Source Workstation: computer name where logon attempt originated

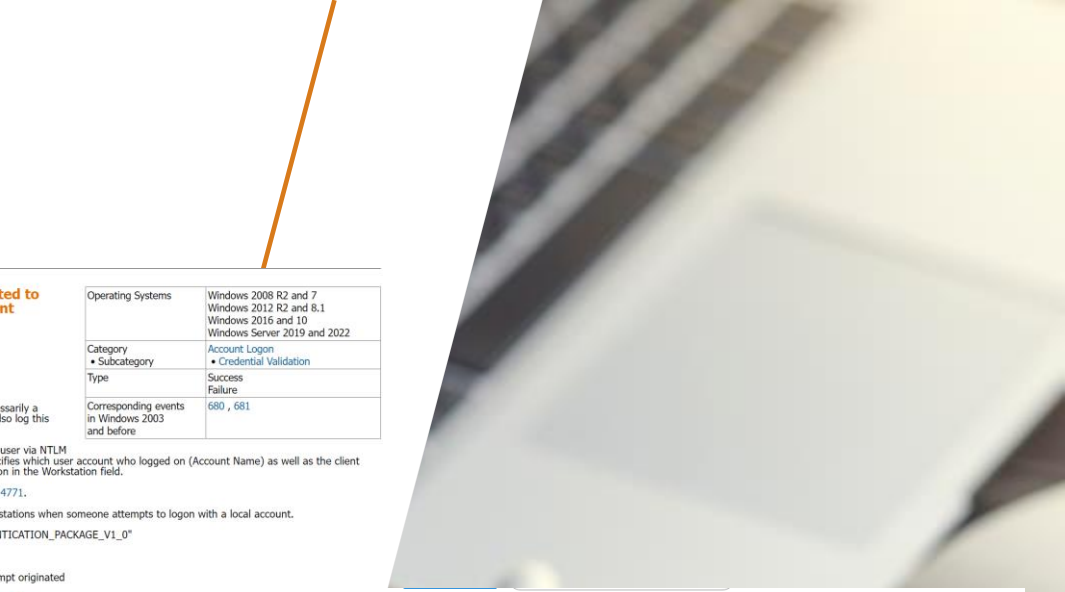
Free Security Log Resources by Randy

- Free Security Log Quick Reference Chart
- Windows Event Collection: Supercharger Free Edtion
- Free Active Directory Change Auditing Solution
- Free Course: Security Log Secrets

Description Fields in 4776

Error Code:

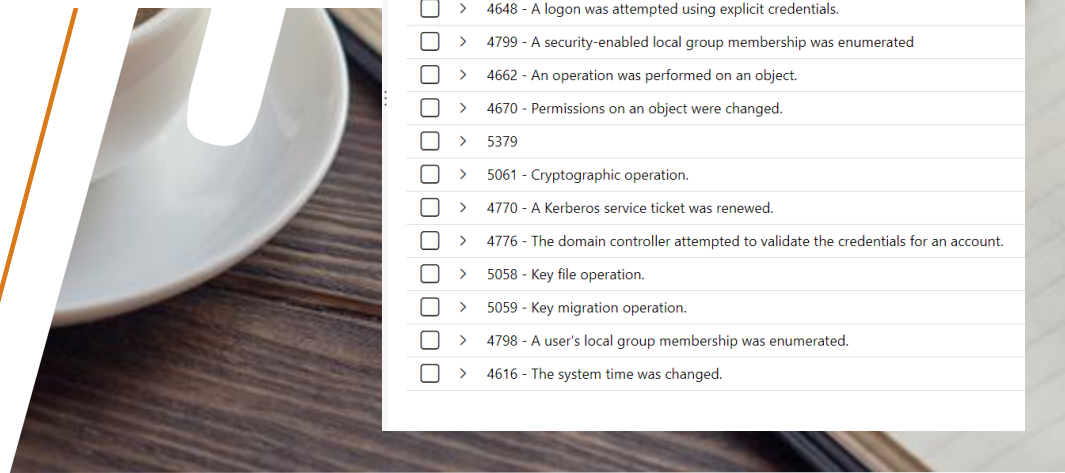
C0000064	user name does not exist
C000006A	user name is correct but the password is wrong
C0000234	user is currently locked out
C0000072	account is currently disabled
C000006F	user tried to logon outside his day of week or time of day restrictions
C0000070	workstation restriction
C0000193	account expiration
C0000071	expired password
C0000224	user is required to change password at next logon
C0000225	evidently a bug in Windows and not a risk



```
1  SecurityEvent | distinct Activity
```

Results | Chart | Add bookmark

Activity
4672 - Special privileges assigned to new logon.
4624 - An account was successfully logged on.
4634 - An account was logged off.
4768 - A Kerberos authentication ticket (TGT) was requested.
4769 - A Kerberos service ticket was requested.
4648 - A logon was attempted using explicit credentials.
4799 - A security-enabled local group membership was enumerated
4662 - An operation was performed on an object.
4670 - Permissions on an object were changed.
5379
5061 - Cryptographic operation.
4770 - A Kerberos service ticket was renewed.
4776 - The domain controller attempted to validate the credentials for an account.
5058 - Key file operation.
5059 - Key migration operation.
4798 - A user's local group membership was enumerated.
4616 - The system time was changed.



WORKSHOP

Using MITRE ATT&CK for Detection Engineering - SecurityEvent

- Type of ATT&CK Tactics detected
  - Credential Access
    - Brute Force (Event ID: 4625 or 5379)
    - Steal or Forge Kerberos Tickets: Kerberoasting (Event ID: 4769)
    - OS Credential Dumping: DCSync (Event ID: 4662)
  - Execution
    - Command and Scripting Interpreter: Windows Command Shell (Event ID: 4688)
    - System Services (Event ID: 4697)
  - Persistence
    - Scheduled Task/Job (Event ID: 4698 & 4700)
    - System Services (Event ID: 4697)
  - Privilege Escalation
    - Group Policy Modification (Event ID: 5136,5137, 5141)
  - Lateral Movement
    - Remote Desktop Protocol (RDP) (Event ID: 4624)
  - Defense Evasion
    - Use Alternate Authentication Material: Pass the Hash (Event ID: 4769, 4624)

Detection

ID	Data Source	Data Component	Detects
DS0026	Active Directory	Active Directory Object Creation	Monitor for newly constructed active directory objects, such as Windows EID 5137.
		Active Directory Object Deletion	Monitor for unexpected deletion of an active directory object, such as Windows EID 5141.
		Active Directory Object Modification	Monitor for changes made to AD settings for unexpected modifications to user accounts, such as deletions or potentially malicious changes to user attributes (credentials, status, etc.).

Detection

ID	Data Source	Data Component	Detects
DS0026	Active Directory	Active Directory Credential Request	Monitor for anomalous Kerberos activity, such as enabling Audit Kerberos Service Ticket Operations to log Kerberos TGS service ticket requests. Particularly investigate irregular patterns of activity (ex: accounts making numerous requests, Event ID 4769, within a small time frame, especially if they also request RC4 encryption [Type 0x17]).

DS0002	User Account	User Account Authentication	Monitor for user authentication attempts. From a classic Pass-The-Hash perspective, this technique uses a hash through the NTLMv1 / NTLMv2 protocol to authenticate against a compromised endpoint. This technique does not touch Kerberos. Therefore, NTLM LogonType 3 authentications that are not associated to a domain login and are not anonymous logins are suspicious. From an Over-Pass-The-Hash perspective, an adversary wants to exchange the hash for a Kerberos authentication ticket (TGT). One way to do this is by creating a sacrificial logon session with dummy credentials (LogonType 9) and then inject the hash into that session which triggers the Kerberos authentication process.

WORKSHOP

Specific log characteristics - Sysmon

- Sysmon is enabled on all Windows hosts
  - Provides Endpoint-level visibility
- Types of Events:
  - KQL:

    Sysmon | distinct RenderedDescription
- More information about each log type can be found in the below links
  - Sysmon log types
  - Windows Security Log Events (Sysmon)

```
1  Sysmon
2  | distinct RenderedDescription
```

Results | Chart | Add bookmark

RenderedDescription
File stream created
Driver loaded
Registry object added or deleted
Sysmon service state changed
CreateRemoteThread detected
Pipe Created
Registry value set
Dns query
Process Create
File created
Image loaded
Pipe Connected
Process accessed
Network connection detected

Event ID 7: Image loaded

The image loaded event logs when a module is loaded in a specific process. This event is disabled by default and needs to be configured with the "`-l`" option. It indicates the process in which the module is loaded, hashes and signature information. The signature is created asynchronously for performance reasons and indicates if the file was removed after loading. This event should be configured carefully, as monitoring all image load events will generate a significant amount of logging.

