UDC: 659.2:004 INFO M: str. 13-17

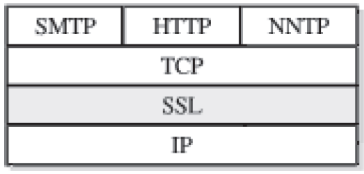
IMPLEMENTACIJA SSL BEZBEDNOSNIH PROTOKOLA NA APACHE WEB-SERVERU
IMPLEMENTING SSL SECURITY PROTOCOLS ON THE APACHE WEB-SERVER

Milan Jandrić, Snežana Vulović, Slobodan Jovanović
Univerzitet Metropolitan, Beograd, Srbija

REZIME: Secure Sockets Layer (SSL) je Web protokol za enkripciju i autentifikaciju baziran na sesiji. Web-server Apache podržava SSL i TLS (Transport Layer Security), pomoću modula mod_ssl. Ovaj rad razmatra implementaciju SSL bezbednosnih protokola na Apache Web-serveru. Takodje objašnjava kako SSL protokoli funkcionišu, i kako se koriste SSL protokoli. Opisuje se instalisanje i konfigurisanje modula mod_ssl u okviru Apache servera. Konačno, članak prezentuje rezultate bezbednosnih testova na Apache server konfiguraciji, pomoću Nessus softvera. Nessus softverski paket je sveobuhvatan program za skeniranje ranjivosti računarskih sistema. Besplatan je za ličnu upotreba u ne-poslovnom okruženju. Cilj programa je da detektuje potencialne ranjivosti računarskih sistema. Na primer, ranjivosti koje bi dopustile hakerima udaljenu kontrolu sistema i pristup osetljivim podacima.
KLJUČNE REČI: Nessus softver, HTTP protokol, Šifrovanje
ABSTRACT: SSL (Secure Sockets Layer) is a Web protocol for encrypting and autentification based on a session. The Apache Web-server supports SSL and TLS (Transport Layer Security), by using module mod_ssl. This paper discusses the implementation of SSL protocols on the Apache Web-server. Also, it explains how SSL protocols work, and how to use SSL protocols. Installing and configuring the mod_ssl module within the Apache server is also discussed. Finally, the paper describes the results of security tests of the Apache server configuration, by using the Nessus software. In computer security, Nessus is a comprehensive vulnerability scanning program, free of charge for personal use in in a non-enterprise environment. Its goal is to detect potential vulnearibilities on the tested sxstems, for example, vulnerabilties that allow a remote cracker to control or access sensitive data on a system.
KEY WORDS: Nessus software, HTTP protocol, Encrypting

1. UVOD

Secure Sockets Layer (SSL) je Web protokol za enkripciju i autentifikaciju baziran na sesiji [1-3]. Osigurava siguran komunikacioni kanal između dvoje učesnika. SSL omogućava autentifikaciju servera i opciono klijenta, kako bi sprečio prisluškivanje, manipulisanje i falsifikovanje poruka u klijent-server aplikaciji. Uspostavljanjem zajedničke tajne između dva korisnika, SSL obezbeđuje privatnost [4-6]. SSL radi na transportnom sloju mrežnog OSI (*Open Systems Interconnection*) sistema (ispood aplikativnog sloja), i nezavisan je od protokola koji aplikacija koristi. Stoga aplikativni protokoliki (HTTP, FTP, TELNET, itd.) mogu transparentno da rade iznad SSL protokola. *The* OSI model predstavlja način pod-podele komunikacionog sistema u manje delove zvane „slojevi“. Određeni komunikacioni sloj pruža usluge sloju koji se u modelu nalazi iznad njega, a koristi usluge sloja koji se nalazi ispod. Kao što je prikazano na slici 1. TCP protokol je jedan od osnovnih iz skupa Internet protokola i operiše na sloju 4 (transportni sloj). IP je takođe jedan od osnovnih protokola iz skupa internet protokola i operiše na sloju 3 (mrežnom sloju), koji definiše kako aplikacije mogu ući na mrežu. HTTP je mrežni protokol aplikacionog sloja izgrađen na vrh TCP-a, i operiše na OSI sloju 7 (aplikacija). SSL je standard za enkriptovanu klijent/server komunikaciju između mrežnih sprava. Mrežni protokol SSL radi iznad TCP/IP sloja i operiše na OSI sloju 6 (prezentacioni sloj). Osnova World Wide Weba je protokol za prenos hiperteksta (eng. HyperText Transfer Protocol, HTTP). TCP je protokol koji se koristi za prenos i povezivanje dokumenata koji sadrže raznovrsne tipove medija, kao što su tekst, slike, zvuk, animacije i videozapisi. HTTP se u priličnoj meri ponaša kao i drugi klijentsko serverski protokoli koji se koriste na Internetu, kao što su SMTP (za elektronsku poštu ili FTP (za prenos datoteka). Netscape je izvorno razvio SSL 1994. godine. Od tada, SSL je postao široko prihvaćen i sada se koristi i podržan je u svim većim web pretraživačima, kao i raznim drugim softverskim i hardverskim proizvodima.



