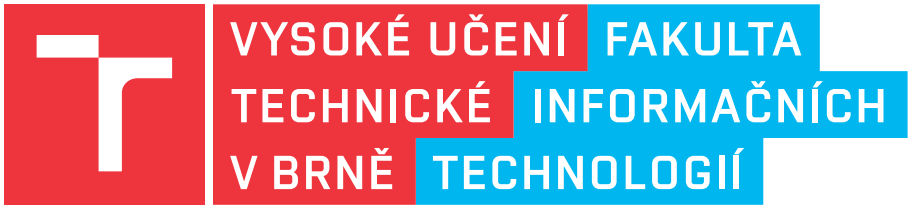
VYSOKÉ UČENIE TECHNICKÉ V BRNĚ
FAKULTA INFORMAČNÝCH TECHNOLOGIÍ



Projekt – BIS 2024

Autor: Michal Ľaš
Kontakt: xlasmi00@stud.fit.vutbr.cz

8. decembra 2024

Bezpečnost informačných systémov

1 Mapovanie siete

Pre mapovanie siete som použil nástroj `nmap`, konkrétne príkaz:

```
nmap -sV -p- --open --exclude 10.89.1.1 10.89.1.0/24
```

Okrem študentských serverov som našiel štyri servery, ktoré sú popísané v tabuľke 1. Následne som ešte pre každý server použil osobitné mapovanie pomocou príkazu `nmap -p- -A <server_ip>`. Na žiadnom serveri som však nenašiel ďalšie bežiace služby. V tejto fáze som sa dozvedel iba názvy HTTP serverov. K ďalším názvom som sa dostal neskôr pri hľadaní tajomstiev.

Názov	IP adresa	Služby
Rivendell	10.89.1.156	SSH server (OpenSSH 9.2p1) – port 22/tcp OS Linux: cpe:/o:linux:linux_kernel
Isengard: The Iron Fortress	10.89.1.157	HTTP server (Apache httpd 2.4.62) – port 80/tcp
Edoras	10.89.1.158	FTP server (vsftpd 2.0.8) – port 21/tcp
Mirkwood	10.89.1.159	HTTP server (Apache httpd 2.4.62) – port 80/tcp

Tabuľka 1: Zmapované servery

2 Zraniteľnosti

Isengard: The Iron Fortress (tajomstvo A)

Jedná sa o HTTP server, takže ako prvé som urobil HTTP request na tento server pomocou nástroja `curl`. Ako odpoveď som dostal nápovedu, že `/cgi-bin/gate` nie je zabezpečená. Zistil som, že je možné vykonať **shellshock attack**, na ktorý som použil príkaz:

```
\verb|curl -v "http://10.89.1.157/cgi-bin/gate" -H 'User-Agent: () { :; }; echo; /bin/bash
    -c "whoami"'|
```

Som sa dostal k terminálu serveru. Prepínač `-H` umožňuje špecifikovať vlastnú HTTP hlavičku. V tomto prípade som použil User-Agent hlavičku, ktorá by normálne identifikovala klienta. Ďalej som podstrčil definíciu prázdnej funkcie a za ňu som dal príkaz, ktorý sa kvôli shellshock bugu taktiež vykoná v termináli serveru a vráti mi odpoveď.

Následne som sa rozhodol preskúmať server a tajomstvo A som našiel v `/etc/shadow`, ktoré malo zle nastavené prístupové práva, tak že ho mohli čítať všetci užívatelia.

Zraniteľnosťou bol CGI skript, cez ktorý bolo možné vykonať **shellshock attack**. Možné zabezpečenie proti shellshock:

- Pravidelné aktualizácie Bash shell.
- Validácia vstupu od užívateľa.
- Pri použití CGI skriptov zbezpečiť aby sa vykonali v sendboxe a nespoliehali sa na shell príkazy.

Mirkwood (tajomstvo C, D)

Opäť sa jedná o HTTP server, takže som postupoval ako v predošlom prípade a dostal som nápovedu, na 2 odkazy `authentication.html` a `upload.html`.