Event ID 8: CreateRemoteThread

The `CreateRemoteThread` event detects when a process creates a thread in another process. This technique is used by malware to inject code and hide in other processes. The event indicates the source and target process. It gives information on the code that will be run in the new thread: `StartAddress`, `StartModule` and `StartFunction`. Note that `StartModule` and `StartFunction` fields are inferred, they might be empty if the starting address is outside loaded modules or known exported functions.

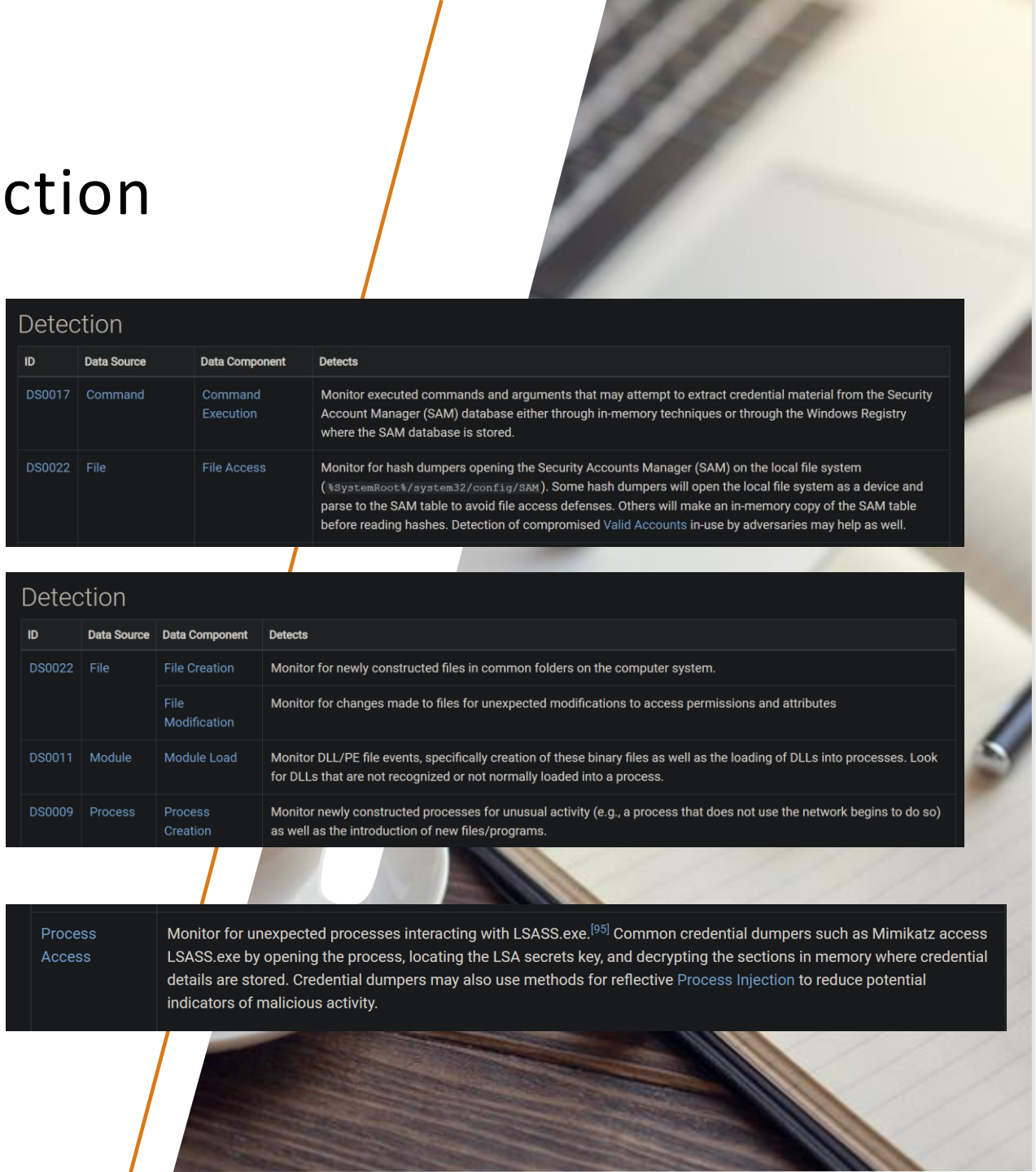
Event ID 9: RawAccessRead

The `RawAccessRead` event detects when a process conducts reading operations from the drive using the `\\.\` denotation. This technique is often used by malware for data exfiltration of files that are locked for reading, as well as to avoid file access auditing tools. The event indicates the source process and target device.

WORKSHOP

Using MITRE ATT&CK for Detection Engineering - Sysmon

- Type of ATT&CK Tactics detected
  - Initial Access
    - Phishing
    - Drive-by Compromise
  - Execution
    - Command and Scripting Interpreter
    - User Execution
  - Persistence
    - Scheduled Task/Job
    - Hijack Execution Flow: DLL Side-Loading
    - Server Software Component: Web Shell
  - Defense Evasion
    - Disable or Modify Tools
    - Obfuscated Files or Information
  - Credential Access
    - OS Credential Dumping
  - Discovery
    - System Information Discovery
    - Query Registry
  - Lateral Movement
    - Remote Desktop Protocol
    - Lateral Tool Transfer
  - Impact
    - Data Destruction
    - Disk Wipe
  - Collection
    - Data from Local System
    - Data from Removable Media
  - Exfiltration
    - Exfiltration Over C2 Channel



Detection

ID	Data Source	Data Component	Detects
DS0017	Command	Command Execution	Monitor executed commands and arguments that may attempt to extract credential material from the Security Account Manager (SAM) database either through in-memory techniques or through the Windows Registry where the SAM database is stored.
DS0022	File	File Access	Monitor for hash dumpers opening the Security Accounts Manager (SAM) on the local file system (`%SystemRoot%/system32/config/SAM`). Some hash dumpers will open the local file system as a device and parse to the SAM table to avoid file access defenses. Others will make an in-memory copy of the SAM table before reading hashes. Detection of compromised Valid Accounts in-use by adversaries may help as well.

Detection

ID	Data Source	Data Component	Detects
DS0022	File	File Creation	Monitor for newly constructed files in common folders on the computer system.
		File Modification	Monitor for changes made to files for unexpected modifications to access permissions and attributes
DS0011	Module	Module Load	Monitor DLL/PE file events, specifically creation of these binary files as well as the loading of DLLs into processes. Look for DLLs that are not recognized or not normally loaded into a process.
DS0009	Process	Process Creation	Monitor newly constructed processes for unusual activity (e.g., a process that does not use the network begins to do so) as well as the introduction of new files/programs.

Process Access	Monitor for unexpected processes interacting with LSASS.exe.[95] Common credential dumpers such as Mimikatz access LSASS.exe by opening the process, locating the LSA secrets key, and decrypting the sections in memory where credential details are stored. Credential dumpers may also use methods for reflective Process Injection to reduce potential indicators of malicious activity.