Slika 1: – *Pozicija SSL protokola u odnosu na mrežni model*

Ovaj protokol trenutno dolazi u tri verzije : SSLv2, SSLv3 i TLSv1 (također poznat kao SSLv3.1). SSLv3 je rešio mnoge nedostatke iz verzije SSLv2. Trenutno, IETF standardizuje SSLv3 u Transport Layer Security (TLS). TLSv1 je veoma slična SSLv3, sa samo manjim izmenama u protokolu. Prvo oficialno izdanje TLS-a je bilo 1999. g. Najpoznatija upotreba SSL protokola je za enkripciju sadržaja preko HTTP protokola. Hypertext Transfer Protocol Secure (HTTPS) je kombinacija Hypertext Transfer protokola sa SSL/TLS protokolom, radi pružanja šifrovane komunikacije i sigurne identifikacije mrežnog web servera. HTTPS konekcije se često koriste za

novčane transakcije na WWW i za osetljive transakcije u korporativnim informacionim sistemima.

Web server je softver koji je konfigurisan tako da odgovori na HTTP zahteve [7]. Apache HTTP server, poznat kao „Apache“, je vrlo često korišćen Web-server softver. Obično, Apache je instalisan na Linux ili sličnim operativnim sistemima, i to je *open-source* (besplatan) softver. Procenjuje se da Apache je instalisan na oko 60% Websajtova na Web-u. Apache omogućuje puno različitih opcija, i one se obično implementiraju kao kompajlirani moduli. Npr. Apache podržava SSL i TSL (Transport Layer Security), pomoću modula mod_ssl.

Nessus je softver koji je na svetu najviše korišćen za procenu ranjivosti neke računarske konfiguracije. Verzija Nessus 5.o se odlikuje velikom brzinom, pretragom osetljivih podataka, i analizom ranjivosti bezbednosti sistema. Nessus je iskorišćen u mnogim kompanijama da poveća korisnost, efektivnost, efikasnost, i komunikativnost računarskih sistema [8].

U oblasti komjuterske bezbednosti, ranjivost (*vulnerability*) je slabost koja dozvoljava nekom napadaču (*attacker)* da snizi informacionu bezbednost sistema. Da bi iskoristio ranjivost, napadač mora da ima neki alat ili tehniku koju može da poveže sa slabošću sistema. Analizator tj. „skener“ ranjivosti (*vulnerability scanner*) je softver koji dizajnran da oceni slabosti (tj. „probojnost“) kompjuterske sisteme i kompjuterske mreže i kompjuterske aplikacije. Termini informaciona bezbednost (*information security*) ili kompjuterska bezbednost (*computer security*) ili informaciono osiguranje (*information assurance*) se često koriste kao slični termini [8].

Ovaj članak diskutuje SSL bezbednosne Web-protokole, i njihovu implementaciju na Apache Web-serveru. Takodje, razmatra se testiranje bezbednsti tj. test probojnosti konfiguracije Appache servera, pomoću Nessus softvera. Opisuje se i instalisanje i konfigurisanje modula mod_ssl u okviru Apache servera.

