MITRE ATT&CK-T1003/006,, ATT&CK-T1207, ATT&CK-TA0006,

**SUMMARY:**

Crowd Strike was able to detect Mimikatz via the use of the Mimikatz File Hash reputation against Crowd Strikes IOC reputation database,  but was unable to detect the TTPs associated with Mimikatz performing DCSync techniques.

It is possible to detect a DCSync attack by monitoring network traffic to every domain controller, or by analyzing Windows event logs.

* Network monitoring:  
  Monitor network traffic for DRSUAPI RPC requests for the operation DsGetNCChanges and compare the source host against a list of domain controllers. If the source host does not appear on that list, then a DCSync attack is suspected. However, without significant investment in packet analysis it is not possible to determine which objects were replicated.
* Windows event logs  
  Event ID 4662 in the subcategory Audit Directory Service Access audits basic information about users performing operations within Active Directory for events specified in an object’s system access-control list (SACL). Using this event, it is possible to see when a user exercises their Replicating Directory Changes All extended right by filtering the properties field to include {1131f6ad-9c07-11d1-f79f-00c04fc2dcd2} which is the control access rights GUID for replicating directory changes.

**GOAL**

Based on a  threat Intel report, the CTI team reached out to the DELL Threat Hunt team to perform a validation hunt to determine if a DCSync attack is executed in DELL.COM domain controllers,  will threat detection capabilities and tools in the current security stack be able to detect the attack and every TTP that is associated with the attack steps and tools used.

Modern hacking tools are able to abuse the DRS Remote Protocol used by DCs for synchronization and replication. Mimikatz includes a feature called DCSync, which effectively “impersonates” a DC and requests account password data from a real DC via DRS.

**CATEGORIZATION:**

Technique:  
MITRE ATT&CK T1003/006 - OS Credential Dumping: DCSync  
MITRE ATT&CK TA0006 - Credential Access

**TECHNICAL DESCRIPTION:**

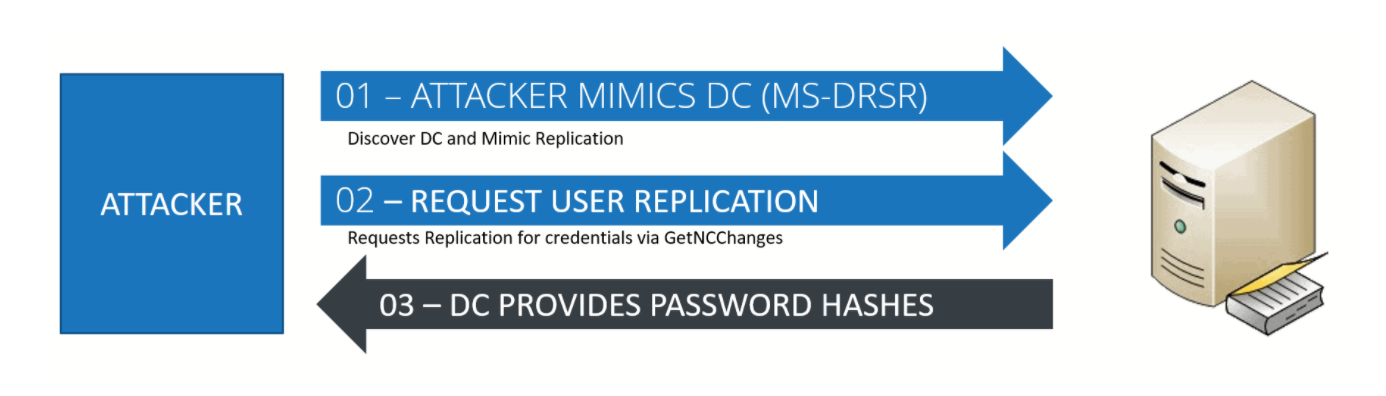
Logs required to detect DCSync:

* WinEvent ID 4662 (an operation was performed on an object)
* AD Event ID 2108 and AD Event ID 1084 (occur during inbound replication of Active Directory Domain Services)

Members of the Administrators, Domain Admins, and Enterprise Admin groups or computer accounts on the domain controller are able to run DCSync to pull password data from Active Directory, which may include current and historical hashes of potentially useful accounts such as KRBTGT and Administrators. The hashes can then in turn be used to create a Golden Ticket for use in Pass the Ticket or change an account's password as noted in Account Manipulation.

