

CTF Writeup: 2024 ICS/IO CTF by Dragos

FER tim

Contents

1 Overview	3
2 Digitalna forenzička analiza (<i>digital forensics analysis</i>)	3
2.1 <i>Arpanet of the Future</i>	3
2.2 <i>When Are They?</i>	3
2.3 <i>De-Loreanize This</i>	3
2.4 <i>All Se Characters In Installation</i>	4
2.5 <i>Walking Into A Store And Buying Plutonium</i>	5
3 Dragos trivia	6
4 Embedded uređaji	7
4.1 <i>BAD - onBoard Access Device Embedded</i>	7
4.2 <i>BAD Login Challenge</i>	7
4.3 <i>BAD What Three Words</i>	9
4.4 <i>BAD Unit Test</i>	9
4.5 <i>BAD XZOR</i>	10
5 Otkrivanje	11
5.1 <i>Something Seems Off Doc</i>	11
5.2 <i>Where Do We Start?</i>	11
5.3 <i>Let's Talk About Port Hardening</i>	12
6 Prvobitni pristup (<i>initial access</i>)	13
6.1 <i>Knocking at the Door</i>	13
6.2 <i>Keys to the Kingdom</i>	13
6.3 <i>Connecting the Capacitor</i>	14
7 Postojanost (<i>Persistance</i>)	15
7.1 <i>Fluff Capacitor</i>	15
7.2 <i>We Don't Need No Roads</i>	15
7.3 <i>We Don't Need No Roads too</i>	16
8 Analiza ICS protokola	17
8.1 <i>Doc's Mysterious Transmission</i>	17
8.2 <i>Temporal Network Anomaly</i>	17
8.3 <i>Emmet's Energy Puzzle</i>	17
8.4 <i>The Biff Tannen Override</i>	18
8.5 <i>Save the Clock Tower Hack</i>	18
8.6 <i>Operation Lights Out</i>	20
9 Kontrola (<i>Command and control</i>)	23
9.1 <i>Botnet Binary of the Future</i>	23
10 Reversing	23
10.1 <i>Pastastrophe</i>	23
11 Zahvale	24

1 Overview

Početkom studenog održalo se virtualno Capture the Flag (CTF) natjecanje koje je trajalo 48 sati i bilo namijenjeno svima koji se bave ICS/OT sustavima, bez obzira na iskustvo. Sudionici su mogli igrati sami ili u timovima te rješavati izazove poput analize phishing e-mailova, PCAP datoteka ICS protokola, Windows logova, memorijskih slika, PLC programa i mrežnih dijagrama. Natjecanje je trajalo od 2. do 4. studenog i pružilo priliku za istraživanje različitih aspekata OT kibernetičke sigurnosti kroz praktične zadatke.

2 Digitalna forenzička analiza (*digital forensics analysis*)

2.1 Arpanet of the Future

Zadatak nas upućuje da istražimo Windows Event Viewer log file (.evtx). Uz traženje puno različitih događaja vidimo da je u jednom trenutku pokrenuta Powershell naredba sa zastavicom -enc koja predstavlja naredbu *encode*. Defaultno kodiranje je B64 te ako dekodiramo B64 kodirani string koji se nalazi u toj komandi dobijemo flag za ovaj zadatak.

FLAG NIJE OSTAVLJEN

2.2 When Are They?

SCADA database file se nalazi u /home/operator/.openscada/St.db i ondje je flag. Može se koristiti SQLite ili bilo koji drugi slični program.

FLAG NIJE OSTAVLJEN

2.3 De-Loreanize This

Zadatak

We caught someone that seemed to be from the past to issue archaic commands at our engineering workstation. Can you figure out what they did?

Our controller at 10.13.37.37 behaved oddly slow and rebooted once recently.

We only have Powershell Logs ... if we only had established monitoring for that controller before....

Ponovno kao i prije dobili smo samo .evtx datoteku (loceanized.evtx). Prilikom dekodiranja vidimo sljedeće:

```
modbus-send.exe 10.13.37.37 --payload  
111110101100111100110110110111001010101111011111101111110110  
10110000100100111111010110111010110111010011011101011001000010  
10011111110100110111010100011011010111110111011101001111001011010  
010011001100100110001111001101000001000001100010100011001101010101  
011000000111011010110111011101111111000
```

U dekodiranoj datoteci nalazi se tisuće takvih b64 kodiranih linija. Ideja je da se svi ti zapisi spoje u jednu datoteku te da se onda iz te datoteke pročita zastavica.

Prije svega valja pretvoriti .evtx datoteku u nekakav čitljivi format. Poslužiti ćemo se <https://github.com/omerbenamram/evtx> koristeći naredbu: evtx_dump loreanized.evtx > loreanized.xml Sada možemo lako pregledati zapisnike i vidjeti da se poziva powershell s kodiranim naredbama.

Možemo izdvojiti samo kodirane podatke koristeći osnovne alate:

```
cat loreanized.xml | grep 'powershell.exe -enc' | uniq | cut -d ' ' -f 3 > base64.txt
```

Primjer kako se jedna linija ovih kodiranih podataka može dekodirati:

```
import base64

line = <kodirani_podatak>
print(base64.b64decode(line).decode('utf-16'))
```

Dobiveni string može se podijeliti razmacima, a binarni niz može se izdvojiti.

Sljedeći skript dekodira linije, izdvaja binarne nizove i spaja ih u binarnu datoteku:

```
import base64

def decode_base64_to_binary(input_file, binary_output_file):
    concatenated_bits = ""

    with open(input_file, 'r') as infile:
        for line in infile:
            line = line.strip()
            decoded_line = base64.b64decode(line).decode('utf-16')
            binary_segment = decoded_line.split()[3]
            concatenated_bits += binary_segment

    with open(binary_output_file, 'wb') as outfile:
        byte_data = int(concatenated_bits, 2).to_bytes((len(concatenated_bits) + 7) // 8, byteorder='big')
        outfile.write(byte_data)

input_file = 'base64.txt'
binary_output_file = 'output.bin'

decode_base64_to_binary(input_file, binary_output_file)
```

I sada možemo jednostavno pronaći zastavicu:

```
strings output.bin | grep flag
```

Zastavica

```
flag{in_the_future_we_still_use_modbus}
```

2.4 All Se Characters In Installation

Zadatak

Someone managed to access our HMI. They popped an MS-Paint screen and weird characters appeared. They did not seem to type it.

