

LOCKED ADMIN PANEL

AUTHOR:

CYBEXER TECHNOLOGIES

International Cyber Security
Challenge



SEPT 2021



1. DESCRIPTION

This time you must investigate a recent attack on your web server. Attackers have dropped some password-protected site there. Your friends from Incident Response Department managed to get network capture file from one of the attackers' computers.

Check out the file and see if you can figure out how to access the site.

The PCAP is here: `http://<target>/capture.pcapng` and the site is at `http://<target>:85/`

2. CHALLENGE SPECIFICATIONS

- Category: Forensics
- Difficulty: Medium
- Estimated time: 30-60 min

3. QUESTIONS AND ANSWERS

3.1 WHAT FLAG IS DISPLAYED UPON SUCCESSFUL LOGIN?

```
icsc{Adm1nP@g3F@g}
```

4. SETUP INSTRUCTIONS

Dockerfile and *docker-compose.yml* are provided to run the task in a container. Multiple parameters can be given through docker-compose environment, see **.env**:

```
$ cat .env
TARGETPORT=85
PCAPPOR=80
FLAG=icsc{Adm1nP@g3F@g}
ACCOUNT=icscadmin
PASSWORD=Onc3@g@1nW3n33d@P@ssw0$d
```



TARGETPORT is where the admin panel is listening. *PCAPPORT* is another HTTP service that is just serving the PCap file. *ACCOUNT* and *PASSWORD* are credentials to log in to the service.

NOTE: DEFAULT VALUES OF ACCOUNT AND PASSWORD MATCH TO KEYSTROKES IN PCAP. IF THESE ARE CHANGED, PCAP MUST BE ADJUSTED ACCORDINGLY!

Run the container with:

```
docker-compose up --build
```

FLAG, *ACCOUNT* and *PASSWORD* are inserted to container at build time, ports are mapped at start-up.

4.1 CREATING NEW PCAP FILE

The *capture.pcapng* is created using Wireshark and USBpcap. It can be recreated with few steps:

1. Install USBPcap and Wireshark
2. Unplug any unnecessary USB devices
3. Start Wireshark, capture from USB and network simultaneously
4. Connect to target with browser
5. Log in to the admin page with known credentials. To make task more fun, attempts with wrong password or correction of typos (arrows, backspaces, etc) can be inserted.
6. Inspect HTTP packets in the captured traffic and filter out packets that contain submission of the form
7. Save the filtered traffic

5. ARTIFACTS PROVIDED

File	SHA-256
locked-admin.tar.gz	4ca02d6a96b3f4a166834b16efc4bd1ad101128b43ccfdd000ad18b68f3e79bf

6. TOOLS NEEDED

- Wireshark
- Scripting language, e.g., Python with scapy module

7. WALKTHROUGH

Look at the target with browser. It is a login page:

Login

Username:

Password:

Look at the provided PCap:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	2.7.1	host	USB	35	URB_INTERRUPT in
2	0.000028	host	2.7.1	USB	27	URB_INTERRUPT in
3	0.005987	2.7.1	host	USB	35	URB_INTERRUPT in
4	0.006013	host	2.7.1	USB	27	URB_INTERRUPT in
5	0.016984	2.7.1	host	USB	35	URB_INTERRUPT in
6	0.017006	host	2.7.1	USB	27	URB_INTERRUPT in
7	0.033000	2.7.1	host	USB	35	URB_INTERRUPT in
8	0.033038	host	2.7.1	USB	27	URB_INTERRUPT in

It contains USB traffic. When this is filtered out, some HTTP can be noticed also:

No.	Time	Source	Destination	Protocol	Length	Info
613	0.653254	LCFChafe_d7...	LLDP_Multicast	LLDP	170	LA/laptop-n2cmihm7.LA(port:001.20.Syn-LAP
3592	5.952108	10.32.12.123	10.32.12.124	TCP	66	51570 → 80 [SYN] Seq=0 Win=64240 [TCP CHECKSUM=0] Seq=0 Win=0 Len=0
3593	5.952367	10.32.12.124	10.32.12.123	TCP	66	80 → 51570 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0
3594	5.952433	10.32.12.123	10.32.12.124	TCP	54	51570 → 80 [ACK] Seq=1 Ack=1 Win=2102272 Len=0
3595	5.952725	10.32.12.123	10.32.12.124	HTTP	583	GET / HTTP/1.1
3596	5.952998	10.32.12.124	10.32.12.123	TCP	54	80 → 51570 [ACK] Seq=1 Ack=530 Win=64128 Len=0
3597	5.953586	10.32.12.124	10.32.12.123	HTTP	1280	HTTP/1.1 200 OK (text/html)
3598	6.002187	10.32.12.123	10.32.12.124	TCP	54	51570 → 80 [ACK] Seq=530 Ack=1227 Win=2102272 Len=0
3599	6.023762	10.32.12.123	10.32.12.124	HTTP	541	GET /favicon.ico HTTP/1.1

