



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



Contents

- Hypothesis
- Technical Context
- Offensive Tradecraft
- Pre-Recorded Security Datasets
- Analytics
- Known Bypasses
- False Positives
- Hunter Notes
- Hunt Output
- References

SysKey Registry Keys Access

Hypothesis

Adversaries might be calculating the SysKey from registry key values to decrypt SAM entries.

Technical Context

Every computer that runs Windows has its own local domain; that is, it has an account database for accounts that are specific to that computer. Conceptually, this is an account database like any other with accounts, groups, SIDs, and so on. These are referred to as local accounts, local groups, and so on. Because computers typically do not trust each other for account information, these identities stay local to the computer on which they were created.

Offensive Tradecraft

Adversaries might use tools like Mimikatz with lsadump::sam commands or scripts such as Invoke-PowerDump to get the SysKey to decrypt Security Account Mannager (SAM) database entries (from registry or hive) and get NTLM, and sometimes LM hashes of local accounts passwords. Adversaries can calculate the Syskey by using RegOpenKeyEx/RegQueryInfoKey API calls to query the appropriate class info and values from the HKLM:\SYSTEM\CurrentControlSet\Control\Lsa\JD, HKLM:\SYSTEM\CurrentControlSet\Control\Lsa\Skew1, HKLM:\SYSTEM\CurrentControlSet\Control\Lsa\GBG, and HKLM:\SYSTEM\CurrentControlSet\Control\Lsa\Data keys.

Additional reading

- https://github.com/OTRF/ThreatHunter-Playbook/tree/master/docs/library/windows/security_account_manager_database.md
- <https://github.com/OTRF/ThreatHunter-Playbook/tree/master/docs/library/windows/syskey.md>

Pre-Recorded Security Datasets

Metadata	Value
docs	https://securitydatasets.com/notebooks/atomic/windows/credential_access/SDWIN-190625103712.html
link	https://raw.githubusercontent.com/OTRF/Security-Datasets/master/datasets/atomic/windows/credential_access/host/empire_mimikatz_sam_access.zip

Download Dataset

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

url = 'https://raw.githubusercontent.com/OTRF/Security-Datasets/master/datasets/atomic/windows/credential_access/host/empire_mimikatz_sam_access.zip'
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

Look for handle requests and access operations to specific registry keys used to calculate the SysKey. SACLs are needed for them.

Data source	Event Provider	Relationship	Event
Windows registry	Microsoft-Windows-Security-Auditing	Process accessed Windows registry key	4663
Windows registry	Microsoft-Windows-Security-Auditing	Process requested access Windows registry key	4656

Logic

```
SELECT `@timestamp`, ProcessName, ObjectName, AccessMask, EventID
FROM dataTable
WHERE LOWER(Channel) = "security"
      AND (EventID = 4656 OR EventID = 4663)
      AND ObjectType = "Key"
      AND (
        lower(ObjectName) LIKE "%jd"
        OR lower(ObjectName) LIKE "%gbg"
        OR lower(ObjectName) LIKE "%data"
        OR lower(ObjectName) LIKE "%skew1"
      )
```

Pandas Query

```
(
df[['@timestamp','Hostname','ProcessName','ObjectName','AccessMask','EventID']]

[(df['Channel'].str.lower() == 'security')
 & ((df['EventID'] == 4656)|(df['EventID'] == 4663))
 & (df['ObjectType'] == 'Key')
 & (
  (df['ObjectName'].str.lower().str.endswith('jd', na=False))
  | (df['ObjectName'].str.lower().str.endswith('gbg', na=False))
  | (df['ObjectName'].str.lower().str.endswith('data', na=False))
  | (df['ObjectName'].str.lower().str.endswith('skew1', na=False))
 )
]
)
```

Known Bypasses

Idea	Playbook
Apparently the registry keys needed to calculate the SysKey are accessed by processes such as smss.exe, winlogon.exe and syskey.exe, but when the system boots. An adversary can migrate to those processes to blend in.	

False Positives

Hunter Notes

- An audit rule needs to be added to the SACL of the following keys to monitor for ReadKey rights
 - HKLM:\SYSTEM\CurrentControlSet\Control\Lsa\JD

- HKLM:\SYSTEM\CurrentControlSet\Control\Lsa\Skew1
 - HKLM:\SYSTEM\CurrentControlSet\Control\Lsa\GBG
 - HKLM:\SYSTEM\CurrentControlSet\Control\Lsa\Data
- Defenders can correlate known processes accessing those registry keys with events that tell you when the system boots up.
- Look for the same process accessing all those registry keys in a short period of time.

Hunt Output

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

References

- <https://github.com/gentilkiwi/mimikatz/wiki/module-~-lsadump>
- https://adsecurity.org/?page_id=1821#LSADUMPSAM
- <http://www.harmj0y.net/blog/activedirectory/remote-hash-extraction-on-demand-via-host-security-descriptor-modification/>
- <https://docs.microsoft.com/en-us/dotnet/api/system.security.accesscontrol.registryrights?view=netframework-4.8>
- <https://docs.microsoft.com/en-us/windows/desktop/sysinfo/registry-key-security-and-access-rights>

[Previous](#)
[Domain DPAPI Backup Key Extraction](#)

[Next](#)
[SAM Registry Hive Handle Request](#)

By Roberto Rodriguez @Cyb3rWard0g
© Copyright 2022.