WORKSHOP

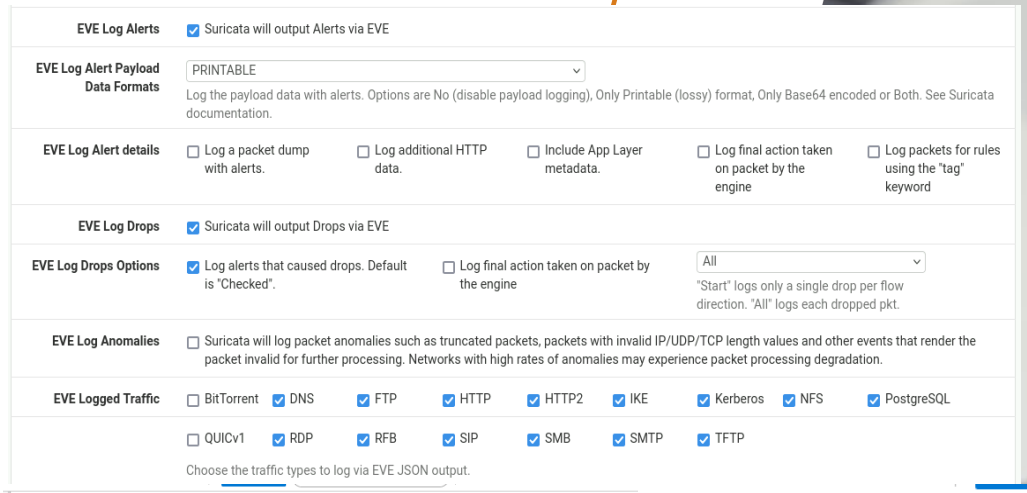
Specific log characteristics - Suricata

- Suricata logging is enabled
  - Provides excellent Network-level visibility
  - Except from alerts, it can also produce logs based on specific traffic e.g. HTTP

- Types of Events:
  - KQL:

```
Suricata | distinct event_type
| where event_type != ''
```

- Extending
  - Extra information can be extracted from specific columns based on the types of events

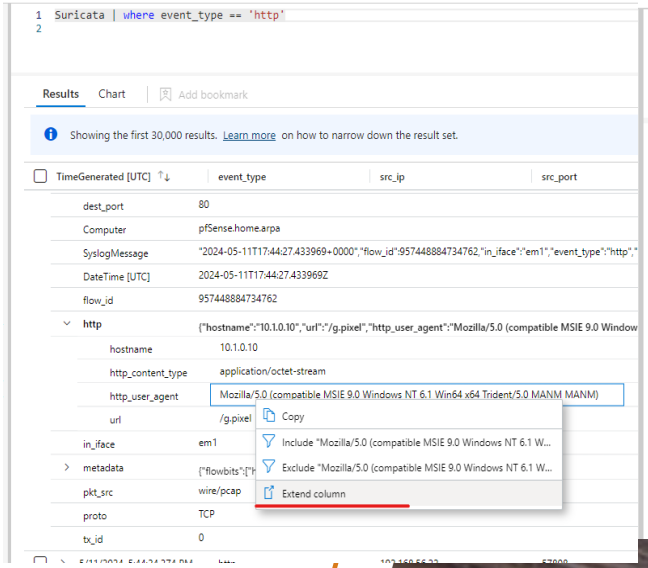


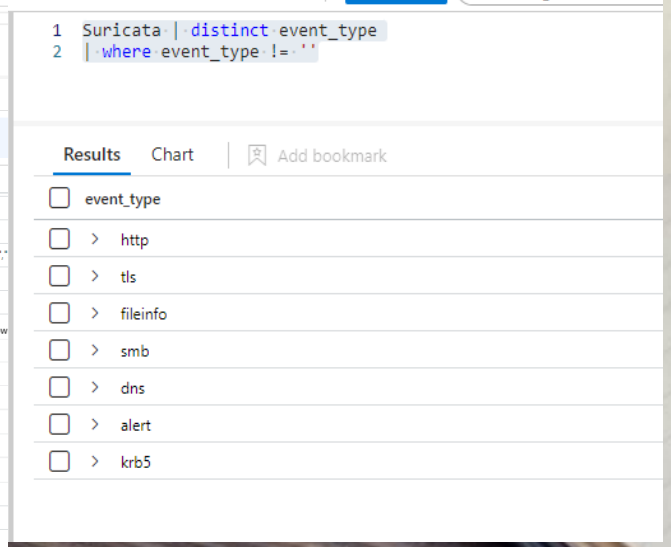

```
Suricata | where event_type == 'http'
| extend http_user_agent_ = tostring(http.http_user_agent)
```

Results | Chart | Add bookmark

Showing the first 30,000 results. Learn more on how to narrow down the result set.

TimeGenerated [UTC]	http_user_agent_	event_type
5/11/2024, 5:44:34.378 PM	Mozilla/5.0 (compatible MSIE 9.0 Windows NT 6.1 Trident/5.0) LBBROWSER	http
5/11/2024, 5:44:34.378 PM	Mozilla/5.0 (compatible MSIE 9.0 Windows NT 6.1 Win64 x64 Trident/5.0 MANM MANM)	http
5/11/2024, 5:44:34.378 PM	Mozilla/5.0 (compatible MSIE 9.0 Windows NT 6.1 Trident/5.0) LBBROWSER	http
5/11/2024, 5:44:34.377 PM	Mozilla/5.0 (compatible MSIE 9.0 Windows NT 6.1 WOW64 Trident/5.0 MALC)	http
5/11/2024, 5:44:34.377 PM	Mozilla/5.0 (compatible MSIE 9.0 Windows NT 6.1 WOW64 Trident/5.0 MANM MANM)	http
5/11/2024, 5:44:34.376 PM	Mozilla/5.0 (compatible MSIE 9.0 Windows NT 6.1 WOW64 Trident/5.0 MANM MANM)	http





WORKSHOP

Examples - Boot or Logon Autostart Execution: Registry Run Keys / Startup Folder

Boot or Logon Autostart Execution: Registry Run Keys / Startup Folder

- Placing a program within a startup folder will also cause that program to execute when a user logs in. There is a startup folder location for individual user accounts as well as a system-wide startup folder that will be checked regardless of which user account logs in.
- The startup folder path for the current user is **C:\Users\[Username]\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup**.
- The startup folder path for all users is **C:\ProgramData\Microsoft\Windows\Start Menu\Programs\StartUp**.

Steps to follow to create the logic behind the detection

Q: What data type is going to provide us this information?

A: Sysmon

Q: What type of event we want from the Sysmon data table?

A: File created

Q: Is the detection event-based or volume-based?

A: Event-based

Q: What column(s) are important so we can use them as filters?

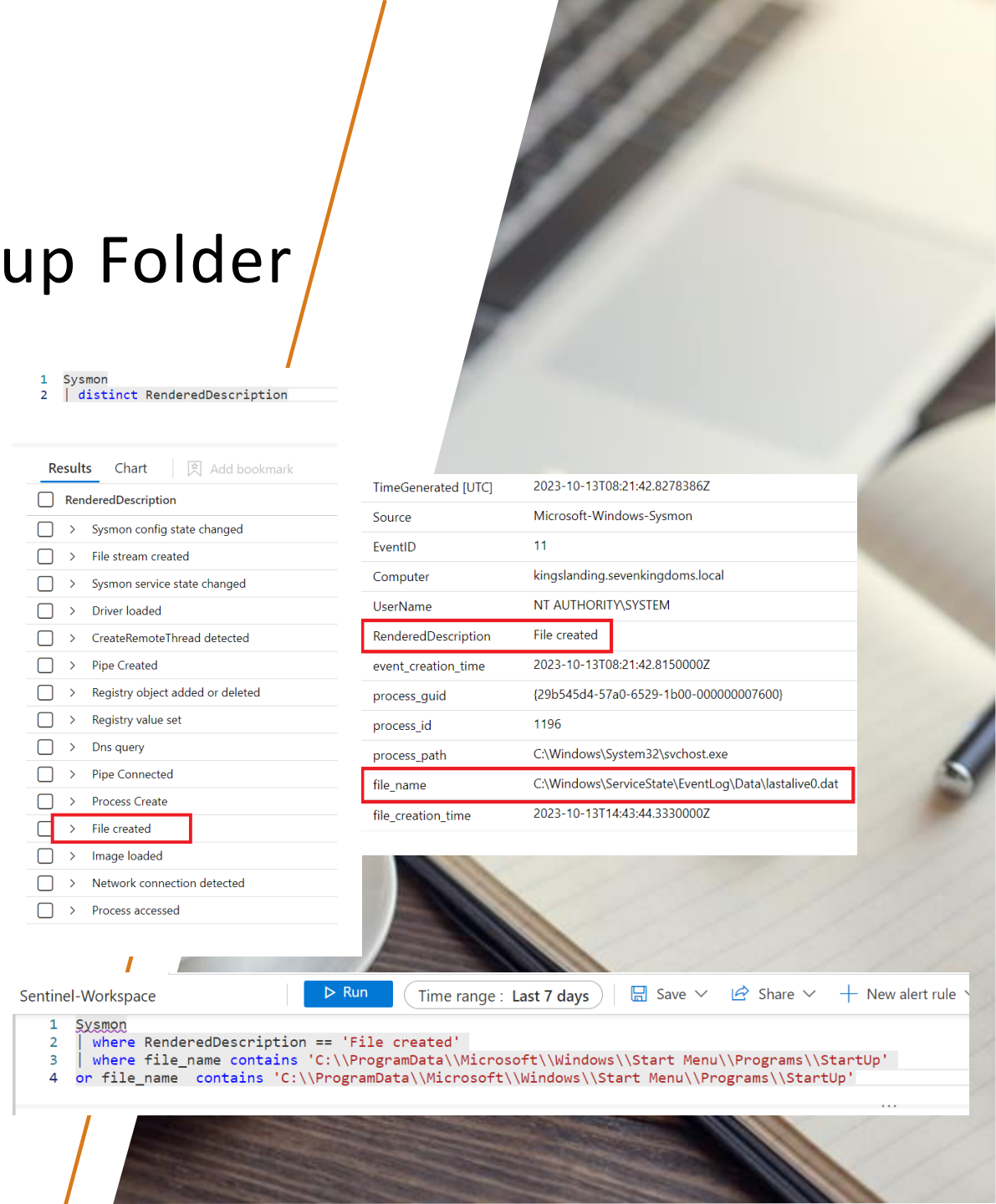
A: Rendered Description, file_name

Q: What filter I will put in the important columns:

A: 1. Rendered Description equals 'File created'

2. file_name contains '\\AppData\\Roaming\\Microsoft\\Windows\\Start Menu\\Programs\\Startup' or file_name contains 'C:\\ProgramData\\Microsoft\\Windows\\Start Menu\\Programs\\StartUp' *

*KQL syntax tip: Replace \ with \\ because the backslash is used to escape characters



WORKSHOP

Examples - Exfiltration Over Web Service

Exfiltration Over Web Service

- Adversaries may use an existing, legitimate external Web service to exfiltrate data rather than their primary command and control channel. Popular Web services acting as an exfiltration mechanism may give a significant amount of cover due to the likelihood that hosts within a network are already communicating with them prior to compromise. Firewall rules may also already exist to permit traffic to these services.
- Web service providers also commonly use SSL/TLS encryption, giving adversaries an added level of protection.

Steps to follow to create the logic behind the detection

Q: What data type is going to provide us this information?

A: Sysmon or PFSenseFirewallEvents

Q: From those 2 data tables which provides info about the number of bytes transferred

A: PFSenseFirewallEvents

Q: Is the detection event-based or volume-based?

A: Volume-based

Q: What function of the SIEM is important for volume-based detections?

A: Event Aggregation (summarize)

Q: What column(s) are important so we can use them as filters?

A: DestinationPort, Protocol, DeviceAction and Destination IP should be Public

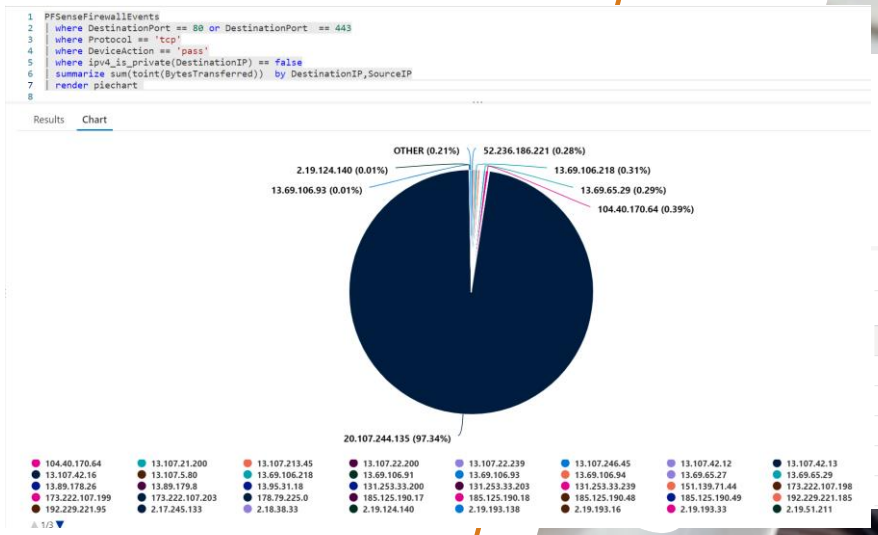
Q: What columns are important for the aggregation?

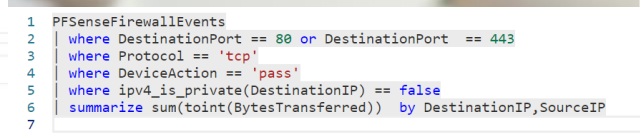
A: SourceIP, DestinationIP, BytesTransferred

Q: How we need to aggregate the results?

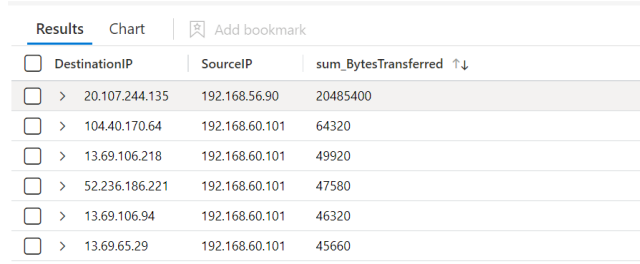
A: By summing the BytesTransferred per Destination and Source IP

Q (BONUS) : Is it better to display aggregation results as data or graphs?



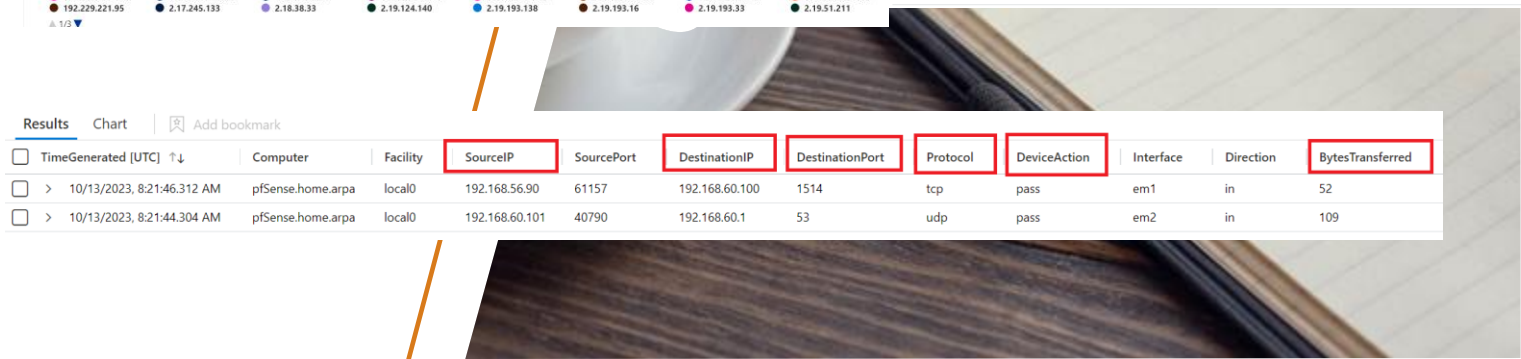


```
PFSenseFirewallEvents
| where DestinationPort == 80 or DestinationPort  == 443
| where Protocol == 'tcp'
| where DeviceAction == 'pass'
| where ipv4_is_private(DestinationIP) == false
| summarize sum(toint(BytesTransferred))  by DestinationIP,SourceIP
```



Results | Chart | Add bookmark

DestinationIP	SourceIP	sum_BytesTransferred
20.107.244.135	192.168.56.90	20485400
104.40.170.64	192.168.60.101	64320
13.69.106.218	192.168.60.101	49920
52.236.186.221	192.168.60.101	47580
13.69.106.94	192.168.60.101	46320
13.69.65.29	192.168.60.101	45660



Results | Chart | Add bookmark

TimeGenerated [UTC]	Computer	Facility	SourceIP	SourcePort	DestinationIP	DestinationPort	Protocol	DeviceAction	Interface	Direction	BytesTransferred
10/13/2023, 8:21:46.312 AM	pfSense.home.arpa	local0	192.168.56.90	61157	192.168.60.100	1514	tcp	pass	em1	in	52
10/13/2023, 8:21:44.304 AM	pfSense.home.arpa	local0	192.168.60.101	40790	192.168.60.1	53	udp	pass	em2	in	109

WORKSHOP

Examples - Account Discovery: Domain Account

- Account Discovery: Domain Account
  - Adversaries may attempt to get a listing of domain accounts. This information can help adversaries determine which domain accounts exist to aid in follow-on behavior such as targeting specific accounts which possess particular privileges.
  - Discovers valid usernames by brute force querying likely usernames against a Kerberos service. When an invalid username is requested, the server will respond using the Kerberos error code KRB5KDC_ERR_C_PRINCIPAL_UNKNOWN, allowing us to determine that the username was invalid. Valid usernames will illicit either the TGT in a AS-REP response or the error KRB5KDC_ERR_PREAUTH_REQUIRED, signaling that the user is required to perform pre authentication
  - Tools: krbrute, nmap, crackmapexec

- Steps to follow to create the logic behind the detection - Endpoint level

Q: What data type is going to provide us this information?
A: SecurityEvents

Q: What type of event we want from the SecurityEvents data table?
A: 4768 - A Kerberos authentication ticket (TGT) was requested.

