# Dokumentace k projektu do předmětu BIS

Kateřina Fořtová (xforto00)

prosinec 2021

## 1 Zmapování vnitřní sítě

Pro zjištění IP adresy a masky jsem po připojení na `student@bis.fit.vutbr.cz` (BIS server) provedla příkaz `ip addr`. Zjištěná IP adresa a maska jsou 192.168.10.152/24. Následně bylo provedeno skenování této sítě pomocí `nmap 192.168.10.152/24 -Pn`. Ve výpisu různých stanic studentů a dalších mě zaujaly stanice potenciálně ukrývající tajemství:

```
$ nmap -sP 192.168.10.152/24 -Pn | grep sv
Nmap scan report for xsvacd00 (192.168.10.37)
Nmap scan report for sv6 (192.168.10.145)
Nmap scan report for sv1 (192.168.10.150)
Nmap scan report for sv2 (192.168.10.166)
Nmap scan report for sv3 (192.168.10.170)
Nmap scan report for xsvenk00 (192.168.10.194)
Nmap scan report for sv4 (192.168.10.199)
```

Když nebudeme uvažovat i dvě nalezené studentské stanice, tak získáme podezřelé servery `sv1`, `sv2`, `sv3`, `sv4` a `sv6`. Následně jsem pro každý server zmapovala otevřené porty se službami, které by mohly být klíčem k různým druhům tajemství:

```
$ nmap -sT -P0 192.168.10.150
Nmap scan report for sv1 (192.168.10.150)
Host is up (0.72s latency).
Not shown: 998 filtered ports
PORT     STATE SERVICE
22/tcp   open  ssh
2049/tcp open  nfs

$ nmap -sT -P0 192.168.10.166
Nmap scan report for sv2 (192.168.10.166)
Host is up (0.48s latency).
Not shown: 999 filtered ports
PORT   STATE SERVICE
22/tcp open  ssh

$ nmap -sT -P0 192.168.10.170
Nmap scan report for sv3 (192.168.10.170)
Host is up (0.61s latency).
Not shown: 996 filtered ports
PORT     STATE  SERVICE
22/tcp   open   ssh
80/tcp   open   http
443/tcp  closed https
3306/tcp open   mysql

$ nmap -sT -P0 192.168.10.199
Nmap scan report for sv4 (192.168.10.199)
Host is up (0.42s latency).
Not shown: 998 filtered ports
PORT   STATE SERVICE
21/tcp open  ftp
```

```
22/tcp open  ssh

$ nmap -sT -P0 192.168.10.145
Nmap scan report for sv6 (192.168.10.145)
Host is up (0.00072s latency).
Not shown: 998 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
5000/tcp open  upnp
```

## 2 Tajemství B, A a C

Snažím se stáhnout obsah webové stránky na serveru `sv3` na portu 80 pomocí příkazu `curl`. Dozvídám se, že na dané stránce se nachází webový přihlašovací formulář. Vytvořím SSH tunel pro zobrazení webové stránky ve svém prohlížeči a pokouším se o jeden z nejčastějších možných útoků, které na webové přihlašovací formuláře mohou být provedeny – SQL Injection. Pro útok používám login `' or 1=1 --` a heslo `johnny`. SQL Injection byla úspěšná a po přihlášení se mi zobrazí tajemství B:

`Tajemstvi B_03-12-13-15-01_179115ac45fb6763dc7de63308f867042f7aa3f87746cfa805f7684b8fbb481c`

Vedle přihlašování pomocí formuláře se na stejné stránce nachází i odkaz na přihlášení administrátora – na stránce `admin.html` je pole pro zadání osobního kódu. Přepínám se v prohlížeči Chrome na zobrazení zdroje této webové stránky a nacházím soubor `index.js`, kde je princip vypočítání checksum a následně checksum, která je pro dané zobrazení webové stránky správná. Postupem odzadu – dosazením správné checksum do poslední rovnice, úpravami rovnic, se dostávám k osobnímu kódu, který po výpočtech dává správnou checksum. Pro můj osobní případ byla checksum 226531314 a já se postupným zpětným výpočtem dostala ke kódu 18877. Po zadání tohoto kódu do daného pole jsem získala tajemství A:

`A_03-12-18-45-02_c570db9ffc17337450c4a4b61a7cb34c7479729a63a5a59549a5ede213b34c79`