* Adversary tool of preference :
  + Mimikatz
  + PowerShell Empire
  + Metasploit

Methodology and steps on how to execute DCSYNC attack:



* Adversary tool of preference :
  + Mimikatz
  + PowerShell Empire
  + Metasploit

Pre attack DC Forest enumeration:

* First the adversary can enumerate the current trusts in the AD Domain controller forest in a few different ways- either by using PowerView 2.0 and run the following commands
  + Get-NetForestDomain
  + Invoke-MapDomainTrust -LDAP
* The initial foothold must be against a domain account with domain replication privileges; the Directory Replication Service Remote Protocol (MS-DRSR); MS-DRSR is a legitimate Active Directory service that cannot be disabled.
  + By default, these privileges are limited to the:
    - domain administrators
    - enterprise administrators
    - administrators
    - domain controller groups.
  + However, in certain cases, ordinary domain owners may have the needed permissions to launch a DCSync attack.Those roles have replication permissions that include the following rights that enable a DCSync attack:
    - Replicating Directory Changes
    - Replicating Directory Changes All
    - Replicating Directory Changes In Filtered Set
* Pre attack DC Forest enumeration:  
  First the adversary can enumerate the current trusts in the AD Domain controller forest in a few different ways- either by using PowerView 2.0 and run the following commands
  + Get-NetForestDomain
  + Invoke-MapDomainTrust -LDAP
* The attacker who manage to compromise an account with adequate permissions would load Mimikatz and run the DCSync command from the lsadump module, specifying the targeted domain and user account.
  + one example would be the KRBTGT account which is used to encrypt and sign Kerberos tickets within a domain.
  + A domain controller would use this account to decrypt and validate those tickets, authenticating accounts to access network resources.
* The DCSync command in Mimikatz allows an attacker to pretend to be a domain controller and retrieve password hashes from other domain controllers, without executing any code on the target. It does so over the MS-DRSR protocol via the DSGetNCChanges method that replicates updates from a naming context (NC) replica on the server.
* In addition to the crucial NTLM password hash, earlier password hashes are also returned. An attacker who can compromise those, as well, may deduce patterns a particular user employs to set passwords and may be able to crack those offline.
* Those passwords may also be returned in cleartext using the Powersploit tool, Microsoft PowerShell scripts that enable reverse encryption and allows an attacker access to the plaintext version of the secret.

**VALIDATION METHODOLOGY:**  
The first step is to acquire Mimikatz, powersploit, powerview and powershell-empire and load it in the SimSpace lab environment.

**Mimikatz:**

 Mimikatz has a feature (dcsync) which utilizes the Directory Replication Service (DRS) to retrieve the password hashes from the NTDS.DIT file. This technique eliminates the need to authenticate directly with the domain controller as it can be executed from any system that is part of the domain from the context of domain administrator. Therefore, it is the standard modus operandi of attackers as it is less noisy.

* + The following Mimikatz command will dump password hashes to a csv for adversary extraction.  
    lsadump::dcsync /domain:dell.com /all /csv
  + By specifying the domain username with the /user parameter Mimikatz can dump all the account information of this particular user including his password hash.  
    lsadump::dcsync /domain:dell.com /user:admin\_user
  + Alternatively executing Mimikatz directly in a compromised domain controller, password hashes can be dumped via the lsass.exe process.  
    privilege::debug  
    lsadump::lsa /inject
  + The following information can or will be retrieved after a successful Mimikatz DCSync attack
    - SAM Account name
    - Account Type
    - User Account Control options
    - Account Expiration
    - Password last set date
    - Security ID (SID)
    - RID
    - Current and previous (password history) NTLM password hashes which shows the password history, at least in NTLM hash format.
    - Current and previous LM password hashes which shows the password history in LM hash format.
    - Clear text password (requires reversible encryption)

**PowerShell Empire:**

PowerShell Empire has two modules which can retrieve domain hashes via the DCSync attack. Both modules need to be executed from the perspective of domain administrator and they are using Microsoft replication services.  
These modules rely on the Invoke-Mimikatz PowerShell script in order to execute Mimikatz commands related to DCSync.

