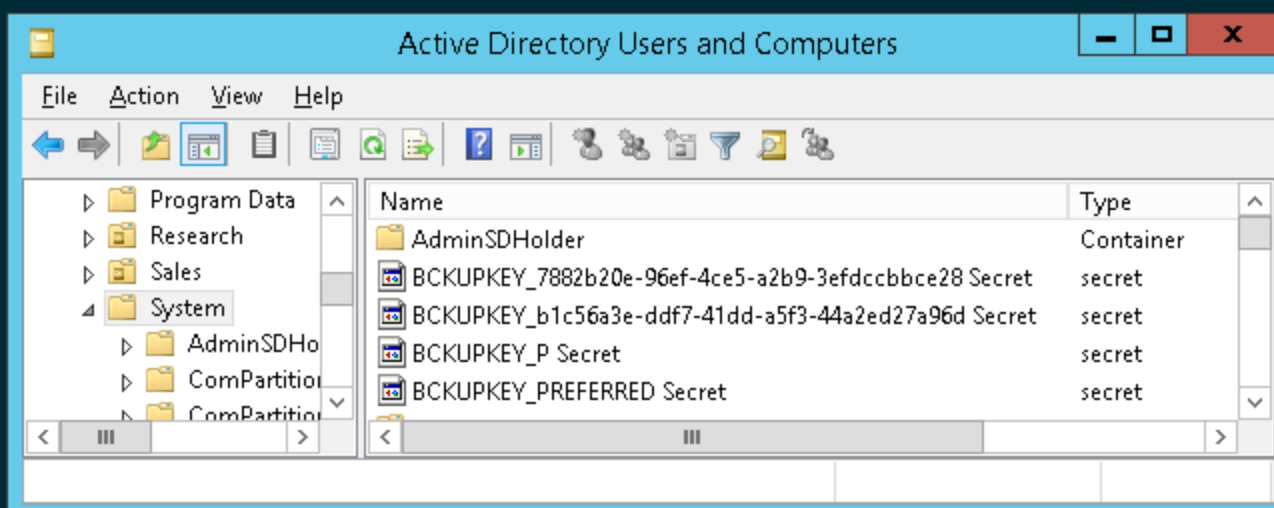
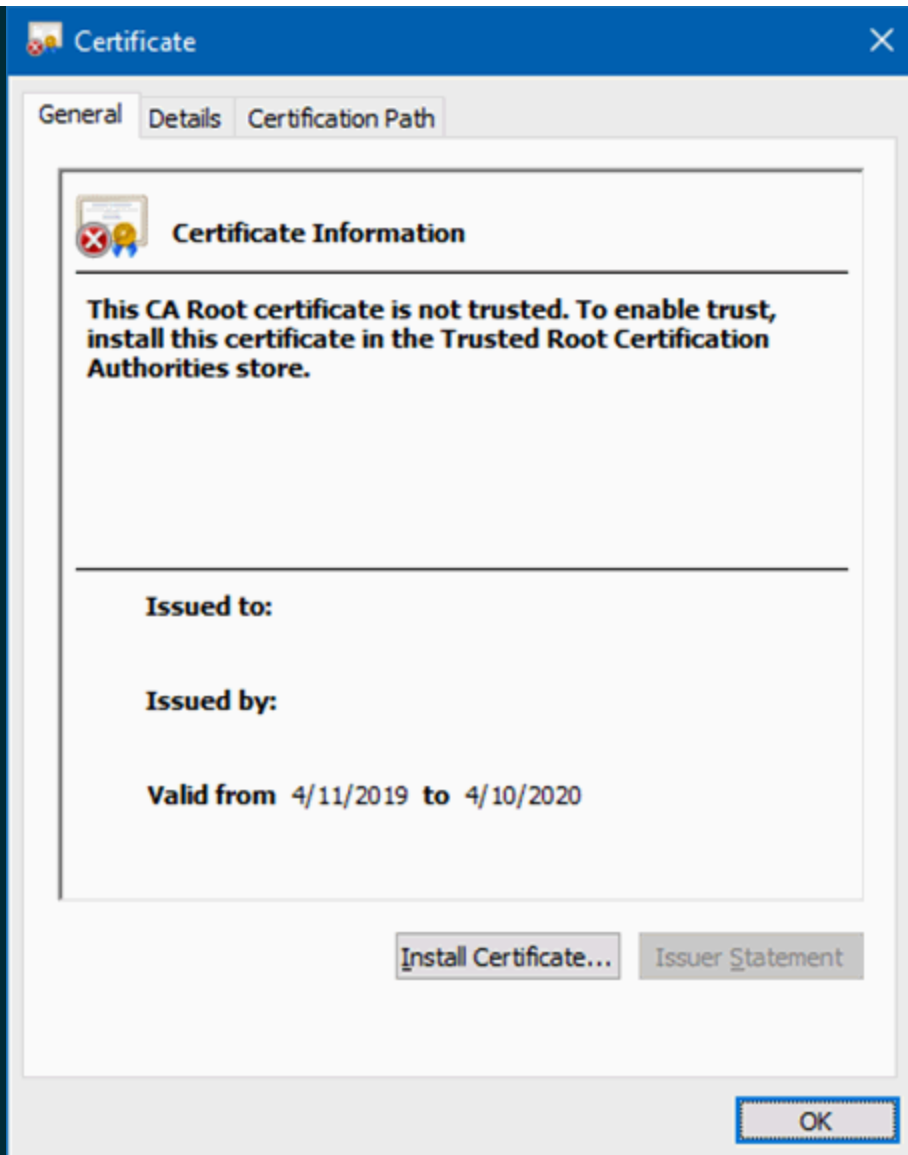