2. SSL WEB-PROTOKOLI

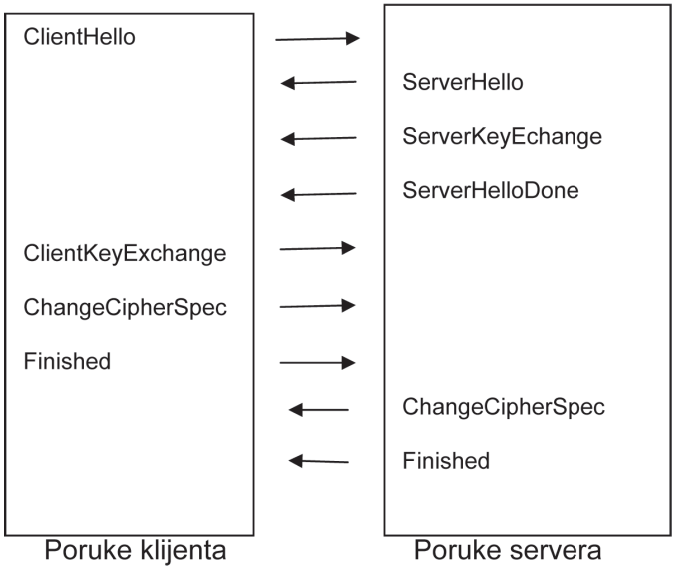
Glavna funkcija SSL protokola je da obezbedi klijentu i serveru da uspostave siguran kanal komunikacije [5]. Secure Sockets Layer (SSL) protokol definiše dve vrlo različite uloge za klijenta i za server. Klijent inicira sigurnu komunikaciju, a server odgovara na zahteve klijenta. Npr. kod upotrebe SSL-a kod Web pretraživanja, Web-pretraživač je SSL klijent, a Web-sajt je SSL server. Pošto klijent inicira komunikaciju, on predloži serveru niz SSL opcija, a server odabere one SSL opcije koje će se koristiti u daljoj komunikaciji.

Kada klijent i server komuniciraju upotrebom SSL-a, oni razmenjuju SSL poruke. Pri tome, SSL definiše različite nivoe poruka. Tabela 1 prikazuje SSL poruke na svim nivoima SSL protokola, abecednim redom. Na slici 2 prikazana je razmena poruka kod komuniciranja izmedju klijenta i servera primenom SSL-a, a tabela 2 opsuje korake kod šifrovane komunikacije [5].

Klijentova poruka ClientHello poruka započinje SSL komunikaciju. Tabela 3 prikazuje pojedine komponente ClientHello poruke. Server pravi konačnu odluku o tome koji će kriptografski servisi biti iskorišćeni za komunikaciju, ali je ograničen na ovu listu. Po prijemu ClientHello poruke, server odgovora sa ServerHello. Sadržaj ServerHello je gotovo isti kao i ClientHello. *Npr.* Version polje sadržano u ClientHello poruci specificira verzije SSL koje može da podrži, dok isto polje u ServerHollo poruci odlučuje koja verzija SSL će biti korišćena.

Tabela 1: – *SSL poruke*

Poruka	Opis
Alert	Obaveštava drugu stranku o mogućem proboju sigurnosti Ili o neuspeloj komunikaciji
ApplicationData	Stvarna informacija koju su dve stranke razmenile, koja je šifrovana, autentifikovana, i/ili verifikovana od strane SSL-a
Certificate	Poruka koja nosi pošiljaočev sertifikat javnog ključa
CertificateRequest	Zahtev servera da klijent pošaljesvoj sertifikat javnog ključa
CertificateVerify	Poruka od klijenta koja verifikuje da zna privatni ključ koji odgovara na svoj sertifikat javnog ključa
ChangeCipherSpec	Indikacija za početak korišćenja dogovorenih siguronosnih servisa (poput enkripcije)
ClientHello	Indikacija klijenta koje siguronosne usluge želi i može podržati
ClientKeyExchange	Poruka od klijenta koja nosi kriptografski ključ za komunikaciju
Finished	Indikacija da su svi inicijalni pregovori završeni i da je sigurna ostvarena komunikacija
HelloRequest	Zahtev servera da klijent počne (ili opet počne) proces SSL pregovora
ServerHello	Poruka od servera koja označava siguronosne usluge koje će biti korišćene za komunikaciju
ServerHelloDone	Indikacija od servera da je završio sve zahteve od klijenta za uspostavljanje komunikacije
ServerKeyExchange	Poruka od servera sa kriptografskim ključevima za komunikaciju



Slika 2: – *SSL koristi 9 poruka da ostvari šifrovanu komunikaciju*

Tabela 2: – Pregovor o šifrovanoj komunikaciji

Korak	Akcija
1	Klijent šalje *ClientHello* poruku, predlažući SSL opcije.
2	Server odgovara sa *ServerHello* porukom, birajući SSL opcije.
3	Server šalje svoj javni ključ u *ServerKeyExchange* poruci.
4	Server zaključuje svoj deo pregovaranja sa *ServerHello-Done* porukom.
5	Klijent šalje sesijski ključ (enkriptovan serverovim javnim ključem) u *ClientKeyExchange* poruci.
6	Klijent šalje *ChangeCipherSpec* poruku da aktivira sve dogovorene opcije za sve poruke koje će slati.
7	Klijent šalje *Finished* poruku da dopusti serveru da proveri nove aktivirane opcije.
8	Server šalje *ChangeCipherSpec* poruku da aktivira dogovorene opcije za sve buduće poruke koje će poslati.
9	Server šalje *Finished* poruku da dopusti klijentu da proveri aktivirane opcije.

