Threat Hunter Playbook

Search this book...

KNOWLEDGE LIBRARY

Windows

PRE-HUNT ACTIVITIES

Data Management

GUIDED HUNTS

Windows

LSASS Memory Read Access

DLL Process Injection via CreateRemoteThread and LoadLibrary

Active Directory Object Access via Replication Services

Active Directory Root Domain Modification for Replication Services

Registry Modification to Enable Remote Desktop Conections

Local PowerShell Execution

WDigest Downgrade

PowerShell Remote Session

Alternate PowerShell Hosts

Domain DPAPI Backup Key Extraction

SysKey Registry Keys Access

SAM Registry Hive Handle Request

WMI Win32_Process Class and Create Method for Remote Execution

WMI Eventing

WMI Module Load

Local Service Installation

Remote Service creation

Remote Service Control Manager Handle

Remote Interactive Task Manager LSASS Dump

Registry Modification for Extended NetNTLM Downgrade

Access to Microphone Device

Remote WMI ActiveScriptEventConsumers

Remote DCOM IErtUtil DLL Hijack

Remote WMI Wbemcomn DLL Hijack

SMB Create Remote File

WuauClt CreateRemoteThread Execution

TUTORIALS

Jupyter Notebooks

Powered by Jupyter Book

Active Directory Object Access via Replication Services

Hypothesis

Adversaries might attempt to pull the NTLM hash of a user via active directory replication apis from a non-domain-controller account with permissions to do so.

Technical Context

Active Directory replication is the process by which the changes that originate on one domain controller are automatically transferred to other domain controllers that store the same data. Active Directory data takes the form of objects that have properties, or attributes. Each object is an instance of an object class, and object classes and their respective attributes are defined in the Active Directory schema. The values of the attributes define the object, and a change to a value of an attribute must be transferred from the domain controller on which it occurs to every other domain controller that stores a replica of that object.

Offensive Tradecraft

An adversary can abuse this model and request information about a specific account via the replication request. This is done from an account with sufficient permissions (usually domain admin level) to perform that request. Usually the accounts performing replication operations in a domain are computer accounts (i.e dcaccount$). Therefore, it might be abnormal to see other non-dc-accounts doing it.

The following access rights / permissions are needed for the replication request according to the domain functional level

Control access right symbol	Identifying GUID used in ACE
DS-Replication-Get-Changes	1131f6aa-9c07-11d1-f79f-00c04fc2dcd2
DS-Replication-Get-Changes-All	1131f6ad-9c07-11d1-f79f-00c04fc2dcd2
DS-Replication-Get-Changes-In-Filtered-Set	89e95b76-444d-4c62-991a-0facbeda640c

Additional reading

- https://github.com/OTRF/ThreatHunter-Playbook/tree/master/docs/library/windows/active_directory_replication.md

Pre-Recorded Security Datasets

Metadata	Value
docs	https://securitydatasets.com/notebooks/atomic/windows/credential_access/SDWIN-190301174830.html
link	https://raw.githubusercontent.com/OTRF/Security-Datasets/master/datasets/atomic/windows/credential_access/host/empire_dcsync_dcerpc_drsuapi_DsGetNCChanges.zip

Download Dataset

```
import requests
from zipfile import ZipFile
from io import BytesIO

url = 'https://raw.githubusercontent.com/OTRF/Security-Datasets/master/datasets/at
zipFileRequest = requests.get(url)
zipFile = ZipFile(BytesIO(zipFileRequest.content))
datasetJSONPath = zipFile.extract(zipFile.namelist()[0])
```

Read Dataset

```
import pandas as pd
from pandas.io import json

df = json.read_json(path_or_buf=datasetJSONPath, lines=True)
```

Analytics

A few initial ideas to explore your data and validate your detection logic:

Analytic I

Monitoring for non-dc machine accounts accessing active directory objects on domain controllers with replication rights might be suspicious.

Data source	Event Provider	Relationship	Event
Windows active directory	Microsoft-Windows-Security-Auditing	User accessed AD Object	4662

Logic

```
SELECT `@timestamp`, Hostname, SubjectUserName, SubjectLogonId
FROM dataTable
WHERE LOWER(Channel) = "security"
    AND EventID = 4662
    AND AccessMask = "0x100"
    AND (
        Properties LIKE "%1131f6aa_9c07_11d1_f79f_00c04fc2dcd2%"
        OR Properties LIKE "%1131f6ad_9c07_11d1_f79f_00c04fc2dcd2%"
        OR Properties LIKE "%89e95b76_444d_4c62_991a_0facbeda640c%"
    )
    AND NOT SubjectUserName LIKE "%$"
```

Pandas Query

```
(
df[['@timestamp','Hostname','SubjectUserName','SubjectLogonId']]

[(df['Channel'].str.lower() == 'security')
    & (df['EventID'] == 4662)
    & (df['AccessMask'] == '0x100')
    & (
        (df['Properties'].str.contains('.*1131f6aa-9c07-11d1-f79f-00c04fc2dcd2.*',
        | (df['Properties'].str.contains('.*1131f6ad-9c07-11d1-f79f-00c04fc2dcd2.*
        | (df['Properties'].str.contains('.*89e95b76-444d-4c62-991a-0facbeda640c.*
    )
    & (~df['SubjectUserName'].str.endswith('.*$', na=False))
]
)
```

Analytic II

You can use successful authentication events on the domain controller to get information about the source of the AD Replication Service request.