Ještě před samotným nalezením tajemství A mě zaujal komentář ve zdroji stránky s webovým formulářem:
`<!-- Admin, please care for absconditum directory -->`
Slovo absconditum znamená latinsky skryté, proto se snažím využít tohoto slova a dostat se přes URL odkaz a složku `absconditum` k tajemství `secret.txt`. Jsem však neúspěšná a nakonec využívám nástroj `gobuster` se seznamem slov (`https://github.com/v0re/dirb/blob/master/wordlists/common.txt`), do kterého ještě přidávám možná podezřelá slova `absconditum` a `secret`. Ve výpisu získávám mimo jiné odkaz na cestu `http://localhost:3880/hidden`, což by odpovídalo anglickému překladu daného latinského slova. Dále se snažím hledat v této složce se stejným upraveným wordlistem soubory s několika typickými příponami:

```
$ gobuster -e -u http://localhost:3880/hidden/ -w common.txt -x txt,php,html,htm
http://localhost:3880/hidden/secret.php (Status: 200)
```

Po přístupu na toto URL konečně získávám tajemství C:

`Tajemstvi C_05-12-14-45-01_3cba78700ec332c61b44b4dba0caf988bfcc5a8572f0fe1bf6a6e09f1f0629af`

## 3 Tajemství J

Na `sv4` se nachází otevřený FTP port. Na vstupním přihlašovacím studentském serveru BIS není dostupný `ftp` příkaz, proto si soubory z FTP serveru stahuji pomocí příkazu `curl`. Mimo jiné jsou součástí stažených souborů i dva zdánlivě stejné obrázky. Snažím se zjistit principy možného kódování zpráv v obrázcích, nakonec oba obrázky nahrávám do steganografického nástroje na webu `https://lukeslytalker.pythonanywhere.com/outguess/scan`. Při analýze `image1.jpg` jsem úspěšná a získávám z obrázku tajemství J:

`Tajemstvi J_05-12-21-15-01_164398c042740190f0a6763ae7217ace2069b0a57f940ff091e00ea0bf5e0ac9`

## 4 Tajemství D

Na `sv6` je otevřen port 5000. Tento port mimo jiné slouží pro provoz dockeru. Na BIS serveru je k dispozici `docker` příkaz. Snažím se zjistit, zda se právě na `sv6` nenachází nějaká docker image. Po chvíli bádání nacházím příkaz pro získání jména repozitáře na vzdáleném serveru:

```
$ curl -X GET http://192.168.10.145:5000/v2/_catalog
{"repositories":["docker_fun"]}
```
Následně se přihlásím pomocí příkazu `docker login 192.168.10.145:5000` s přihlašovacím jménem a heslem `johnny` a konečně provádím stažení dané docker image `docker pull 192.168.10.145:5000/docker_fun`. Po přepnutí do roota se pohybuji ve složce se staženým docker image a provádím příkaz pro nalezení všech podezřelých textových souborů:
```
[root@xforto00 docker]# find . -type f -name "*.txt"
./btrfs/subvolumes/3f657d455d2e1d33ad91ee1b34da895776487208637552e84d7918a0b245f3fe/tmp/secret.txt
./btrfs/subvolumes/5294ab2092419d9ee960f76b9930769d0a6d028b7d0d4ae37de519b55a6e75a0/tmp/secret.txt
./btrfs/subvolumes/cda0ce3ef60ad2978d62c42f0bd4ec15afb1d6ffd6d7edadd07b143e9552d62e/tmp/secret.txt
./btrfs/subvolumes/4eaf246e92753620ae67c30b21ae46df3ae23265e004d4eb97b033421c9d38d5/tmp/secret.txt
```
Dva soubory obsahují tajemství D – zde uvádím to, které má současný datum, druhé tam pravděpodobně zůstalo při tvorbě zadání projektu:
```
# cat ./btrfs/subvolumes/4eaf246e92753620ae67c30b21ae46df3ae23265e004d4eb97b033421c9d38d5/tmp/secret.txt
Tajemstvi D_16-12-20-45-01_e987c8acf8f73bb68001f2f33f3764cd97fd15c1c7228e033e6a1e992064c259
```