Tabela 3: – Važne komponente ClientHello poruke.

Polje	Upotreba
Version	Identifikuje najvišu verziju SSL protokola koju klijent može podržati.
RandomNumber	32-bitni nasumični broj koršćen da prilikom kriptografskih kalkulacija.
SessionID	Identifikuje specifičnu SSL sesiju.
CipherSuites	Lista kriptografskih parametara koju klijent može da podrži.
CompressionMethods	Identifikuje metod kompresije podataka koje klijent može da podrži.

3. APACHE WEB-SERVER

Zastupljenost Apache-a na tržištu je dvostruko veća od sledećeg konkurenta u redu, Microsoft-a. Do ovakve situacije nije dovela samo činjenica da je Apache *freeware*, te je njegova distribucija besplatna, već i to što je kod otvorenog tipa (*open source*), što svim korisnicima omogućava uvid u isti. Upravo zbog ove stalne kontrole koju sprovode hiljade korisnika, pouzdanost rada Apache-a daleko je veća od komercijalnih softvera kod kojih je brojnost i perceptivnost kontrolora ograničena unutarnjom strukturom tima koji je razvijao softver. Značaj ovoga postaje posve jasan ako se sve sagleda iz ugla sigurnosti podataka, gde naočigled trivijalne greške mogu dovesti do zastrašujućih posledica.

Instalacija Apache Web-servera je veoma jednostavna. Sastoji se od sledecih koraka:

- Pokrenemo konzolu
- Prebacimo se u root mod komandom su

```
su
```

- Pomocu yum-paketa downloadujemo i instaliramo Apache

```
yum install httpd
```

- Startujemo Apache HTTP server (httpd)

```
/etc/init.d/httpd start
```

- Podesimo Linux da automatski startuje Apache HTTP server (httpd) pri sledecem startovanju operativnog sistema.

```
chkconfig --levels 235 httpd on
```

- U Web pretrazivacu kucamo http://localhost da proverimo da li je Apache pokrenut.
- Podešavamo FireWall da omogući konekciju na port 80 drugim računarima u mreži.

```
nano -w /etc/sysconfig/iptables
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 80 -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 443 -j ACCEPT
```

- Testiramo konekciju sa drugog racunara na mreži. http://192.168.1.101

4. PODEŠAVANJE MOD_SSL

Većina Web pretraživača i komjuterskih sistema, koji podržavaju SSL, imaju listu sertifikacionih tela kojima automatski veruju. Ako Web pretaživač naiđe na sertifikat koji je autorizovan od strane sertifikaciong tela koji nije u pomenutoj listi, Web pretraživač zahteva od korisnika da ili prihvati sertifikat ili ga odbije. Takođe, druge aplikacije čak mogu da generišu grešku i odbiju da nastave da rade kada je upitanju sertifikat koji nije potpisan od sertifikaciong tela.

Proces dobijanja sertifikata od sertifikaciong tela je jednostavan i sastoji je od sledećih koraka:

- Buduci da Apache ne dolazi sa predhodno instaliranom mod_ssl bibliotekom, potrebno je da je instaliramo
- Kreiramo privatni i javni kriptografski ključ
- Na osnovu javnog ključa kreiramo zahtev za sertifikat. Ovaj zahtev sadrži informacije o serveru.
- Šaljemo zahtev za sertifikat, zajedno sa dokumentima koji garantuju naš identitet sertifikacionom telu.
- Kada sertifikaciono telo ustanovi da smo mi stvarno oni za koje se predstavljamo, šalju nam digitalni sertifikat.
- Instaliramo sertifikat na siguran web server, i konfigurišemo web server da ga koristi.

Instalacija mod_sll biblioteke na Fedora-linux distribuciji je maksimalno pojednostavljena korišćenjem yum paketa:

```
yum install mod_ssl*
```

Da bi smo generisali zahtev za sertifikat koristimo *opensource* program **openssl**.