Can you help to recover the message?

We managed to get a partial screenshot, but the important part of scrambled. Can you find something in the HMIs memory?

Good thing we have proven technology with XPerienced operators.

Nema postupka

FLAG NIJE OSTAVLJEN

2.5 Walking Into A Store And Buying Plutonium

Zadatak

We managed to came back to the store with the Finnish operating system and its HMI... its 2022 and the year of the Linux desktop ... they now have plutonium in stock, but the HMI has error messages.

The owner refuses to sell us plutonium until we find the root cause of the error messages.

He managed to capture a screenshot ... but it seems manipulated. Luckily he captured a Linux triage package.

Nije jasan postupak rješavanja.

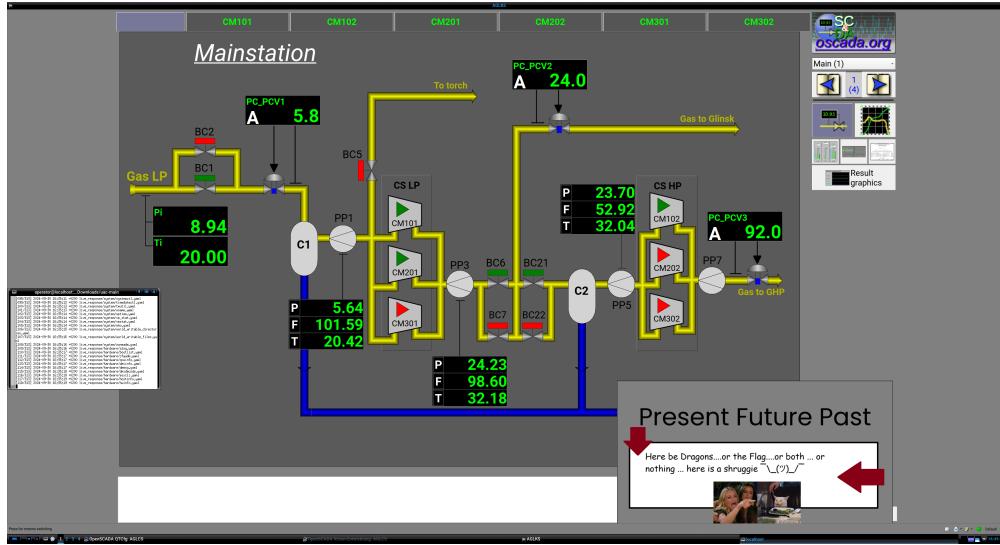


Figure 1: Slika iz zadatka.

FLAG NIJE OSTAVLJEN

3 Dragos trivia

Ovdje zaista nije potrebno previše govoriti. Dobijemo službene dokumente Dragosa iz kojih je potrebno izvući zastavice.

Zastavice nisu teške te se manje-više brzo pronalaze. Ovaj dio je popularan na svim kompanijskim CTF-ovima jer imaju priliku podijeliti javno dostupne dokumente te kompanije, točnije podižu svijest o njihovom postojanju.

Možda vam pri pronalaženju pomognu <https://youtu.be/babYCpnBbgk?t=1351> ili https://hub.dragos.com/hubfs/Reports/Dragos-FrostyGoop-ICS-Malware-Intel-Brief-0724_r2.pdf?hsLang=en

Sretno u potrazi :)

4 Embedded uređaji

4.1 *BAD - onBoard Access Device Embedded*

Zadatak

Zadano je pitanje što se ispisuje na komandnoj liniji kada se sustav pokrene. Uz zadatak dobivamo i ZIP datoteku koja sadrži `hddimg` datoteku.

- **Vrsta datoteke:** `hddimg` je slika diska, što se potvrđuje korištenjem naredbe `file`:

```
BAD.x86.hddimg: DOS/MBR boot sector, code offset 0x58+2, OEM-ID "SYSLINUX",
sectors/cluster 4, reserved sectors 4, root entries 512, Media descriptor
0xf8, sectors/FAT 212, sectors/track 32, heads 8, sectors 216512 (volumes >
32 MB), reserved 0x1, serial number 0xab4cfffc, label: "boot      ", FAT (16 bit)
```

- Budući da se radi o slici diska, možemo je pokrenuti pomoću QEMU emulatora.

Disk slika se pokreće koristeći sljedeću naredbu:

```
qemu-system-x86_64 -hda BAD.x86.hddimg -m 2048
```

- Opcija `-hda` govori QEMU-u da tretira datoteku kao sliku diska.
- Opcija `-m` dodjeljuje 2048 MB RAM-a virtualnom sustavu.

FLAG NIJE OSTAVLJEN

4.2 *BAD Login Challenge*

Zadatak

Zadano je pitanje što se ispisuje nakon što se prijavimo na uređaj za koji smo dobili sliku diska. Također su nam dani još dva formata iste slike diska: raspberry datotečna slika i OVA slika.

- **Vrsta datoteke:** `hddimg` je slika diska, što se potvrđuje korištenjem naredbe `file`:

```
BAD.x86.hddimg: DOS/MBR boot sector, code offset 0x58+2, OEM-ID "SYSLINUX",
sectors/cluster 4, reserved sectors 4, root entries 512, Media descriptor
0xf8, sectors/FAT 212, sectors/track 32, heads 8, sectors 216512 (volumes >
32 MB), reserved 0x1, serial number 0xab4cfffc, label: "boot      ", FAT (16 bit)
```

- Budući da se radi o slici diska, možemo je pokrenuti pomoću QEMU emulatora.