V prípade `authentication.html` som zistil, že autentifikácie prebieha cez databázu SQLite3. Skúsil som útok pomocou SQL injection. Použil som príkaz:

```
curl -v -X POST http://10.89.1.159/authenticate.php -d "username='OR''='\&password='OR''='"
```

ktorý mi odhalil existenciu súboru `list.php`. Následne som skúšal niekoľko príkazov, ktorými som sa snažil zobraziť si jednotlivé tabuľky databázy. K tým som sa nakoniec dostal pomocou príkazu:

```
curl -v "http://10.89.1.159/list.php?id=1"\%20OR\%201=1\%20UNION\%20SELECT\%201,\%20name
    ,\%20sql\%20FROM\%20sqlite_master""
```

A ďalším príkazom som si vypísal obsah tabuľky `users`, v ktorej boli mená a heslá užívateľov. Jedno z hesiel bolo aj tajomstvo D.

```
curl -v "http://10.89.1.159/list.php?id=1"%20OR%201=1%20UNION%20SELECT%201,%20username,%20
    password%20FROM%20users""
```

Zraniteľnosťou v tomto prípade bola možnosť vykonať **SQL injection** útok na serverovú databázu. Možná prevencia proti SQL injection:

- Používanie parametrizovaných queries
- Validácia vstupu
- Nepovoliť všetkým užívateľom prístup ku kritickým príkazom ako `DROP`, a pod

Ďalej som pracoval so súborom `upload.html`. Tento súbor obsahuje formulár pre nahrávanie obrázkov na server. Ako prvé mi napadlo do obrázku prepašovať nejaký kód, ktorý by som mohol neskôr spustiť. Podarilo sa mi nahrať obrázok so spustiteľným PHP kódom, avšak nepodarilo sa mi nájsť miesto kde sa obrázky ukladajú. Nakoniec som sa mi podaril úspešný útok keď som na začiatok PHP kód pridal byty charakteristické pre JPG hlavičku (`\xFF\xD8\xFF\xE0`) a premenoval súbor na `payload.jpg.php`. V tomto formáte sa už kód spustil a odhalil mi tajomstvo C.

Zraniteľnosťou v tomto prípade je nedostatočná kontrola vložených súborov. Ako zabezpečenie je vhodná lepšia kontrola vstupu. Nekontrolovať len hlavičku a príponu, ale aj obsah a nedovoliť aby sa obsah mohol spustiť napríklad ukladať súbory do priečinku bez práv pre ich spustenie.

Edoras (tajomstvo B)

Jedná sa o FTP server, na ktorý je nutné vedieť prihlasovacie údaje. Napadlo mi vyskúšať prihlasovacie údaje, ktoré som našiel v tabuľke `users` na serveri Mirkwood. Bol som úspešný prihlásil som sa ako `admin` s heslom `iloveyou`. Následne som sa rozhodol preskúmať server. V zložke `/home` som našiel priečinok `theoden`, ktorého názov pravdepodobne odkazuje na sériu LoTR. Tento priečinok však nebol prístupný pre užívateľa `admin`. Stiahol som si súboru `/etc/shadow` pomocou príkazu:

```
curl ftp://10.89.1.158//etc/shadow --user admin:iloveyou -o ftp/shadow|
```

Z neho som zistil, že užívatelia `admin` a `theoden` majú rovnaké heslo. Prihlásil som sa teda ako používateľ `theoden` a dostal som sa do priečinku `/home/theoden`, kde som našiel súbor `secret.txt` s tajomstvom B.

Zraniteľnosť v tomto prípade bola SQL injection z predchádzajúceho serveru a použitie rovnakého hesla a **soli (salt)** pre heslá dvoch užívateľov. Zároveň ako zraniteľnosť možno považovať voľný prístup do `/etc/shadow`. Prevenciou je určite používanie silných hesiel, náhodnej soli pri hešovaní hesiel a obmedzenie prístupu do `/etc/shadow`.

Rivendell (tajomstvo E)

Jedná sa o SSH server a nápoveda bola na riešiteľskej stanici v súbore `~/.ssh/config`. V priečinku `~/.ssh` bol RSA kľúč a do `~/.ssh/config` stačilo pridať IP adresu serveru Rivendell. Následne sa bolo možné pripojiť pomocou príkazu `ssh rivendell` na tento server. Rozhodol som sa preskúmať tento server a našiel som súbor `/shrine/chest.img`. Tento súbor som si stiahol k sebe a zistil som pomocou nástroja `binwalk`[1], že sa jedná o šifrovaný `.zip` archív, ktorý obsahuje XML súbor. Archív som musel ešte opraviť pomocou `zip -FF`. XML súbory majú pevne danú štruktúru, kde hlavička začína ako `<?xml version="1.0"` a vďaka týmto pár bytom je možný **known-plaintext-attack**. Ten som vykonal pomocou nástroja `bkcrack`[2] a našiel som tajomstvo E.

Zraniteľnosť bola v tom, že bol šifrovaný súbor s predvídateľným obsahom a teda bolo možné na ňom vykonať known-plaintext-attack. Ako prevenciu je určite vhodné používať silné šifrovanie, používanie náhodných inicializačných vektorov a určite sa vyhnúť šifrovaniu pomocou ECB (Electronic Codebook).

[1] https://linuxcommandlibrary.com/man/binwalk
[2] https://github.com/kimci86/bkcrack