HTTP traffic contains the same admin panel, but password is not found there. Moving back to USB traffic. We can see communication with two devices – 2.7.1 and 2.11.5:

No.	Time	Source	Destination	Protocol	Length	Info
4574	11.287002	2.11.5	host	USB	35	URB_INTERRUPT in
4576	11.342986	2.11.5	host	USB	35	URB_INTERRUPT in
4578	11.494983	2.11.5	host	USB	35	URB_INTERRUPT in
4580	11.550986	2.11.5	host	USB	35	URB_INTERRUPT in
4582	11.630986	2.11.5	host	USB	35	URB_INTERRUPT in
4584	11.678982	2.11.5	host	USB	35	URB_INTERRUPT in
4586	11.790987	2.11.5	host	USB	35	URB_INTERRUPT in
4588	11.878985	2.11.5	host	USB	35	URB_INTERRUPT in

Let's make a wild guess that either of these is a keyboard. Since the input from keyboard is not printable strings, but events with scan-codes, a script is necessary to process this data.

Let's take device 2.11.5 first as it has sent less packets.



```
for p in rdpcap('capture.pcapng'):  
    if not p.haslayer(usb.USBpcap) \  
        or p[usb.USBpcap].bus != 2 \  
        or p[usb.USBpcap].device != 11 \  
        or (p[usb.USBpcap].endpoint & 0x7f) != 5 \  
        or len(p[usb.USBpcap].payload) == 0:  
        continue
```

Packet from USB HID is 6 bytes:

- Modifier
- Reserved
- up to 4 keys that can be simultaneously pressed

The parsing can be easily done with dictionaries in Python:

```
modifiers = { 0x80:"<LeftCtrl>", 0x40:"<AltGr>", ... }  
chars = { 0x04:"a", 0x05:"b", ... }  
payload = bytes(p[usb.USBpcap].payload)  
if load[2]:  
    if load[0]:  
        flag.extend([modifiers[m] for m in modifiers if load[0] & m])  
    flag.append(chars[load[2]])
```

Running such script gives output like this:

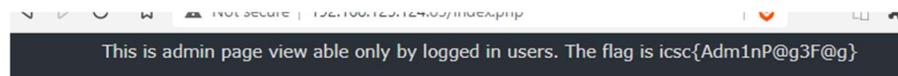
```
$ python3 solver.py  
admin<TAB>pass<BS><BS>ssword<RET>icscadmin<TAB>0nc3<HOME><DEL><RightShift>o<END><AltGr>2g<AltGr>2i  
n<LEFT><BS>1<RIGHT><RightShift>w3need<LEFT><BS><BS>33<END><AltGr>2p<AltGr>2<LEFT><BS><RightShift>p  
<RIGHT>sswo<BS>0<AltGr>4d<LeftShift>1<BS><RightShift>-<BS>
```

This must now be correlated with the web form – as you can see there are <TAB> and <RET> keys pressed which mean jumping between fields and submitting the form.

Eventually a set of usernames and passwords can be found.

There is one trick though – the PCap does not include any information about keyboard layout that was used during capture. This must be figured out by trial and error.

Once the password is guessed, you get the flag from target server:



Done.

8. REFERENCES

<https://gist.github.com/MightyPork/6da26e382a7ad91b5496ee55fdc73db2>



ENISA
European Union Agency for Cybersecurity

Athens Office
1 Vasilissis Sofias Str.
151 24 Marousi, Attiki, Greece

Heraklion Office
95 Nikolaou Plastira
700 13 Vassilika Vouton, Heraklion, Greece



ISBN xxx-xx-xxxx-xxx-x
doi:xx.xxxx/xxxxxx
TP-xx-xx-xxx-EN-C



enisa.europa.eu