Q: Is the detection event-based or volume-based?
A: Volume-based

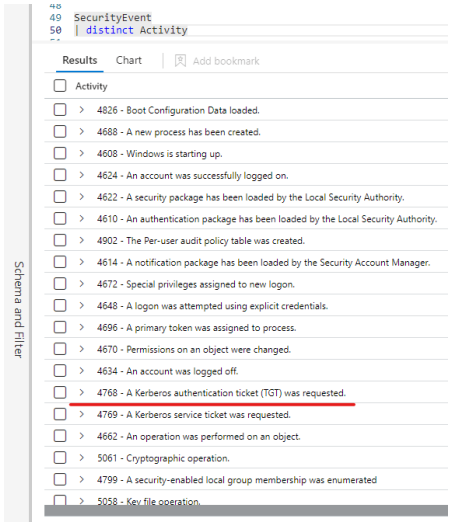
Q: What function of the SIEM is important for volume-based detections?
A: Event Aggregation (summarize)

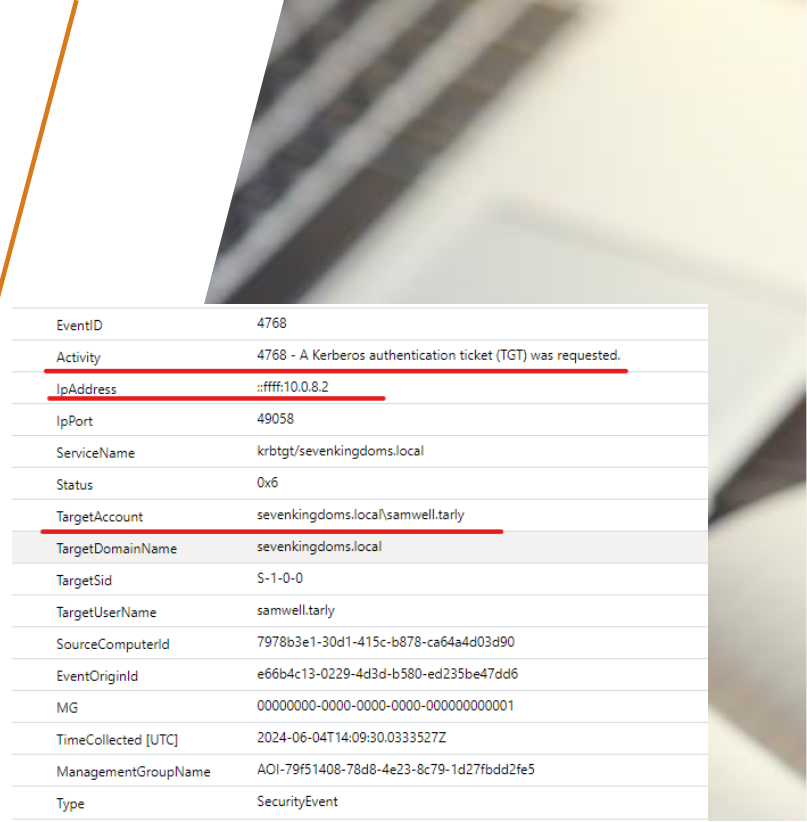
Q: What column(s) are important so we can use them as filters?
A: Activity

Q: What columns are important for the aggregation?
A: IpAddress, TargetAccount, TimeGenerated

Q: How we need to aggregate the results?
A: IP Address field is from where the Kerberos Ticket requests are originating. So, we look for IP addresses with Kerberos Ticket requests for multiple different accounts.

Q (BONUS) : Is it better to display aggregation results as data or graphs?
A: Graphs (render columnchart)






```
SecurityEvent
| where Activity == "4768 - A Kerberos authentication ticket (TGT) was requested."
| summarize dcount(TargetAccount),make_set(TargetAccount) by IpAddress,bin(TimeGenerated,1h),Activity
```

esults Chart Add bookmark

IpAddress	TimeGenerated [UTC]	Activity	dcount_TargetAccount	set_TargetAccount
::ffff:10.0.8.2	6/4/2024, 2:00:00.000 PM	4768 - A Kerberos authenticati...	152	["sevenkingdoms.local\\nmap","seve
::1	6/4/2024, 1:00:00.000 PM	4768 - A Kerberos authenticati...	8	["SEVENKINGDOMS.LOCAL\\KINGSL
::1	6/4/2024, 11:00:00.000 AM	4768 - A Kerberos authenticati...	8	["SEVENKINGDOMS.LOCAL\\KINGSL
::ffff:192.168.56.22	6/4/2024, 1:00:00.000 PM	4768 - A Kerberos authenticati...	5	["north.sevenkingdoms.local\\CASTE
::ffff:192.168.56.22	6/4/2024, 11:00:00.000 AM	4768 - A Kerberos authenticati...	5	["north.sevenkingdoms.local\\sql_svc
::1	6/4/2024, 12:00:00.000 PM	4768 - A Kerberos authenticati...	3	["NORTH\\robb.stark","NORTH\\edd
::ffff:192.168.56.23	6/4/2024, 1:00:00.000 PM	4768 - A Kerberos authenticati...	3	["ESSOS.LOCAL\\BRAAVOS$","essos.l

WORKSHOP

Examples - Brute Force: Password Spraying

Brute Force: Password Spraying

- Adversaries may use a single or small list of commonly used passwords against many different accounts to attempt to acquire valid account credentials. Password spraying uses one password (e.g., 'Password01'), or a small list of commonly used passwords, that may match the complexity policy of the domain. Logins are attempted with that password against **many different accounts** on a network to avoid account lockouts that would normally occur when brute forcing a single account with many passwords.

Steps to follow to create the logic behind the detection

Q: What data type is going to provide us this information?

A: SecurityEvents

Q: What type of event we want from the SecurityEvents data table?

A: 4625 - An account failed to log on.

Q: Is the detection event-based or volume-based?

A: Volume-based

Q: What function of the SIEM is important for volume-based detections?

A: Event Aggregation (summarize)

Q: What column(s) are important so we can use them as filters?

A: None actually. By filtering for the EventID 4625 we have the dataset that we want to investigate.

Q: What columns are important for the aggregation?

A: IPAddress, Account.

Q: How we need to aggregate the results?