Prvo generišemo privatni ključ:

```
openssl genrsa -des3 –out server.key 1024
```

Kreiranje zahteva za sertifikat

```
openssl req –new –kez server.key –out server.csr
```

Generisani server.csr fajl šaljemo sertifikacinom telu.

Ako želimo da kreiramo sertifikat koji ćemo sami i da potpišemo koristimo opciju

```
openssl x509 –req –days 365 –in server.csr –signkey server.key –out server.crt
```

Da bismo instalirali sertifikate kopiramo ih u podrazumevani direktorijum

```
cp server.crt /etc/pki/tls/certs/
cp server.key /etc/pki/tls/private/
```

Ostaje nam jos samo da konfigurišemo apache da koristi sertifikate koje smo upravo instalirali menjajuci direktive SSLCertificateFile i SSLCertificateKeyFile.

Nakon aktiviranja modula moram da restartujemo apache server

```
/etc/init.d/ httpd restart
```

5. PROBOJNO TESTITRANJE KONFIGURACIJE

Pre početka akcija u smislu ojačavanja servera, odlučeno je da se sprovede probojno testiranje, kako bi se utvrdile slabe tačke i identifikovale akcije koje treba izvršiti kako bi se server mogao eksploatisati u najsigurnijem modu. Za probojno testiranje korišćen je Nessus softverski paket, koji služi za otkrivanje i testiranje ranjivosti sistema. Nessus softverski paket je sveobuhvatan program za skeniranje računarskih sistema. Besplatan je za ličnu upotreba u ne-poslovnom okruženju. Cilj programa je da detektuje potencialne ranjivosti računarskih sistema. Na primer:

- Ranjivosti koje bi dopustile hakerima udaljenu kontrolu sistema i pristup osetljivim podacima
- Nepravilno konfigurisane sisteme (otvoren mail server, neupdate-ovan softver...)
- Podrazumevane šifre (root, admin...) Nessus takođe može da koristi Hydra (eksterni program) radi pokušaja proboja sistema pomoću rečnika često korišćenih šifri
- Pripremanje za PCI DSS proveru

Prema istraživanju organizacije *sectools.org*, softver Nessus je nekoliko prošlih godina bio proglašen za najpopularniji skener ranjivosti sistema. Procenjuje se da ga koristi preko 75000 organizacija iz celog sveta. Pri tipičnoj upotrebi, Nessus počinje testiranje sistema skeniranjem svih otvorenih portova na ciljnom računaru tj. računarima. Za skeniranje koristi svoj interni skener, ali može da koristi i Amap ili Nmap. Kada naiđe na otvorene portove pokušava raznim testovima da pronađe sve eventualne ranjivosti. Testovi su napisani koristeći NASL (Nessus Attack Scripting Language), skriptni programski jezik, specialno optimizovan za mrežne interakcije. Kompanija Tenable Networ Security koja je proizvela Nessus, proizvodi nekoliko desetina novih testova („pluginova“) svake nedelje. Ovi testovi su dostupni besplatno za privatnu upotrebu. Dok za profesionalnu upotrebu postoje i dodatne skripte.

Rezultati skeniranja mogu biti predstavljeni u brojnim formatima. Neki od njih uključuju, običan tekst, XML, HTML i LaTeX. Na Unix sistemima skeniranje može biti automatizovano pomoću klijenta koji koristi komandnu liniju. Nessus „policy“ se sastoji od konfiguracionih opcija povezanih za skeniranjem. Ove opcije uključuju, ali nisu ograničene na:

- Parametre koji kontrolišu aspekte skeniranja kao što su vreme, proj računara, tips skenera portova itd.
- Pristupne podatke za servise (SSH, Oracle Database, HTTP, FTP, POP, IMAP ili Kerberos autentifikaciju).

Pri instalaciji Nessus dolazi sa 4 predefinisane konfiguracije:

- External Network Scan
- Internal Network Scan
- Web App Tests
- Prepare for PCI DSS audits.

Pri testiranju severa korišćen je Internal Network Scan, koji je prilagođen za skeniranje sistema koji se nalaze u intranetu zajedno sa Nessus skenerom. Izmenjen je profil tako da koristi sve skirpte za proboj sistema koje dolaze u verziji Nessus skenera za ličnu upotrebu. Slede rezultati probojnog testiranja (Tabela 4).