# 5 Tajemství E a F

Na vstupním serveru `student@bis.fit.vutbr.cz` analyzuji složku `.ssh` a nacházím soubor `config` s přístupem na `SV2`. Přes SSH se připojuji na `SV2` a na něm objevuji dva podezřelé soubory: `was_es_ist` ve složce `/mnt/root` a `crack_me.zip` ve složce `/trash`.
Při mnoha pokusech o zjištění hesla archivu jsem neúspěšná – dlouho přemýšlím, zda není souvislost mezi archivem a souborem `was_es_ist`, zda nemá heslo nějakou spojitost s příběhem v zadání nebo nějakým německým slovem. Nakonec užívám nástroj `yazc`, u kterého zkouším různé kombinace malých, velkých písmen abecedy a čísel. Tento nástroj má však i přepínač `-s` sloužící pro analýzu hesel se speciálními symboly. Konečně heslo zjišťuji následovně:
```
$ yazc bruteforce -a -A -n -s  crack_me.zip
Password is:  /$
```
Heslo tedy začíná mezerou. Nakonec pomocí jednoduchého Python skriptu rozbaluji archiv a získávám z souboru `geheimnis.txt` tajemství E:
```
Tajemstvi E_16-12-21-15-01_abb737b8d111b9d6502891e316105aa439a7d210e96690554cfdc6e1471fb11d
```

Soubor `was_es_ist` se částečně průběžně mění. Vypsala jsem si více obsahů souborů po různých časových intervalech. Několik znaků je vždy zachováno – toto by mohlo inklinovat k neměnícím se částem obecného vzhledu tajemství, několik znaků se mění v určitém časovém intervalu – toto by mohlo určovat měnící se datum a čas v tajemství a několik znaků je nahodilých – toto by mohl být možný hash, co tajemství následuje. Zpočátku se snažím si vypsat neměnící se znaky, poté ty které se mění v určitém intervalu a z nich sestavit první část tajemství – avšak stále nevidím nějaký vzor, podle kterého bych mohla poskládat i náhodný hash. Tady mě napadá i myšlenka, že vedle zašifrování musí být soubor i nějak zakódovaný a proto si na zkoušku převedu části jako `Tajemstvi` a možný vzhled data a času do jednoho z nejznámějších kódování base64 – výsledek je slibný a začíná se mi potvrzovat domněnka, že by tajemství mohlo být ještě po dešifrování zakódováno tímto způsobem. Po čase stráveném bádáním se snažím i v části tajemství bez hashe nalézt nějaký vzor. Nakonec jsem si seřadila znaky z souboru `was_es_ist` bez hashe následovně (čtení probíhá postupně po řádcích):

| | | | | | |
|---|---|---|---|---|---|
| V | I | M | | | |
| G | p | E | t | j | f |
| F | Z | Z | I | I | J |
| q | H | f | T | t | D |
| Z | d | M | M | M | M |
| W | z | T | t | T | t |
| 1 | k | U | | | |

Vlastní napsaný skript v Pythonu mi znaky přeskládal následujícím způsobem: VGFqZW1zdHZpIEZfMTktMTItMjItMTUtMDJf (`Tajemstvi F_19-12-22-15-02_`). Konečně jsem začala nacházet možný vzor – při čtení po sloupcích střídavým směrem dolů a poté nahoru postupně probíhá daný způsob šifrování – jedná se o známou šifru Zig-Zag (Rail Fence) šifru. Využila jsem nástroj pro dekódování této šifry (`https://www.boxentriq.com/code-breaking/rail-fence-cipher`) a při zadání správného počtu sedmi rails – počet řádků a tedy úrovní dané šifry jsem získala následující kód:
VGFqZW1zdHZpIEZfMTktMTItMjItMTUtMDJfMWFmNDZiNmM2MzUxOGRhNzNjYzZjNTliZTk1ZTM5NTdlMTdkNTYyNDg0MWI3OWIzNjkzNDM2NjQ1Y2QxMTRmYw
Následně tuto sekvenci zkusím dekódovat z base64 a získávám tajemství F:
```
Tajemstvi F_19-12-22-15-02_1af46b6c63518da73cc6c59be95e3957e17d5624841b79b3693436645cd114f
```