Koraci rješenja

Prvo moramo pristupiti datotekama `/etc/passwd` i `/etc/shadow`. Da bismo to postigli, trebamo montirati sliku diska na naš lokalni stroj kako bismo pristupili datotečnom sustavu na disku:

```
sudo mkdir /mnt/BAD
sudo mount -o loop BAD.x86.hddimg /mnt/BAD
```

Ovo će montirati datotečni sustav u direktorij /mnt/BAD. Opcija -o loop je potrebna jer mount očekuje fizički uređaj, a ovo je samo datoteka, pa stvaramo virtualni loopback uređaj.

Ako pogledamo direktorij /mnt/BAD, vidjet ćemo sljedeće datoteke:

```
> ls /mnt/BAD
bzImage  EFI  initrd  ldlinux.c32  ldlinux.sys  libcom32.c32  libutil.c32  rootfs.img  startup.nsh  sys...
```

Ovo su tipične datoteke za Linux sustav, a datoteka koja nas zanima je `rootfs.img`. Ova datoteka sadrži datotečni sustav koji koristi sustav nakon što se pokrene, pa korisnik koji je tamo smješten bit će korisnik koji se očekuje da se prijavi. Budući da je ovo još jedna slika datotečnog sustava, možemo je montirati baš kao i `.hddimg` datoteku.

```
sudo mkdir /mnt/BADrootfs
sudo mount -o loop rootfs.img /mnt/BADrootfs
```

Kada pogledamo direktorij /mnt/BADrootfs, vidjet ćemo tipičnu Linux strukturu direktorija. Tamo možemo vidjeti datoteke koje tražimo.

Ako pogledamo datoteku `/etc/shadow`, izgledat će ovako:

```
> cat /etc/shadow
root:$5$4RZXNQ5iSvB1GjaJ$54E3xgE5fH2mTX8bho/W9LBHTHmeeQ/NhxpAqcv8Yd1:15069:0:99999:7:::
daemon:*:15069:0:99999:7:::
bin:*:15069:0:99999:7:::
sys:*:15069:0:99999:7:::
sync:*:15069:0:99999:7:::
games:*:15069:0:99999:7:::
man:*:15069:0:99999:7:::
lp:*:15069:0:99999:7:::
mail:*:15069:0:99999:7:::
news:*:15069:0:99999:7:::
uucp:*:15069:0:99999:7:::
proxy:*:15069:0:99999:7:::
www-data:*:15069:0:99999:7:::
backup:*:15069:0:99999:7:::
list:*:15069:0:99999:7:::
irc:*:15069:0:99999:7:::
_apt:*:15069:0:99999:7:::
nobody:*:15069:0:99999:7:::
```

Iz ovoga možemo vidjeti da postoji root korisnik koji ima SHA256 hashiranu lozinku. Tu lozinku možemo uzeti i pokušati dešifrirati pomoću alata poput `john the ripper`, no nisam uspio u tome.

Zato možemo iskoristiti drugi trik. Svaka promjena koju napravimo unutar montiranog datotečnog sustava bit će odražena u datoteci iz koje smo montirali datotečni sustav. Dakle, ako promijenimo datoteku `/etc/shadow` u direktoriju /mnt/BADrootfs, to će biti odraženo u datoteci `rootfs.img` u direktoriju /mnt/BAD, koja će potom biti odražena u našoj početnoj slici diska `BAD.x86.hddimg`. Promijenit ćemo datoteku tako da root korisnik nema lozinku. Ovo možemo postići jednostavnim uklanjanjem druge kolone u datoteci, na sljedeći način: `root::15069:0:99999:7:::`. Ako spremimo ove promjene i odmontiramo oba datotečna sustava pomoću naredbe `umount`, trebali bismo imati promijenjenu sliku diska u kojoj root korisnik nema lozinku.

```
sudo umount /mnt/BADrootfs
sudo umount /mnt/BAD
```

Da bismo to potvrdili, možemo pokrenuti sliku pomoću QEMU-a:

```
qemu-system-x86_64 -hda BAD.x86.hddimg -m 2048
```

Nakon što slika završi s pokretanjem, jednostavno se prijavite kao root korisnik i primijetit ćete da nema prompta za lozinku, a sustav ispisuje flag.

FLAG NIJE OSTAVLJEN

4.3 *BAD What Three Words*

Zadatak

Zadatak traži da brute-forcamo lozinku iz prethodnog zadatka. Također, dobili smo obrazac izgleda lozinke: `sprink**.fevere*.shoc**`.

Koraci rješenja

Za početak, moramo pretvoriti datoteke `/etc/shadow` i `/etc/passwd` u format koji `john` razumije. Naredba za to izgleda ovako:

```
unshadow /etc/passwd /etc/shadow > unshadow.txt
```

Zatim, pokrenimo `john` s odgovarajućim maskama za brute-forcing:

```
john --mask="sprink?1?1.fevere?1?1.shock?1?1" unshadow.txt
```

Ovdje `?1` označava da `john` pokušava bilo koje malo slovo na tom mjestu.

Na mom računaru, ovo je trajalo oko 15 minuta da bi se pronašla točna lozinka: `sprinkle.fevered.shocks`. Ako ste slučajno očistili terminal ili iz nekog razloga trebate ponovo vidjeti lozinku, možete koristiti sljedeću naredbu:

```
john --show unshadow.txt
```

FLAG NIJE OSTAVLJEN

4.4 *BAD Unit Test*

Zadatak

Pronašli smo tri riječi i znamo koje tri riječi imaju sličan format. Sada moramo saznati što se nalazi na toj lokaciji.

Razvoj ovog računara čini se da potiče iz neobičnog izvora. Još uvijek nismo u mogućnosti prepoznati jezik i moramo se osloniti na geografiju. Guten Tag?

Možete li pronaći broj jedinice odgovorne za manipulaciju povezani s tom lokacijom?

Format flag-a: Unit NUMBER

FLAG NIJE OSTAVLJEN

4.5 BAD XZOR

Zadatak

Uspjeli smo obrnuto inženirati standard za pohranu podataka na BAD uređaju. I to je loše. Možda ste već pronašli zip arhiv na našem firmware imidžu - ako niste, pronađite je. Koristite tri riječi (sve malim slovima, spojene točkama, npr. banana.potato.squirrel) za otključavanje arhive.