Tabela 4: – *Rezultat testiranja*

PLUGIN ID#	#	PLUGIN NAME	SEVERITY
11213	2	HTTP TRACE / TRACK Methods Allowed	Medium
51192	1	SSL Certificate signed with an unknown Certificate Authority	Medium
22964	3	Service Detection	Low
43111	2	HTTP Methods Allowed (per directory)	Low
24260	2	HyperText Transfer Protocol (HTTP) Information	Low
20108	2	Web Server / Application favicon.ico Vendor Fingerprinting	Low
10107	2	HTTP Server Type and Version	Low
54615	1	Device Type	Low
51891	1	SSL Session Resume Supported	Low
45590	1	Common Platform Enumeration (CPE)	Low
35716	1	Ethernet Card Manufacturer Detection	Low
25220	1	TCP/IP Timestamps Supported	Low
21643	1	SSL Cipher Suites Supported	Low
19506	1	Nessus Scan Information	Low
11936	1	OS Identification	Low
11197	1	Multiple Ethernet Driver Frame Padding Information Disclosure (Etherleak)	Low
10863	1	SSL Certificate Information	Low
10287	1	Traceroute Information	Low
10114	1	ICMP Timestamp Request Remote Date Disclosure	Low

Osim prva dva testa sa ID brojevima 11213 i 51192, svi ostali problemi su označeni niskom značajnošću za sigurnost, ili su u pitanju manje važne informacije ili informacije o postojanju nekih usluga. Takođe je sprovedeno i skeniranje svih portova na serveru. Svi portovi osim porta 80 (http) i porta (443) su bili blokirani.

Problem 11213 se jednostavno rešava korišćenjem posebne direktive (TraceEnable off) u httpd.conf fajlu. A problem 51192 je očekivan, jer prilikom testiranja nije bilo mogućnosti da se nabavi sertifikat potpisan i izdat od strane zvaničnog

Sertifikaciong Tela. Iako su testovi pokazali da Fedora pri standardnoj instalaciji dolazi zadovoljavajuće zaštićena, i da prilikom testiranja nisu pronađeni veliki sigurnosni propusti, odlučeno je da se provere sve servise koje Fedora pri standardnoj instalaciji pokreće prilikom startovanja sistema.

Listing svih aktivnih servisa dobijamo komandom *chkconfig*:

```
chkconfig --list | grep 3:on
abrt-ccpp       0:off  1:off  2:off  3:on  4:off  5:on  6:off
abrt-oops       0:off  1:off  2:off  3:on  4:off  5:on  6:off
auditd          0:off  1:off  2:on   3:on  4:on   5:on  6:off
cpuspeed        0:off  1:on   2:on   3:on  4:on   5:on  6:off
cups            0:off  1:off  2:on   3:on  4:on   5:on  6:off
httpd           0:off  1:off  2:on   3:on  4:off  5:on  6:off
ip6tables       0:off  1:off  2:on   3:on  4:on   5:on  6:off
iptables        0:off  1:off  2:on   3:on  4:on   5:on  6:off
iscsi           0:off  1:off  2:off  3:on  4:on   5:on  6:off
iscsid          0:off  1:off  2:off  3:on  4:on   5:on  6:off
livesys         0:off  1:off  2:off  3:on  4:on   5:on  6:off
livesys-late    0:off  1:off  2:off  3:on  4:on   5:on  6:off
lldpad          0:off  1:off  2:on   3:on  4:on   5:on  6:off
lvm2-monitor    0:off  1:on   2:on   3:on  4:on   5:on  6:off
mdmonitor       0:off  1:off  2:on   3:on  4:on   5:on  6:off
netfs           0:off  1:off  2:off  3:on  4:on   5:on  6:off
portreserve     0:off  1:off  2:on   3:on  4:on   5:on  6:off
sendmail        0:off  1:off  2:on   3:on  4:on   5:on  6:off
```

Posle gašenja svih nepotrebnih servisa izlaz iste komande je:

```
abrt-ccpp       0:off  1:off  2:off  3:on  4:off  5:on  6:off
abrt-oops       0:off  1:off  2:off  3:on  4:off  5:on  6:off
auditd          0:off  1:off  2:on   3:on  4:on   5:on  6:off
httpd           0:off  1:off  2:on   3:on  4:off  5:on  6:off
iptables        0:off  1:off  2:on   3:on  4:on   5:on  6:off
lldpad          0:off  1:off  2:on   3:on  4:on   5:on  6:off
lvm2-monitor    0:off  1:on   2:on   3:on  4:on   5:on  6:off
portreserve     0:off  1:off  2:on   3:on  4:on   5:on  6:off
```