The following module will extract the domain hashes to a format similar to the output of Metasploit hashdump command.  
           usemodule credentials/mimikatz/dcsync\_hashdump

* The DCSync module requires a user to be specified in order to extract all the account information.  
  usemodule credentials/mimikatz/dcsync  
  set user username  
  execute

**Invoke-dcsync:**

* The [Invoke–DCSync](https://gist.github.com/monoxgas/9d238accd969550136db)is a PowerShell script that leverages PowerView, Invoke-ReflectivePEInjection and a DLL wrapper of PowerKatz to retrieve hashes with the Mimikatz method of DCSync. Executing directly the function will generate the following output:  
  The results will be formatted into four tables: Domain, User, RID and Hash.  
  Command: Invoke-DCSync – PowerShell
* Invoke-DCSync with the parameter -PWDumpFormat will retrieve the hashes in the format: user:id:lm:ntlm:::  
  Command: Invoke-DCSync – PowerShell PWDump Format

**OUTCOME:**

Upon detonating Mimikatz commands in the SimSpace Lab, the following tools were able to detect DCSync related TTPs.

* Crowdstrike:
  + was able to detect the presence of Mimikatz on the host via the exploit's file hash
  + DCSync event detection log was generated via an event alert CS logs event\_simpleName=DCSyncAttempted on the Domain Controller
* Splunk:
  + Currently there are no analytics to detect DCSync related TTPs The only other controls that we can leverage is destination via host telemetry but we currently do not have any content written to detect this type of attack.
  + Upon writing a sample detection analytic, we were able to detect the DCSync activity that the hunt team enumerated in the SimSpace lab environment.

index=wineventlog sourcetype="WinEventLog:Security" EventCode =4624 [ search index=wineventlog sourcetype="WinEventLog:Security" EventCode =4662 | eval GUID\_Access =case (GUID == "1131f6aa-9c07-11d1-f79f-00c04fc2dcd2", "DS-Replication-Get-Changes", GUID == "1131f6ad-9c07-11d1-f79f-00c04fc2dcd2", "DS-Replication-Get-Changes-All", GUID == "89e95b76-444d-4c62-991a-0facbeda640c", "DS-Replication-Get-Changes-In-Filtered-Set") | fields Logon\_ID] | mvexpand Account\_Name | search NOT Account\_Name IN ("$","-","system","svc","dwm")  process\_ID="0x10\*" member\_id="addell" Workstation\_Name="dc" src\_nt\_domain"addell" (GUID = "1131f6aa-9c07-11d1-f79f-00c04fc2dcd2" OR GUID = "1131f6ad-9c07-11d1-f79f-00c04fc2dcd2" OR GUID = "89e95b76-444d-4c62-991a-0facbeda640c") | table \_time Account\_Name Process\_Name GUID GUID\_Access Process\_ID Access\_Mask

* DCSync Splunk detection Analytic analysis thought process is as follows:
  + The detection based on event 4662 is implemented as a subsearch in the search context of events 4624.
  + In case a DC replication occurs, event 4662 - An operation was performed on an object is recorded.
  + In the second part of Properties field of this event, there is a tree of GUID values belonging to Access Control Entry (ACE) of the AD object, for which the operation was performed.
  + DC replication can be determined from the GUID value contained in the Properties field
    - The “DS-Replication-Get-Changes” extended right  
      CN: DS-Replication-Get-Changes  
      GUID: 1131f6aa-9c07-11d1-f79f-00c04fc2dcd2
    - The “Replicating Directory Changes All” extended right  
      CN: DS-Replication-Get-Changes-All  
      GUID: 1131f6ad-9c07-11d1-f79f-00c04fc2dcd2
    - The “Replicating Directory Changes In Filtered Set” extended right  
      CN: DS-Replication-Get-Changes-In-Filtered-Set  
      GUID: 89e95b76-444d-4c62-991a-0facbeda640c  
        
      GUID fields are missing in the Win Event logs being forwarded to Splunk so the work around was for me to create the field and try to detect the activity via the new field using the following command.  
      Splunk SPL: | eval GUID\_Access =case(GUID == "1131f6aa-9c07-11d1-f79f-00c04fc2dcd2", "DS-Replication-Get-Changes", GUID == "1131f6ad-9c07-11d1-f79f-00c04fc2dcd2", "DS-Replication-Get-Changes-All", GUID == "89e95b76-444d-4c62-991a-0facbeda640c", "DS-Replication-Get-Changes-In-Filtered-Set")
  + The event 4662 provides the name of an account, but no network information. Hence in order, to distinguish between legitimate and malicious activities, a further context of the related account is necessary by performing a subsearch using Win Event ID 4624 which provides us the logon Account Name.
  + The analytic extracts Logon\_ID field from the detected events and finds corresponding logon information – Account\_Name provided by events 4624.
  + Correlation of these events based on the identifier Logon ID is possible because this identifier remains unique between reboots.
  + Next I filtered IP addresses of DCs are excluded from the search using inputlookup provided by file domain\_controllers.csv