## 5.1 Tajemství H, G a I

Při extrakci archivu soubor `geheimnis.txt` neobsahuje pouze tajemství F, ale pod ním i větu:
`Fredek, next time please don't forget your default SSH credentials for the SV1 (192.168.10.150) server`
Na `sv1` je otevřený port pro službu NFS. Mým cílem je provést mount a tak si zobrazit danou složku se soubory. Musím však provést přihlášení na `sv1`, abych zjistila, pro jakou složku se soubory mám přesně mount provést. Uvažuji, že věta v archivu by s vysokou pravděpodobností mohla být nápověda pro přihlašovací údaje a proto se na `sv1` přihlašuji pod loginem `Fredek`. Co se týče hesla, tak zkouším různé možnosti nejvíce užívaných hesel uvedených na stránce `https://nordpass.com/most-common-passwords-list/`. Postupně zkouším nejvíce užívaná a nakonec jsem úspěšně přihlášena při zadání 22. hesla v žebříčku `iloveyou`. Následně procházím `sv1` a nacházím potenciální složku s NFS soubory `/home/shared_dir`, do které se může dostat pouze root a tím se na `sv1` uživatel Fredek nemůže stát. Snažím se o instalaci nástroje `nfs-utils` na svůj studentský server `student@bis.fit.vutbr.cz`, kde mohu získat oprávnění roota. Avšak na tomto serveru není dostupné připojení k internetu, tudíž na instalaci knihovny musím jít složitější cestou.
Na svém školním účtu na serveru `merlin.fit.vutbr.cz` provedu stažení RPM instalačního balíčku pro `nfs-utils` (zvolím verzi pro daný OS, na kterém běží BIS server – Fedora 35). Následně ho pomocí `scp` příkazu zkopíruji na BIS server a zde se snažím vykonat instalaci:

```
$ sudo rpm -i nfs-utils-2.5.4-2.rc3.fc35.x86_64.rpm
error: Failed dependencies:
gssproxy >= 0.7.0-3 is needed by nfs-utils-1:2.5.4-2.rc3.fc35.x86_64
keyutils is needed by nfs-utils-1:2.5.4-2.rc3.fc35.x86_64
quota is needed by nfs-utils-1:2.5.4-2.rc3.fc35.x86_64
rpcbind is needed by nfs-utils-1:2.5.4-2.rc3.fc35.x86_64
```

Provedu na merlinovi stažení závislých balíčků ve formátu RPM stejným způsobem, překopírování na BIS server a jejich následnou instalaci. Řídím se případnými hlášeními o dalších závislých balíčcích a také je nainstaluji. Nakonec po instalaci všech závislostí konečně instaluji nástroj `nfs-utils`. Zakládám si novou složku a na BIS serveru provádím mount NFS serveru:

```
[student@xforto00 ~]$ sudo mount -t nfs 192.168.10.150:/home/shared_dir nfs_mount/
[student@xforto00 ~]$ sudo su
[root@xforto00 student]# cd nfs_mount
```

Ve složce `nfs_mount` se mi zobrazí mnoho obrázků a poté soubory `private.key` a `secret`. Pomocí příkazu `cat` si vypíši do terminálu obsah souboru `secret` a získávám tajemství H:
`Tajemstvi H_21-12-11-15-02_ff317df94b370abe11561064b3240181163e293d4e33062d2f8252bd26357f50`

Soubor `private.key` je PGP klíčem, tento soubor obsahuje i passphrase. Již při připojení k FTP serveru na `sv4` jsem objevila vedle obrázku vedoucí k tajemství J i soubor `secret.asc` s PGP zprávou. S nadějí, že jsem konečně našla vše potřebné k rozluštění této PGP zprávy nahrávám klíč, passphrase a zprávu do nástroje pro dekódování PGP zpráv ((`https://8gwifi.org/pgpencdec.jsp`) a získávám tajemství G:
`Tajemstvi G_05-12-21-15-01_1bec1c17eeafddcf026d9dd5c99bee13dfc927c633e2b43276491a05d74e4844`

Zobrazím si podrobnosti o obrázcích umístěných na NFS serveru. Jeden z nich – `112302.jpg` – se průběžně aktualizuje stejně jako obecné vlastnosti již získaných tajemství. Snažím se v tomto obrázku najít nějaké skryté podezřelé řetězce – používám nástroj `strings` a nastavenou minimální délkou řetězce:

```
$ strings -92 112302.jpg
]Ahqltzacp P_21-12-13-45-02_3hm7811ji0hklkj32k9i0i80j9i60086i67im0j1828l27h69km9i15271im4h5k
```

I když mě trochu zmate symbol závorky na začátku, tak přemýšlím, jakým způsobem je tajemství zašifrované. Už teď je skoro jisté, že se o nějakou šifru bude jednat, zvláště díky struktuře řetězce. Použiji nástroj pro hledání všech možných dekódování pomocí Caesarovy šifry (`https://www.dcode.fr/caesar-cipher`) a jeden možný výsledek má následující podobu:
`]Tajemstvi I_21-12-13-45-02_3af7811cb0adedc32d9b0b80c9b60086b67bf0c1828e27a69df9b15271bf4a5d`
Získávám tedy tajemství I:
`Tajemstvi I_21-12-13-45-02_3af7811cb0adedc32d9b0b80c9b60086b67bf0c1828e27a69df9b15271bf4a5d`