Uspjeli smo identificirati da čini se da nije potreban poseban ključ za dekriptiranje datoteke - sve je tu, čim sustav pokrene.

Koja je godina... Budućnost, Sadašnjost, Prošlost? Izgubio sam trag. Možda trebamo napraviti korak unatrag.

Ljubazno podsjećanje da će timovi biti KAZNJENI za savjete na izazovima sa 800 i 1000 bodova!

Prvo, pronađite zip arhiv na firmware slici i otključajte je pomoću tri riječi koje ste dobili (sprinkle.fevered.shocks).

Zatim, pronađite osnovnu datoteku unutar arhive. Nakon što je pronađete, trebate provesti operaciju XOR na datoteci koristeći njezin vlastiti ključ. To možete postići tako da svaku bajt vrijednost datoteke XORRate sa ključem koji je već prisutan u datoteci.

Kada završite s prvim korakom, datoteku treba dodatno obraditi. Svaki byte osnovne datoteke treba XOR-irati s vrijednošću 48. Za ovo ćete vjerojatno koristiti skriptu koja će obraditi sve bajtove.

Sljedeći korak je učitati rezultat XORanja. Nakon što su svi bajtovi obrađeni, koristite naredbu `cat` kako biste ispisali datoteku i pregledali sadržaj.

Ovisno o formatu izlazne datoteke, trebate analizirati sliku ili drugi zapis u kojem će biti isписан flag. Obratite pažnju na rezultate i uvjerite se da ste ispravno obradili sve bajtove.

Na kraju, trebali biste moći prepoznati i izdvojiti flag.

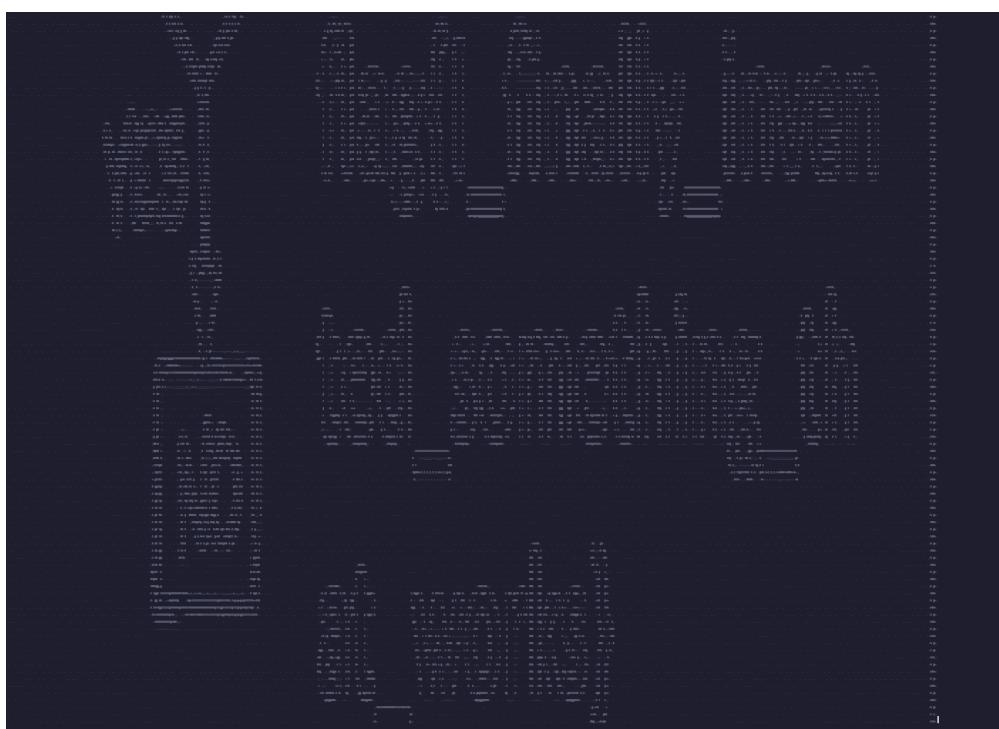


Figure 2: Izgled flaga

5 Otkrivanje

5.1 Something Seems Off Doc

Zadatak

Zadana nam je pcap datoteka za analizu kako bismo pronašli ime domaćinskog računala.

- **Postupak rješenja:** Budući da ne postoji DNS protokol u prometu, pretpostavljamo da se radi o lokalnoj mreži, pa pokušavamo pronaći **nbns** (NetBIOS Name Service) protokol.

- **Koraci:**

- Analizirajte pcap datoteku za pronađak **nbns** protokola.
 - Nakon što pronađemo **nbns** promet, provjeravamo odgovor na upit za ime.

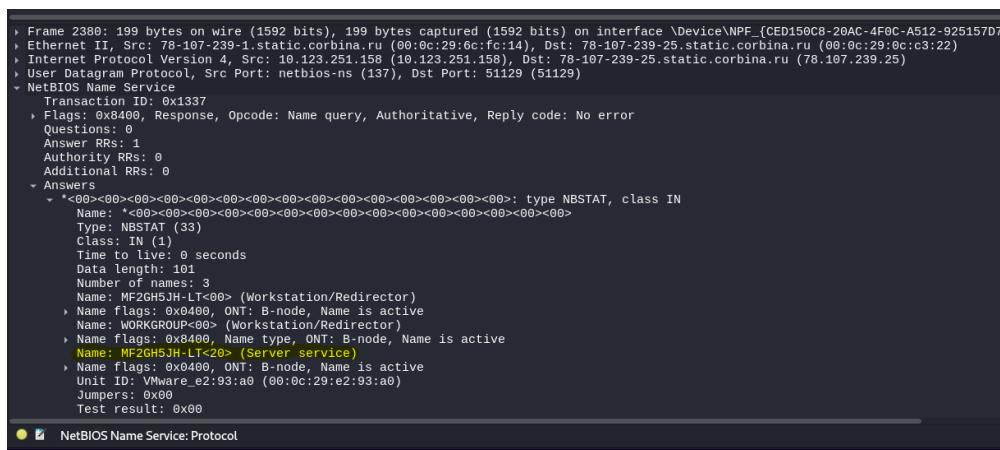


Figure 3: Izgled iz Wiresharka

MF2GH5JH-LT

5.2 Where Do We Start?

Zadatak

Ovaj zadatak je nastavak analize pcap datoteke iz zadatka "Something Seems Off Doc". Ovdje je cilj pronaći grad povezan s vanjskom IP adresom.