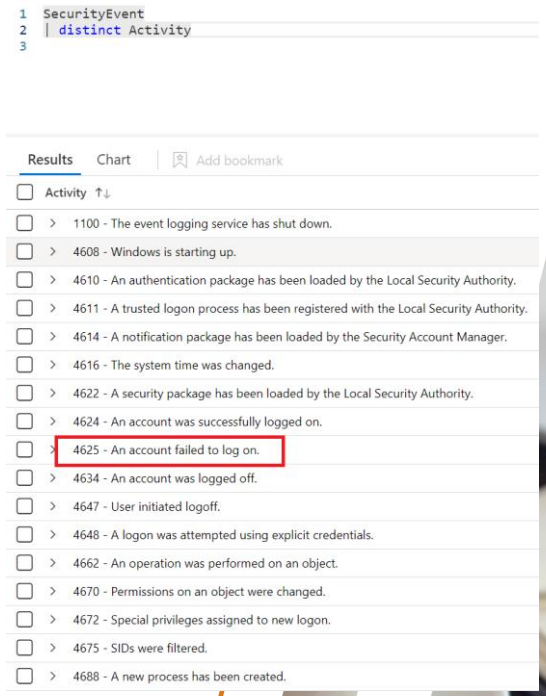
A: IP Address field is from where the failed logins occur. So, we look for IP addresses with multiple failed logins with different accounts. *

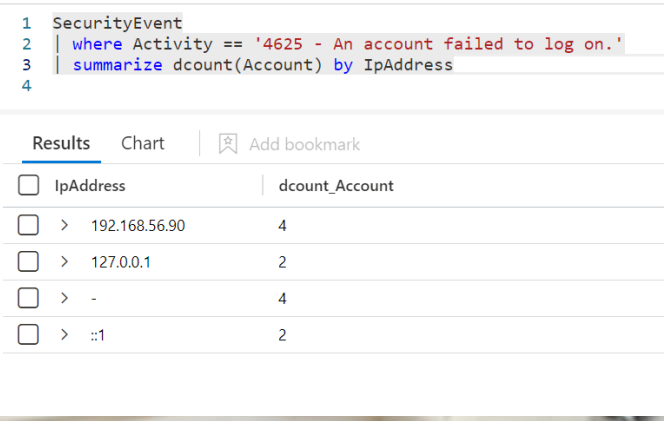
Q (BONUS) : Is it better to display aggregation results as data or graphs?

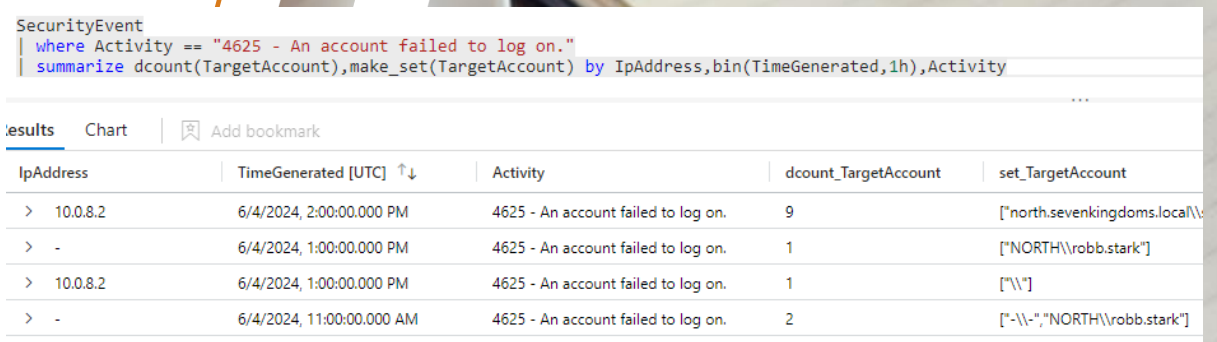
A: Graphs

*Domain authentication logging:

Some protocols like NTLM require extra configuration to provide the Source IP of the failed authentication







WORKSHOP

Examples - Server Software Component: Web Shell

- Server Software Component: Web Shell
  - A Web shell is a Web script that **is placed** on an openly accessible Web server to allow an adversary to access the Web server as a gateway into a network. A Web shell **may provide a set of functions to execute** or **a command-line interface** on the system that hosts the Web server.
  - **File monitoring** may be used to detect changes to files in the **Web directory of a Web server** that do not match with updates to the Web server's content and may indicate implantation of a Web shell script.

- Steps to follow to create the logic behind the detection - File Creation

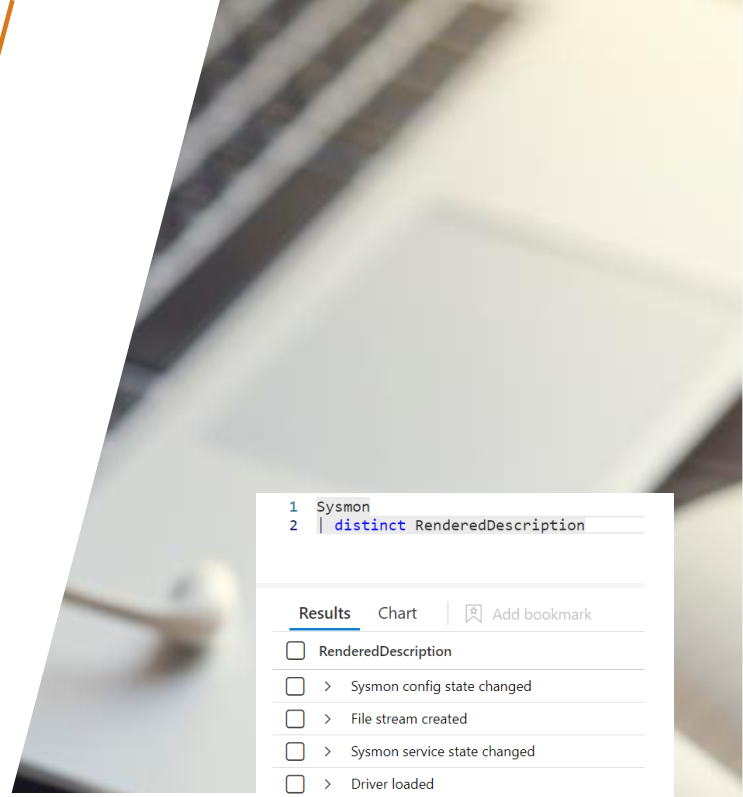
  Q: What data type is going to provide us this information?
  A: Sysmon

  Q: What type of event we want from the Sysmon data table?
  A: File created

  Q: Is the detection event-based or volume-based?
  A: Event-based

  Q: What column(s) are important so we can use them as filters?
  A: Rendered Description, file_name
  The file_name column displays the file that was created with the full path.

  Q: What filter I will put in the important columns:
  A: 1. Rendered Description equals 'File created'
  2. file_name starts with {Web Server Directory}
  3. file_name ends with {Webshell file extension}
  (Tip: Common Webshell file extensions: php, asp, aspx, cfm, jsp)



```
1  Sysmon
2  | distinct RenderedDescription
```

Results | Chart | Add bookmark

RenderedDescription
Sysmon config state changed
File stream created
Sysmon service state changed
Driver loaded
CreateRemoteThread detected
Pipe Created
Registry object added or deleted
Registry value set
Dns query
Pipe Connected
Process Create
File created
Image loaded
Network connection detected
Process accessed

```
9  Sysmon
10 | where RenderedDescription == "File created"
11 | where file_name startswith "C:\\xampp\\"
12 | where file_name endswith ".asp" or file_name endswith ".aspx" or file_name endswith ".cfm" or file_name endswith ".jsp" or file_name endswith ".php"
```

Results | Chart | Add bookmark

TimeGenerated [UTC]	Source	EventID	Computer	UserName	RenderedDescription
6/4/2024, 10:50:18.369 PM	Microsoft-Windows-Sysmon	11	castelblack.north.sevenkingdo...	NT AUTHORITY\SYSTEM	File created

TimeGenerated [UTC]	2024-06-04T22:50:18.3691025Z
Source	Microsoft-Windows-Sysmon
EventID	11
Computer	castelblack.north.sevenkingdoms.local
UserName	NT AUTHORITY\SYSTEM
RenderedDescription	File created
event_creation_time	2024-06-04T22:50:18.3660000Z
process_guid	{35604ab1-1752-665f-5000-000000009f00}
process_id	3404
process_path	C:\xampp\tomcat\bin\tomcat8.exe
file_name	C:\xampp\tomcat\webapps\cmd\cmd.jsp
file_creation_time	2024-06-04T22:50:18.3660000Z

*KQL syntax tip: Replace \ with \\ because the backslash is used to escape characters

WORKSHOP

Examples - Server Software Component: Web Shell

- Server Software Component: Web Shell
  - A Web shell is a Web script that **is placed** on an openly accessible Web server to allow an adversary to access the Web server as a gateway into a network. A Web shell **may provide a set of functions to execute** or **a command-line interface** on the system that hosts the Web server.
  - **Process monitoring** may be used to detect Web servers that perform suspicious actions such as spawning cmd.exe or accessing files that are not in the Web directory.
  - A web shell is a web script placed on an openly accessible web server to allow an adversary to use the server as a gateway in a network. As the shell operates, **commands will be issued from within the web application into the broader server operating system**.