# Detecting DPAPI Backup Key Theft

 Jul 31, 2023  [Michael Grafnetter](#)

## Introduction

The [Data Protection API \(DPAPI\)](#) in Windows is used to encrypt passwords saved by browsers, certificate private keys, and other sensitive data. Domain controllers (DCs) [hold backup master keys](#) that can be used to decrypt all such secrets encrypted with DPAPI on domain-joined computers. These backup keys are stored as self-signed certificates in Active Directory (AD) objects of type `secret` called `BCKUPKEY_*`:



Attackers with sufficient permissions can fetch these backup keys from AD through the Local Security Authority (Domain Policy) Remote Protocol (MS-LSAD / LSARPC) and use them to decrypt any secrets protected by DPAPI

on all domain-joined Windows machines.

It is therefore important for organizations to be able to detect the theft of DPAPI backup keys from AD by malicious actors. This article describes various ways of discovering this attack technique.

## Attack Classification

|  |   |
|--|---|
| MITRE ATT&CK® Tactic                   | Credential Access (TA0006)  |
| MITRE ATT&CK® Technique                | Credentials from Password Stores (T1555)                            |
| MITRE ATT&CK® Sub-Technique            | Unsecured Credentials: Private Keys (T1552.004)                     |
| Tenable® Indicator of Attack           | DPAPI Domain Backup Key Extraction                                  |
| Microsoft® Defender for Identity Alert | Malicious request of Data Protection API master key (alert ID 2020) |

## Detection on Domain Controllers

The most reliable way of detecting this attack technique is to monitor domain controllers for suspicious operations.

### Domain Controller Security Event Logs

When a DPAPI backup key is retrieved from a domain controller (DC) through the MS-LSAD protocol, an [undocumented](#) event with the following properties is generated on that DC:

|               |                            |  |
|---------------|----------------------------|--|
| Log Name      | Security                   |  |
| Event ID      | 4662                       |  |
| Keywords      | Audit Success              |  |
| Task Category | Other Object Access Events |  |

|               |                              |  |
|---------------|------------------------------|--|
| Object Server | LSA                          |  |
| Object Type   | SecretObject                 |  |
| Accesses      | Query secret value           |  |
| Object Name   | Policy\Secrets\G\$BCKUPKEY_* |  |

Event Properties - Event 4662, Microsoft Windows security auditing.