- **Postupak rješenja:** Koristimo DNS razlučivanje (name resolution) kako bismo dobili trag koji vodi do grada.

- **Koraci:**

- U Wiresharku uredimo postavke za razlučivanje imena za IP adresu.
 - Nakon što uredimo postavke, dobivamo dodatne informacije o IP adresi: **78-107-239-25.static.corbina.ru**.
 - Korištenjem domene **.ru** koja je rezervirana za Rusiju, pretražujemo **corbina** i nalazimo naziv tvrtke čije je sjedište u Moskvi. To je grad koji tražimo.

Grad: Moskva, Rusija

5.3 Let's Talk About Port Hardening

Zadatak

Ovaj zadatak je nastavak zadataka "Something Seems Off Doc" i "Where Do We Start?". Napadač je uspio pristupiti korporativnim resursima iskorištavanjem pogrešno konfiguriranog FTP rješenja. Naš zadatak je otkriti koja je aplikacija korištena i koja verzija te aplikacije.

- **Postupak rješenja:** Analiziramo FTP protokole kako bismo identificirali aplikaciju.
- **Koraci:**
 - Pronalazimo sljedeći odgovor u FTP komunikaciji: 220, što označava pozitivnu potvrdu da je poslužitelj spreman prihvati nove klijentske veze.
 - Prema ovom odgovoru zaključujemo da je aplikacija koja se koristi FileZilla Server u verziji 1.8.2.



Figure 4: Polje paketa u Wiresharku

Aplikacija: FileZilla Server, Verzija: 1.8.2

6 Prvobitni pristup (*initial access*)

6.1 Knocking at the Door

Zadatak

Ovaj zadatak je nastavak prethodnih, "Something Seems Off Doc", "Where Do We Start?" i "Let's Talk About Port Hardening". Datoteka je preuzeta korištenjem FTP protokola. Naš zadatak je otkriti ime preuzete datoteke.

- **Postupak rješenja:** Analiziramo FTP protokole i tražimo naredbu RETR koja se koristi za preuzimanje datoteka.
- **Koraci:**
 - Pronalazimo upotrebu RETR naredbe u FTP komunikaciji, što nam pokazuje da je datoteka preuzeta.
 - Ime preuzete datoteke je `dontforget.txt`, što je naš odgovor.

Ime datoteke: `dontforget.txt`

6.2 Keys to the Kingdom

Zadatak

Ovaj zadatak je nastavak prethodnih, "Something Seems Off Doc", "Where Do We Start?", "Let's Talk About Port Hardening" i "Knocking at the Door". Preuzeta datoteka sadrži korisničko ime i lozinku. Naš zadatak je pronaći lozinku.

FTP	54 Response: 200 PORT command successful.
TCP	75 Request: RETR dontforget.txt
TCP	66 58241 → 4297 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
TCP	66 4297 → 58241 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
FTP	83 Response: 150 Starting data transfer.
TCP	66 58241 → 4297 [ACK] Seq=1 Ack=1 Win=262656 Len=0
TCP	74 58241 → 4297 [PSH, ACK] Seq=1 Ack=1 Win=262656 Len=20
TCP	66 58241 → 4297 [FIN, ACK] Seq=21 Ack=1 Win=262656 Len=0

Figure 5: Paket u Wiresharku

00 0c 29 0c c3 22 00 0c 29 6c fc 14 08 00 45 00 ..)..")l...E..
00 3c 45 fe 40 00 7f 06 72 1f 0a 7b fb 9e 4e 6b <E @... r...{..Nk
ef 19 e3 81 10 c9 7a 5e 0e 8c 2f 25 97 56 50 18z^ .../% VP..
04 02 8e 0a 00 00 76 69 63 6b 69 0d 0a 4d 30 72vi cki..MOr
65 50 72 6f 6a 65 63 74 73 21 eProject s!

Figure 6: HEX zapis lozinke i korisničkog imena

- **Postupak rješenja:** Analiziramo FTP komunikaciju i gledamo podatke koji su poslati nakon preuzimanja datoteke.
- **Koraci:**

- Nakon što je FTP pozvan, vidimo da je započela prijenos podataka. Odabrani TCP segment ima oznaku [PSH, ACK] i dužinu 20 bajtova, što znači da se podaci nalaze u tom segmentu.
- Proučavanjem heksadecimalnog prikaza podataka, možemo vidjeti korisničko ime i lozinku.
- Lozinka je M0reProjects!.

Lozinka: M0reProjects!

6.3 *Connecting the Capacitor*

Zadatak

Svaki put kada povezujemo USB uređaj s Fluxcapacitor-om, pojavljuje se više datoteka nego što smo očekivali. Naš zadatak je pomoći u identificiranju USB uređaja koji je povezan. Sumnjamo da je barem jedan od uređaja prethodno bio korišten na starom industrijskom računalu.
Na kraju, potrebno je pronaći vrijednost **FriendlyName** USB uređaja.

- **Flag format:** Vendor Storage Type (5 Elements separated by spaces).

FLAG NIJE OSTAVLJEN

7 Postojanost (*Persistence*)

7.1 Fluff Capacitor

Zadatak

Vjeverice... vjeverice...

Jedna od tih malih stvorenja uspjela je oštetiti DeLorean. I to ne onako kako biste pomislili... bila je to cyber vjeverica.

Naš računar je sada pun pahuljastih slika, ali samo je jedna od njih "prava". Koja vjeverica je ona koja nosi monokl, leptir-mašnu, cilindar i puši cigaru, a koja se pojavila u pozadini?