- Steps to follow to create the logic behind the detection - Process Creation

Q: What data type is going to provide us this information?
A: Sysmon

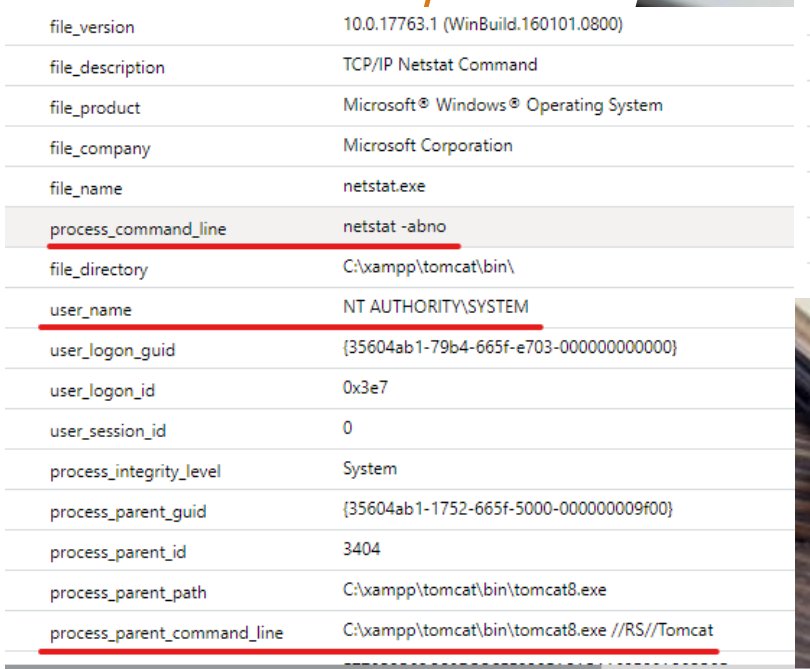
Q: What type of event we want from the Sysmon data table?
A: Process Create

Q: Is the detection event-based or volume-based?
A: Event-based

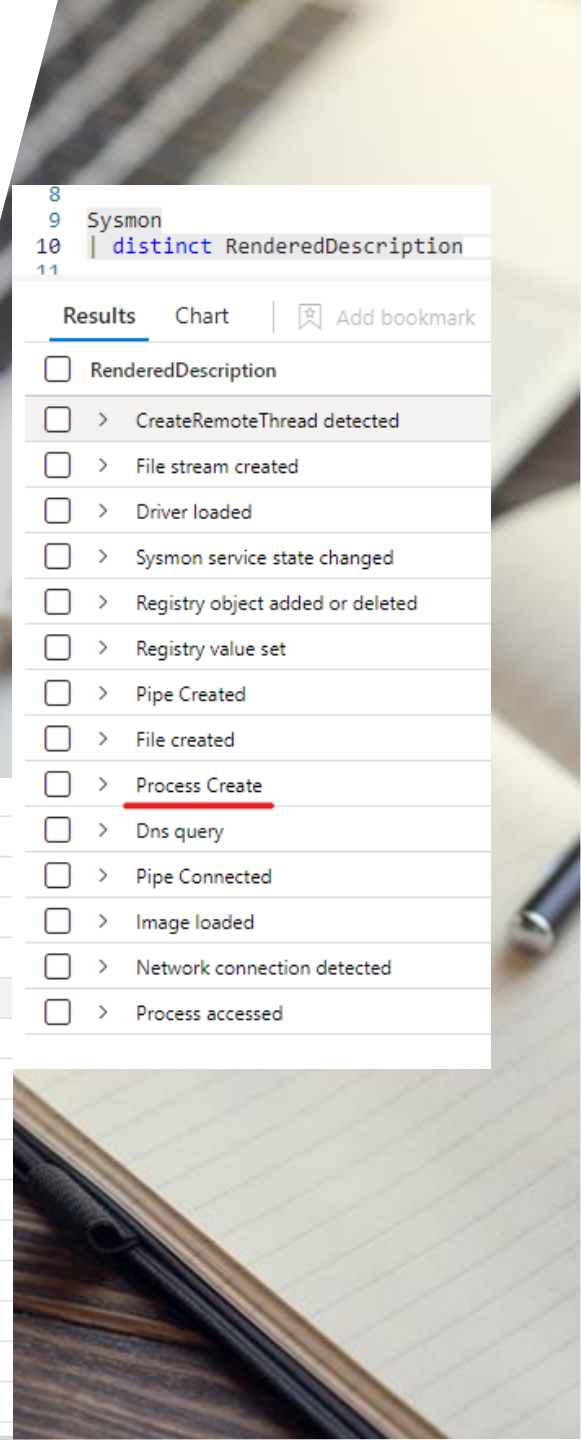
Q: What column(s) are important so we can use them as filters?
A: Rendered Description, process_parent_command_line, process_command_line
Command-line processes that are executed by web server processes
is a high indicator of Web Shell execution.

Q: What filter I will put in the important columns:
A: 1. Rendered Description equals 'Process Create'
2. process_parent_command_line contains common Web Server processes
3. process_command_line contains common command-line processes
(Tip: Common Web Server processes: w3wp.exe, httpd.exe, tomcat.exe, nginx.exe)
(Tip2: Common command-line processes: cmd, powershell, netstat, systeminfo, ipconfig, whoami etc.)

*KQL syntax tip: Replace \ with \\ because the backslash is used to escape characters



file_version	10.0.17763.1 (WinBuild.160101.0800)
file_description	TCP/IP Netstat Command
file_product	Microsoft® Windows® Operating System
file_company	Microsoft Corporation
file_name	netstat.exe
process_command_line	netstat -abno
file_directory	C:\xampp\tomcat\bin\
user_name	NT AUTHORITY\SYSTEM
user_logon_guid	{35604ab1-79b4-665f-e703-000000000000}
user_logon_id	0x3e7
user_session_id	0
process_integrity_level	System
process_parent_guid	{35604ab1-1752-665f-5000-000000009f00}
process_parent_id	3404
process_parent_path	C:\xampp\tomcat\bin\tomcat8.exe
process_parent_command_line	C:\xampp\tomcat\bin\tomcat8.exe //RS//Tomcat



WORKSHOP

Examples - Server Software Component: Web Shell

- Server Software Component: Web Shell
  - A Web shell is a Web script that **is placed** on an openly accessible Web server to allow an adversary to access the Web server as a gateway into a network. A Web shell **may provide a set of functions to execute** or **a command-line interface** on the system that hosts the Web server.
  - Monitor and analyze **traffic patterns and packet inspection** associated to protocol(s) that do not follow the expected protocol standards and traffic flows (e.g extraneous packets that do not belong to established flows, gratuitous or anomalous traffic patterns, anomalous syntax, or structure).

- Steps to follow to create the logic behind the detection - Network Traffic Content

  Q: What data type is going to provide us this information?
  A: Suricata

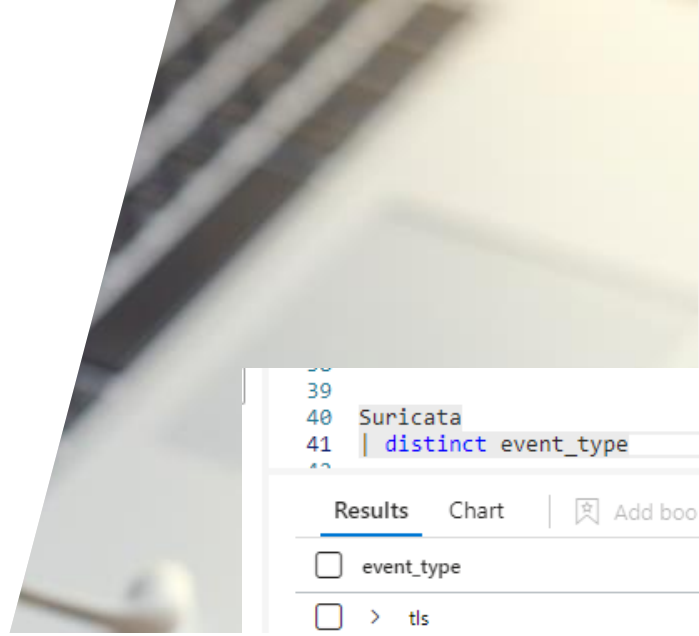
  Q: What type of event we want from the Sysmon data table?
  A: event_type > http

  Q: Is the detection event-based or volume-based?
  A: Event-based

  Q: What nested field is important inside the http item?
  A: url
  Some Web shells work by receiving the suspicious commands as parameters in the URL