GeneralDetails

An operation was performed on an object.

Subject:

Security ID: contoso\Admin  
Account Name: Admin  
Account Domain: contoso  
Logon ID: 0x1DF9DC

Object:

Object Server: LSA  
Object Type: SecretObject  
Object Name: Policy\Secrets\G\$BCKUPKEY\_8ce0b46b-edcd-4c93-86a0-10662dd3e305  
Handle ID: 0x2608cc477e0

Operation:

Operation Type: Query  
Accesses: Query secret value  
  
Access Mask: 0x2  
Properties: -

Additional Information:

Log Name: Security

Source: Microsoft Windows security

Event ID: 4662

Level: Information

User: N/A

OpCode: Info

Logged: 7/25/2023 4:11:19 PM

Task Category: Other Object Access Events

Keywords: Audit Success

Computer: DC.contoso.com

Auditing of `Success` events of type `Audit Other Object Access Events` from the `Object Access` category in `Advanced Audit Policy Configuration` must first be enabled on all DCs.

## Domain Controller Network Traffic

The misuse of the MS-LSAD / LSARPC protocol can also be detected through deep packet inspection of domain controller traffic:

|                      |                                      |
|----------------------|--------------------------------------|
| RPC protocol UUID    | 12345778-1234-ABCD-EF00-0123456789AB |
| RPC operation name   | LsarRetrievePrivateData              |
| RPC operation number | 43                                   |

Both `RPC/TCP` (TCP port 135 + ephemeral port) and `RPC/NP` (TCP port 445) bindings can be used by clients. In WireShark, the `lsarpc.opnum == 43` display filter can be used to identify this type of traffic:

\*HackerFest 0

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

lsarpc.opnum == 43

| No.  | Time       | Source          | Destination     | Protocol | Length | Info   |
|------|------------|-----------------|-----------------|----------|--------|--|
| 1569 | 971.612816 | PC1.contoso.com | dc.contoso.com  | LSARPC   | 370    | lsa_RetrievePrivateData request[Long frame (168 bytes)]                    |
| 1570 | 971.614462 | dc.contoso.com  | PC1.contoso.com | LSARPC   | 518    | lsa_RetrievePrivateData response, Unknown error 0x00020000, Error: Unknown |
| 1586 | 972.642599 | PC1.contoso.com | dc.contoso.com  | LSARPC   | 314    | lsa_RetrievePrivateData request[Long frame (112 bytes)]                    |

> Frame 1569: 370 bytes on wire (2960 bits), 370 bytes captured

> Ethernet II, Src: PC1.contoso.com (00:17:fb:00:00:07), Dst: dc

> Internet Protocol Version 4, Src: PC1.contoso.com (10.85.0.6),

> Transmission Control Protocol, Src Port: 52517, Dst Port: 445,

> NetBIOS Session Service

> SMB2 (Server Message Block Protocol version 2)

> Distributed Computing Environment / Remote Procedure Call (DCE)

> Local Security Authority, lsa\_RetrievePrivateData

>   Operation: lsa\_RetrievePrivateData (43)

>   [Response in frame: 1570]