Uspjeli smo prikupiti sve datoteke sa sustava na daljnju analizu. Neke datoteke na C: su bile "pahuljaste". Umjesto dumpa memorije, dobili smo sistemske datoteke.

Možemo li prepoznati sliku vjeverice koja je "fluff" i de-fluffati je?

FLAG NIJE OSTAVLJEN

7.2 We Don't Need No Roads

Zadatak

Napomena: Koristite priloženi ZIP za odgovaranje na sljedeća dva pitanja!

Platili smo skupe usluge za potpuno autonomno putovanje kroz vrijeme, ali još uvijek ne radi. Čini se da nam ipak trebaju ceste da bismo negdje stigli, a naš "pretplata za putovanje kroz vrijeme" izgleda da se može daljinski kontrolirati.

Sumnjamo da nešto nije u redu nakon posljednjeg posjeta prodavaču automobila. Spomenuli su nešto o ažuriranju firmvera i "nole modu".

Koje "Ime tvrtke" je odgovorno za probleme s našim DeLoreanom?

Flag Format: flag{enterthistext}

- Za početak, analizirat ćemo snimku memorije pomoću volatility2 (jer volatility3 nije radio).
- Prvo koristimo imageinfo kako bismo identificirali tip slike memorije:

```
python2 vol.py -f /path/to/memory/image.vmem imageinfo
```

Ovdje ćemo vidjeti da je slika memorije Windows XP i to ćemo koristiti za daljnje analize.

- Sljedeći korak je lista svih procesa pomoću pslist komande:

```
python2 vol.py -f /path/to/memory/image.vmem --profile=WinXPSP2x86 pslist
```

Ovdje ćemo primijetiti sumnjiv proces nazvan `We don't need no roads.exe`.

- Za detalje o ovom procesu koristimo verinfo komandu koja će nam otkriti ime tvrtke odgovorne za ovaj proces:

```
python2 vol.py -f /path/to/memory/image.vmem --profile=WinXPSP2x86 verinfo
```

Iz podataka o verziji pronaći ćemo ime tvrtke.

FLAG NIJE OSTAVLJEN

7.3 We Don't Need No Roads too

Zadatak

DeLorean se ponaša drugačije nego inače. Svaki put kad pokrenemo stroj, svjetla trepere nerazmjerno. Svaki put kad obavimo postupak pokretanja.

Kad smo istraživali, nismo još pregledali sve registre. Možete li nam pomoći pronaći što točno ometa pokretanje? Znamo da pokreće neki program, ali nemamo ključ za vrijednosti.

Dijelovi memorije su oštećeni i moramo biti uporni. Ključevi registra su u čudnom jeziku.

Podsjetnik: Timovi će biti kažnjeni ako uzmu pomoć za izazove s 800 i 1000 bodova!

Flag Format: Slično normalnim flagovima, ali malo drugačije. Flag ne sadrži heksadecimalne brojeve.

Koraci: Ovaj zadatak zahtijeva analizu memoriske slike, specifično registar podataka. Koristit ćemo **volatility** kako bismo pristupili registar podacima i identificirali program odgovoran za ometanje pokretanja.

- Prvo, pregledavamo memorisku sliku pomoću **hivelist** komande u **volatility** kako bismo otkrili dostupne registar hives.

```
python2 vol.py -f /path/to/memory/image.vmem --profile=WinXPSP2x86 hivelist
```

Ovaj korak će identificirati sve registre na sustavu. Ključne stavke su oni koji se odnose na **NTUSER.DAT** datoteke, što su korisnički registri.

- Zatim, istražujemo registar koji se odnosi na **Run** ključeve, jer ovi ključevi sadrže podatke o programima koji se automatski pokreću prilikom startanja računala. U ovom slučaju, koristimo **printkey** komandu:

```
python2 vol.py -f /path/to/memory/image.vmem --profile=WinXPSP2x86 printkey -o 0xe1ce0008 -K "S
```

U ovom primjeru, otkrivamo vrijednosti koje nisu ispravno prikazane zbog nepoznatog jezika u registru.

- Kako bismo u potpunosti dobili sirove podatke iz registra, koristimo **dumpregistry** komandu koja izdvaja kompletan registar u datoteku:

```
python2 vol.py -f /path/to/memory/image.vmem --profile=WinXPSP2x86 dumpregistry --dump-dir reg
```

Ova komanda izdvaja registar u datoteku koju možemo istraživati.

- Zatim možemo koristiti alat kao što je **hivex** za analizu registrirane datoteke:

```
hivexsh registry.0xe1ce0008.NTUSERDAT.reg
```

U ovom alatu, izdvajamo vrijednosti ključeva pomoću **lsvat** komande da bismo dobili vrijednost nepoznatog ključa, koja nam otkriva naziv programa koji je odgovoran za problem.

flag{DeLorean}

8 Analiza ICS protokola

8.1 Doc's Mysterious Transmission

Zadatak

Dok je čistio garažu, Marty McFly je naišao na stari računar s tajanstvenom porukom od Doc-a, datiranu na 26. listopad 1985. Poruka se čini da se odnosi na kontrolni sustav za DeLoreansov flux capacitor, ali završava naglo.

Analiziraj priloženi PCAP fajl, koji sadrži mrežni promet snimljen s Doc-ovog računara. Tvoj zadatak je dekodirati transmisiju i dohvatiti tajnu frazu koju je Doc koristio za aktiviranje flux capacitor-a. Koja je tajna fraza?

FLAG NIJE OSTAVLJEN

8.2 Temporal Network Anomaly

Zadatak

Doc-ov posljednji eksperiment uključuje slanje podataka unazad kroz vrijeme. PCAP fajl s ovog eksperimenta pokazuje čudne uzorke i anomalije koje bi potencijalno mogle otkriti tajne putovanja kroz vrijeme.

Analiziraj priloženi PCAP fajl kako bi razumio prirodu vremenskih anomalija. Tvoj cilj je odrediti točan trenutak u podacima kada se vremenska distorzija dogodila.

Možeš li prepoznati budući datum?