  Q: What filter I will put in the important columns:
  A: 1. event_type equals 'http'
  2. url contains any of the suspicious commands

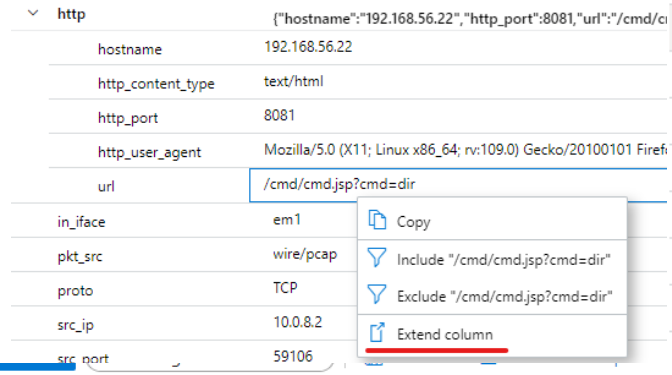
  Q: Did Suricata produce any alert related to this activity?
  A: Suricata | where event_type == 'alert' and app_proto == 'http'



```
39
40 Suricata
41 | distinct event_type
```

Results | Chart | Add boo

event_type
tls
smb
dns
krb5
fileinfo
http
alert
rdp




```
1 Suricata
2 | where event_type == 'http'
3 | extend url_ = tostring(http.url)
4 | where url_ has_any ('systeminfo','whoami','netstat','hostname')
```

Results | Chart | Add bookmark

TimeGenerated [UTC]	url_	Computer	RawData	Type
6/4/2024, 10:58:23.000 PM	/cmd/cmd.jsp?cmd=netstat+-abno	goad-VirtualBox	{"timestamp":"2024-06-04T22:5...	Suricata_CL
6/4/2024, 10:53:46.000 PM	/cmd/cmd.jsp?cmd=systeminfo	goad-VirtualBox	{"timestamp":"2024-06-04T22:5...	Suricata_CL
6/4/2024, 10:53:27.000 PM	/cmd/cmd.jsp?cmd=hostname	goad-VirtualBox	{"timestamp":"2024-06-04T22:5...	Suricata_CL
6/4/2024, 10:53:03.000 PM	/cmd/cmd.jsp?cmd=whoami+%2Fall	goad-VirtualBox	{"timestamp":"2024-06-04T22:5...	Suricata_CL
6/4/2024, 10:52:41.000 PM	/cmd/cmd.jsp?cmd=whoami+%2Fall	goad-VirtualBox	{"timestamp":"2024-06-04T22:5...	Suricata_CL

WORKSHOP

Examples - Server Software Component: Web Shell

- Steal or Forge Kerberos Tickets: AS-REP Roasting

  Adversaries may reveal credentials of accounts that have disabled **Kerberos preauthentication** by Password Cracking Kerberos messages. For each account found without preauthentication, an adversary may send an AS-REQ message without the encrypted timestamp and **receive an AS-REP message with TGT data which may be encrypted with an insecure algorithm such as RC4**. The recovered encrypted data may be vulnerable to offline Password Cracking attacks similarly to Kerberoasting and expose plaintext credentials.
  An account registered to a domain, with or without special privileges, can be abused to list all domain accounts that have preauthentication disabled by utilizing Windows tools like PowerShell with an LDAP filter**. Alternatively, the adversary may send an AS-REQ message for each user. If the DC responds without errors, the account does not require preauthentication and the AS-REP message will already contain the encrypted data.**

- Steps to follow to create the logic behind the detection

  Q: What data type is going to provide us this information?
  A: Suricata

  Q: What type of event we want from the Sysmon data table?
  A: event_type > krb5

  Q: Is the detection event-based or volume-based?
  A: Event-based

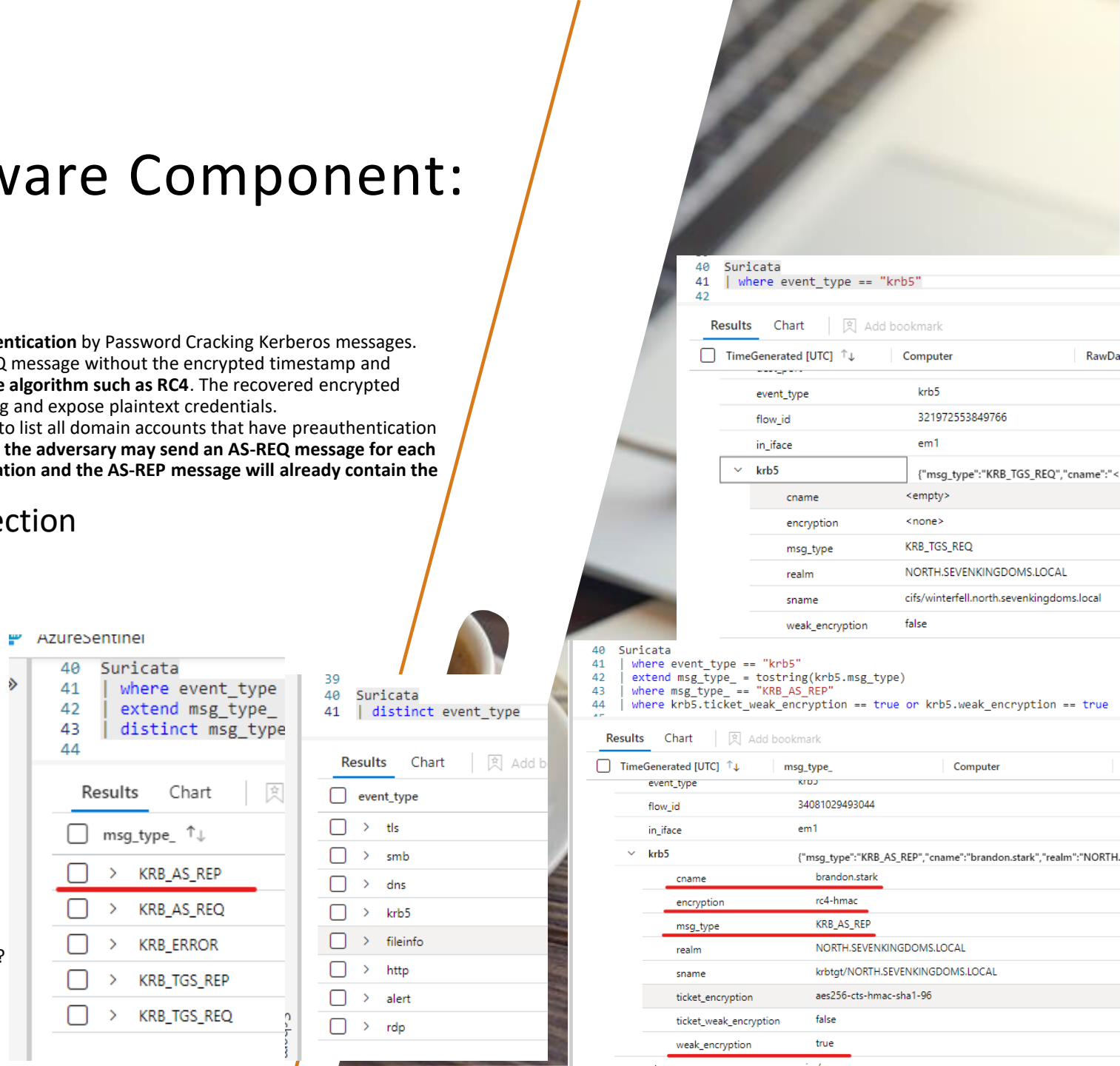
  Q: What nested field is important inside the krb5 item?
  A: msg_type, weak_encryption, ticket_weak_encryption

  Q: What column(s) are important so we can use them as filters?
  A: event_type, msg_type_ , weak_encryption_, ticket_weak_encryption_

  Q: What filter I will put in the important columns:
  A: 1. event_type = krb5
  2. ticket_weak_encryption_ = true or weak_encryption_ = true

  Q: What was the account that it is exposed and vulnerable to AS-REP Roasting attack?
  A: cname nested column > brandon.stark

  Q: What is the column that displays the origin of the attack?
  A: src_ip > 10.0.8.2



Thank you for your attention

Presentation by:

Christos Lazaridis
(Focal Point)

CyberSecPro