>   Long frame

0000 00 17 fb 00 00 04 00 17 fb 00 00 07 08 00 45 00

0010 01 64 d4 71 40 00 80 06 10 70 0a 55 00 06 0a 55

0020 00 03 cd 25 01 bd 45 aa 44 ac 89 61 88 cf 50 18

0030 03 fe 8c 3e 00 00 00 00 01 38 fe 53 4d 42 40 00

0040 01 00 00 00 00 00 0b 00 01 00 38 00 00 00 00 00

0050 00 00 0e 00 00 00 00 00 00 00 ff fe 00 00 01 00

0060 00 00 71 00 00 00 00 60 00 00 c5 34 96 12 49 a4

0070 44 7a 27 43 28 f8 04 55 90 e9 39 00 00 00 17 c0

0080 11 00 83 00 00 00 18 00 00 00 01 00 00 00 18 00

0090 00 00 78 00 00 00 c0 00 00 00 00 00 00 00 78 00

00a0 00 00 00 00 00 00 e4 07 00 00 01 00 00 00 00 00

00b0 00 00 05 00 00 03 10 00 00 00 c0 00 00 00 06 00

00c0 00 00 a8 00 00 00 01 00 2b 00 00 00 00 00 e7 21

00d0 45 b5 61 cf 77 40 b1 b7 66 1f fd c8 28 c5 00 00

00e0 00 00 5e 00 5e 00 00 00 00 00 00 00 02 00 00 00

00f0 00 00 2f 00 00 00 00 00 00 00 00 00 00 00 00 00

0100 00 00 2f 00 00 00 00 00 00 00 47 00 24 00 42 00

0110 43 00 4b 00 55 00 50 00 4b 00 45 00 59 00 5f 00

This detection technique is most probably used by Microsoft Defender for Identity and the already discontinued Advanced Threat Analytics (ATA):

Malicious Data Protection Private Information Request

An unknown user performed 4 successful attempts from M01 to retrieve DPAPI domain backup key from 2 domain controllers.

Note Share Export to Excel Details

Open

M01

Private Information Request

2 domain controllers

(Un)Fortunately, some organizations are slowly deploying SMB3 encryption even on DCs, which breaks this detection method, when the `RPC/NP` binding is used. IPsec tunneling would additionally break the detection at the network level for the `RPC/TCP` binding, but IPsec is rarely used.

## Detection on Endpoints

EDR solutions could theoretically be used to detect when corporate endpoints are misused to retrieve DPAPI backup keys from remote domain controllers. Unfortunately, all detection techniques listed in this section can easily be bypassed by obfuscation.

### Malicious Commands

Execution the the following off-the-shelf hacktools should raise an alert:

- `mimikatz.exe` tool with the `lsadump::backupkeys` parameter.
- `SharpDPAPI.exe` tool with the `backupkey` parameter.
- `Get-LsaBackupKey` PowerShell cmdlet from the DSInternals module.

This detection technique [is used by Microsoft Defender for Endpoint](#), among others.

### Suspicious File Names

Both Mimikatz and DSInternals export stolen DPAPI backup keys into files with the following name format:

- `ntds_capi_*.pfx`
- `ntds_capi_*.pvk`

The presence of these files should thus be considered an indicator of compromise. This detection technique [is utilized by Elastic Security for endpoint](#), among others.

### Suspicious Win32 API Calls

All 3 hacktools mentioned in this chapter perform calls to the `LsaRetrievePrivateData` function from `advapi32.dll`, which can also be picked up by EDRs. This appears to be the most reliable detection method on endpoints, but it could still be bypassed by directly performing the respective RPC calls.

## Alternative Attack Techniques

Usage of the MS-LSAD protocol is [one of many ways](#) of extracting DPAPI backup keys from domain controllers. Other techniques include, but are not limited to:

- Fetching the keys through the directory replication protocol.
- Extracting the keys from `ntds.dit` database files.

The detection of these techniques is out-of-scope of this article.

## Additional Resources

- [Microsoft: DPAPI backup keys on Active Directory domain controllers](#)
- [DSInternals: Retrieving DPAPI Backup Keys from Active Directory](#)
- [CQURE: Extracting Roamed Private Keys from Active Directory](#)
- [SpecterOps: HomeOperational Guidance for Offensive User DPAPI Abuse](#)
- [Sygnia: The Downfall Of Dpapi Top Secret Weapon](#)