Način Rješavanja : Da bismo riješili ovaj zadatak, analizirali smo PCAP fajl i primijetili funkciju koja čita vrijeme. Ova funkcija je pozvana četiri puta, a tri puta se odnosi na godinu 2014. Samo je jedan poziv imao datum iz 2034. godine, što ukazuje na to da je ta vrijednost anomalija iz budućnosti. Ova anomalija je ključna za odgovor na pitanje.

▼ S7 Timestamp: Aug 20, 2034 11:59:44.957

Figure 7: Flag

Aug 20, 2034 00:00:00.000

8.3 Emmet's Energy Puzzle

Zadatak

Imamo pcap fajl u kojem je promijenjena vrijednost. Naša zadaća je analizirati podatke, razumjeti promjenu i otkriti sustav u kojem je vrijednost promijenjena. Znamo da je za Doc-ovo putovanje potrebna energija od 1.21GW.

Tvoj zadatak je pronaći anomaliju i točan sustav u kojem je vrijednost promijenjena.

Način Rješavanja : Za rješenje ovog zadatka otvorili smo pcap fajl u Wiresharku i koristili opciju "Follow TCP Stream" kako bismo pregledali podatke iz razgovora. Podaci su bili šifrirani u base64 formatu, pa smo ih dekodirali kako bi postali čitljivi. Nakon dekodiranja podataka, otkrili smo da je vrijednost potrebne energije postavljena na 3.21GW unutar sustava DeLoreanPowerSystem.EnergyDistribution.TimeTravelCircuits. Ovaj sustav je bio odgovoran za promjenu vrijednosti.

DeLoreanPowerSystem.EnergyDistribution.TimeTravelCircuits

8.4 The Biff Tannen Override

Zadatak

Biffovi prijatelji uspjeli su infiltrirati sustav za upravljanje prometom u Hill Valleyu. Ostavili su konfiguracijsku datoteku na mreži koja može manipulirati semaforima u gradu, uzrokujući nered. Imate pristup snimci prometa s ovog sustava. Vaš zadatak je razumjeti Biffove naredbe za preuzimanje kontrole i vratiti normalno stanje prometa.

Koje je ime jedne od glavnih ulica koje je Biff izmijenio?

Način Rješavanja : Za rješenje zadatka koristili smo dva stremna podataka: `xdata.txt` i `xdata_s.txt`. Prva IP adresa bila je označena kao 1.20, a druga kao 1.30. Koristio sam XOR operaciju za usporedbu oba stremova (`xdata` i `xdata_s`) i za svaki bit primjenio XOR s vrijednošću 01111001. Nakon izvođenja XOR operacija, mogli smo dešifrirati informacije koje su sadržavale ime glavne ulice koju je Biff modificirao.

FLAG NIJE OSTAVLJEN

8.5 Save the Clock Tower Hack

Zadatak

Identificirana prijetnja, Biffovi prijatelji, ciljali su sat na zvoniku u Hill Valleyu, pokušavajući promjeniti rad električnog motora koji podiže utege sata. Srećom, pristup sustavu za kontrolu motora je nadgledan, a snimke mrežnog prometa su dostupne.

Triage tim nije pronašao ništa zlonamjerno u PCAP-ovima, ali Dragos Intelligence tim je pružio dodatne uvide koji bi trebali biti pregledani detaljnije. Čini se da Biffovi prijatelji obično kradu inženjerske specifikacije prije nego što pokušaju manipulirati kontrolnim sustavom.

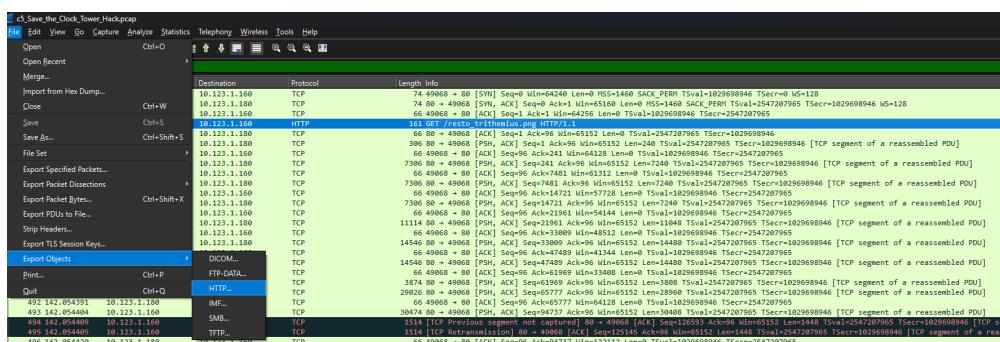


Figure 8: Paket u Wiresharku

Način Rješavanja Iz PCAP datoteke izvukli smo sliku, koju smo zatim obradili pomoću online alata AperiSolve. Rezultat obrade je bio rješenje:



Figure 9: Orginalna slika

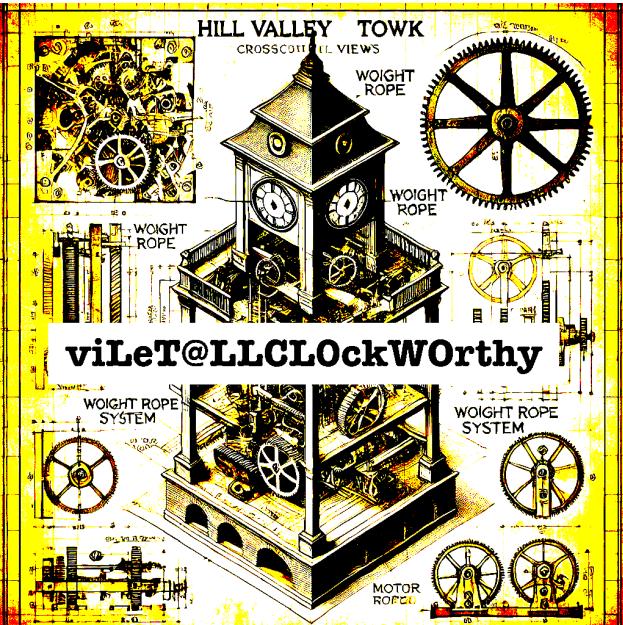


Figure 10: Filtrirana slika

viLeT@LLCLOCKWOrthy

8.6 Operation Lights Out

Zadatak

A mysterious malfunction has occurred within Hill Valley's traffic light control system, causing widespread confusion and traffic jams. Preliminary investigations suggest that someone has tampered with the communication between the master control unit and individual traffic light controllers. A PCAP file containing the network traffic during the incident is the only clue available.