6. ZAKLJUČAK

Secure Sockets Layer (SSL) je Web protokol za enkripciju i autentifikaciju baziran na sesiji . Ovaj rad razmatra implementaciju SSL *(Secure Sockets Layer*) bezbednosnih protokola na Appache Web-serveru. Takodje objašnjava kako SSL protokoli funkcionišu, i kako se koriste SSL protokoli. Opisuje se instalisanje i konfigurisanje modula mod_ssl u okviru Appache servera. Konačno, članak prezentuje rezultate bezbednosnih testova na Appache server konfiguraciji, pomoću Nessus softvera.

Danas u vreme sve šireg povezivanja (Internet i Intranet), nastaje sve veća potreba za sigurnošću. Ipak, iz iskustva na nizu projekata vezanih za Web aplikacije, došlo se do zaključka, da najveći faktor i sigurnosni rizik ne proizilazi iz ograničenosti tehničkih rešenja, nego uglavnom od neinformisanosti i neznanja samih korisnika. Npr., na jednom projektu, iskorišćene su se sve tehnike osiguravanja Web-sajta, od provere ulaznih podataka pomoću raznih alata za sprečavanja XXS napada, i korišćen je TLS za osiguravanje komunikacije između korisnika. Medjutim, posle mesec dana, i ostvarenog značajnog prometa osetljivih informacija, šifra za pristup administratorskom delu aplikacije je i dalje ostala nepromenjena (npr. test-test), što ukazuje da je ljudski faktor najveći faktor nesigurnosti.

ZAHVALNICA:

Ovaj rad podržan je od strane Ministarstva za nauku i obrazovanje (Projekat III44006).

LITERATURA:

[1] *Guide to SSL VPNs, Recommendations of the National Institute of Standards and Technology*, Special Publication 800-113, Jul 2008.
[2] W. Mao, *Modern Cryptography: Theory and Practice*, Prentice Hall PTR, 2003.
[3] D. Calloway, An Introduction to Cryptography: *PGP Press, 2009.*
[4] B. Schneier, Applied Cryptography, Second Edition - John Wiley & Sons, 2010.
[5] S. A. Thomas, SSL & TLS Essentials Securing the Web, John Wiley & Sons, Inc. 2000.
[6] S. Burnett, S. Paine, RSA Security's Official Guide to Cryptography, The McGraw--Hill Companies, 2001.
[7] K. Coar, R. Bowen, Apache Cookbook, O'Reilly, 2009.
[8] www.Wikipedia.org

Milan Jandrić, milanjandric@gmail.com
Inženjer informacionih tehnologija, diplomirao je 2011. g. na Univerzitetu Metropolitan, Fakultetu za informacione tehnologije. Od 2008.g. bavi se razvojem Web-aplikacija, kao i bezbednošću kompjuterskih sistema. Koristi programske jezike Java, C#, PHP, JavaScript. Radio je na nizu projekata za domaće i inostrane klijente, na razvoju i implementaciji i bezbednosti Web-aplikacija i računarskih sistema.

Snežana Vulović, snezana.vulovic@metropolitan,ac.rs
Docent Dr. Inž., predaje na Univerzitetu Metropolitan, Fakultetu za informacione tehnologije, od 2008.g. Njena specijalnost je informaciona bezbednost kompjuterskih sistema, i primena kriptologije. Radila je na nizu naučnih projekata u oblasti računarskih tehnologija, i objavila je niz naučnih radova. Član je Srpskog društva za računsku mehaniku i Srpskog društva za mehaniku.

Slobodan Jovanović,
slobodan.jovanovic@metropolitan,ac.rs
Prof. Dr. Inž., predaje na Univerzitetu Metropolitan u Beogradu, Fakultetu za informacione tehnologije, od 2008.g. Bavi se razvojem Web aplikacija, i „pametnim" električnim mrežama (*smart electric grids*), i veštačkom inteligencijom. Ima veliki boj objavljenih nučnih radova u vodećim internacionalnim časopisima. Predavao je na *Strathclyde University, Glasgow, Scotland*, u periodu 1993-2008. Učestvovao je u nizu naučnih i razvojnih projekata.