Data source	Event Provider	Relationship	Event
Authentication log	Microsoft-Windows-Security-Auditing	User authenticated Host	4624
Windows active directory	Microsoft-Windows-Security-Auditing	User accessed AD Object	4662

Logic

```
SELECT o.`@timestamp`, o.Hostname, o.SubjectUserName, o.SubjectLogonId, a.IpAddress
FROM dataTable o
INNER JOIN (
    SELECT Hostname,TargetUserName,TargetLogonId,IpAddress
    FROM dataTable
    WHERE LOWER(Channel) = "security"
        AND EventID = 4624
        AND LogonType = 3
        AND NOT TargetUserName LIKE "%$"
    ) a
ON o.SubjectLogonId = a.TargetLogonId
WHERE LOWER(o.Channel) = "security"
    AND o.EventID = 4662
    AND o.AccessMask = "0x100"
    AND (
        o.Properties LIKE "%1131f6aa_9c07_11d1_f79f_00c04fc2dcd2%"
        OR o.Properties LIKE "%1131f6ad_9c07_11d1_f79f_00c04fc2dcd2%"
```

```
        OR o.Properties LIKE "%89e95b76_444d_4c62_991a_0facbeda640c%"
    )
    AND o.Hostname = a.Hostname
    AND NOT o.SubjectUserName LIKE "%$"
```

Pandas Query

```
adObjectAccessDf = (
df[['@timestamp','Hostname','SubjectUserName','SubjectLogonId']]

[(df['Channel'].str.lower() == 'security')
    & (df['EventID'] == 4662)
    & (df['AccessMask'] == '0x100')
    & (
        (df['Properties'].str.contains('.*1131f6aa-9c07-11d1-f79f-00c04fc2dcd2.*',
        | (df['Properties'].str.contains('.*1131f6ad-9c07-11d1-f79f-00c04fc2dcd2.*
        | (df['Properties'].str.contains('.*89e95b76-444d-4c62-991a-0facbeda640c.*
    )
    & (~df['SubjectUserName'].str.endswith('.*$', na=False))
]
)

networkLogonDf = (
df[['@timestamp','Hostname','TargetUserName','TargetLogonId','IpAddress']]

[(df['Channel'].str.lower() == 'security')
    & (df['EventID'] == 4624)
    & (df['LogonType'] == 3)
    & (~df['SubjectUserName'].str.endswith('.*$', na=False))
]
)

(
pd.merge(adObjectAccessDf, networkLogonDf,
    left_on = 'SubjectLogonId', right_on = 'TargetLogonId', how = 'inner')
)
```

Known Bypasses

Idea	Playbook
Adversaries could perform the replication request from a Domain Controller (DC) and with the DC machine account (*$) to make it look like usual/common replication activity.	

False Positives

Hunter Notes

- As stated before, when an adversary utilizes directory replication services to connect to a DC, a RPC Client call operation with the RPC Interface GUID of E3514235-4B06-11D1-AB04-00C04FC2DCD2 is performed pointing to the targeted DC. This activity is recorded at the source endpoint, from where the replication request is being performed from, via the Microsoft-Windows-RPC ETW provider. Even though the Microsoft-Windows-RPC ETW provider provides great visibility for replication operations at the source endpoint level, it does not have an event tracing session writing events to an .evtx file which does not make it practical to use it at scale yet.
- You can collect information from the Microsoft-Windows-RPC ETW provider and filter on RPC Interface GUID E3514235-4B06-11D1-AB04-00C04FC2DCD2 with SilkETW. One example here > https://twitter.com/FuzzySec/status/1127249052175872000
- You can correlate security events 4662 and 4624 (Logon Type 3) by their Logon ID on the Domain Controller (DC) that received the replication request. This will tell you where the AD replication request came from. This will also allow you to know if it came from another DC or not.

Hunt Output

Type	Link
Sigma Rule	https://github.com/SigmaHQ/sigma/blob/master/rules/windows/builtin/security/win_ad_replication_non_machine_account.yml

References

- https://docs.microsoft.com/en-us/openspecs/windows_protocols/ms-adts/1522b774-6464-41a3-87a5-1e5633c3fbbb
- https://docs.microsoft.com/en-us/windows/desktop/adschema/c-domain
- https://docs.microsoft.com/en-us/windows/desktop/adschema/c-domaindns
- http://www.harmj0y.net/blog/redteaming/a-guide-to-attacking-domain-trusts/
- https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2003/cc782376(v=ws.10)
- https://docs.microsoft.com/en-us/openspecs/windows_protocols/ms-drsr/f977faaa-673e-4f66-b9bf-48c640241d47

Previous
DLL Process Injection via CreateRemoteThread and LoadLibrary

Next
Active Directory Root Domain Modification for Replication Services

By Roberto Rodriguez @Cyb3rWard0g