General Instructions for the Player: Examine the provided PCAP file to trace the source and nature of the anomalies in the communication between the master control and the traffic lights. Your objective is to determine the changes made and identify any unusual patterns or commands that could explain the disruptions.

Flag: The flag is a mysterious command and a message. YOU MUST BUILD THE WHOLE COMMAND (not just the payload!) ... but don't include lower layers info (e.g., Ethernet, IP, or TCP/UDP data). Submit the flag as a hex string representation without any delimiter:
AABBCCDDEEFF...

NOTE: Both lowercase and uppercase hex are accepted.

Free hints:

- Use the .lua
- Industrial protocols usually have some well-defined structure (header, command, payload, checksum)
- Industrial protocols are usually the payload of other layers *etherenet/ip/tcp|udp/ind_protocol*.
- Just because you don't see a command in a capture, it doesn't mean it doesn't exist.

Način Rješavanja Prvo smo analizirali PCAP datoteku kako bismo pronašli ključne informacije. Izdvojili smo dvije stavke:

- **Lozinku:** GR3@t_Sc0tT_M@Rt1
- **Binary zip arhivu:** koja je bila kodirana u podatkovnom streamu.

Potom smo koristili alat `xxd` za pretvaranje binarnog streama u zip arhivu. Kada smo arhivu raspakirali koristeći pripadajuću lozinku, dobili smo sliku. Na slici je bila prikazana komanda i poruka potrebna za generiranje završnog zahtjeva (flaga).



Figure 11: Paket u Wiresharku

Na temelju analize i primjera strukture datagrama HVLTP-a (Hill Valley Traffic Light Protocol), definirali smo paket kako slijedi:

```
magic.hvltpVer.requestOrResponse.command.length(1byte-2HEXchrs).message.checksum(2bytes-4HEXchars)
```

Primjer skripte za dešifriranje HVLTP protokola:

```
hvtlp = Proto("HVTLP", "Hill Valley Traffic Light")
```

```
-- observed commands
local COMMANDS = {
    [0x10] = "ID",
    [0x11] = "Address",
    [0x12] = "Status",
    [0x14] = "Durations"
}
```

```
hvtlp.fields.magic = ProtoField.string("hvtlp.magic", "Magic")
hvtlp.fields.version = ProtoField.uint8("hvtlp.version", "Version")
hvtlp.fields.type = ProtoField.uint8("hvtlp.type", "Type", base.DEC, {[1] = "Request", [2] = "Response"})
hvtlp.fields.command = ProtoField.uint8("hvtlp.command", "Control Code", base.HEX, COMMANDS)
```

```

function hvtlp.dissector(buffer, pinfo, tree)

    if (buffer:len() < 10) then return end

    local magic = buffer(0,5):string()
    if magic ~= "1.2GW" then return end

    local request = buffer(6,1):uint()
    local cmd = buffer(7,1):uint()

    pinfo.cols.protocol = hvtlp.name
    local subtree = tree:add(hvtlp, buffer(), "Hill Valley Traffic Light")
    subtree:add(hvtlp.fields.magic, buffer(0,5))
    subtree:add(hvtlp.fields.version, buffer(5,1))
    subtree:add(hvtlp.fields.type, buffer(6,1))
    subtree:add(hvtlp.fields.command, buffer(7,1))

    if COMMANDS[cmd] ~= nil then
        pinfo.cols.info = COMMANDS[cmd] .. " " .. (request == 1 and "Request" or "Response")
    else
        pinfo.cols.info = string.format("Unknown cmd [%02x] ", cmd) .. (request == 1 and "Request" or "Response")
    end

end

tcp_table = DissectorTable.get("tcp.port"):add(54321, hvtlp)

```

Na temelju ove strukture kreirali smo naš flag, pazeći da dodamo nule na kraju za checksum. Konačni flag u HEX formatu je:

312e3247570102ce002477335f6e3333645f4e305f726f4064730000

9 Kontrola (*Command and control*)

9.1 Botnet Binary of the Future

Zadatak

Uspjeli smo doći do uzorka. Ovaj put ga nemamo u memoriji, već je izvađen s jednog od USB uređaja povezanih s plotonijem za fluxcapacitor.

Pronašli smo nekoliko P1 centrifugnih planova koji su zanimljivi, ali ne čine se povezani s binarnom datotekom - ili možda jesu?

U svakom slučaju, možeš li saznati koji IOCs (indikatori kompromitacije) možemo koristiti? Čuo sam da su tagovi relevantni za OT komponente.

Lozinka za zip: infected

FLAG NIJE PRONAĐEN

10 Reversing

10.1 Pastastrophe

Zadatak

Ažurirani program upravljanja promjenama je nešto što bi svaka organizacija trebala implementirati kako bi osigurala prijenos znanja... ali nisu sve organizacije to učinile...

I evo nas, prethodni PLC inženjeri su otišli i ostavili malo ili nimalo dokumentacije. Interni tim pokušava dohvati vjerodajnice, ali nemaju sreće. Ne znaju lozinku za ovaj nepoznati uređaj. Možete li izvući lozinku?

Napomena: Timovi će biti kažnjeni za korištenje savjeta u izazovima od 800 i 1000 bodova!

FLAG NIJE PRONAĐEN

11 Zahvale

Zahvaljujem se svima na sudjelovanju u ICS/IO CTFu by Dragos. Čestitam nama na dobrim rezultatima!

Posebne zahvale: prof. dr. sc. Stjepan Groš
mag. ing. Filip Katulić