

UNIVERZITET SINGIDUNUM
BEOGRAD
DEPARTMAN ZA POSLEDIPLOMSKE STUDIJE
I MEĐUNARODNU SARADNJU

**POVEĆANJE POUZDANOSTI SSL PROTOKOLA
PRIMENOM VIŠESTRUKIH KANALA ZA
AUTENTIFIKACIJU**

- DOKTORSKA DISERTACIJA -

Mentor:

Prof. dr Mladen Veinović

Kandidat:

Marko Šarac, M.Sc.

Beograd, 2013.

Sažetak

Korisnici modernih elektronskih sistema podrazumevaju da je stepen zaštite ličnih podataka i resursa na visokom nivou. Svoje poverenje baziraju na pretpostavci da je strana koja pruža uslugu učinila sve da do propusta ne dođe. Komplikovani sistemi zaštite i administracije korisnike odvrćaju od korišćenja usluge ili servisa koje se nude na tržištu. U cilju postizanja najvišeg nivoa zaštite komunikacija profesionalni sistemi moraju razviti sopstvene sisteme zaštite, otporne na sve poznate tipove napada. Međutim profesionalni sistemi prinuđeni su i na balans između funkcionalnosti i zaštite korisnika. Svoje usluge postavljaju na poznatu infrastrukturu i u najvećem broju slučajeva štite SSL/TLS protokolom uz korišćenje podataka koji su korisniku poznati (korisničko ime i lozinka). Na ovaj način postignut je balans između funkcionalnosti i zaštite, zanemarujući pravila bezbednosti i autentifikacije za nečim što korisnik ima i nečim što korisnik jeste. U radu se razmatraju mogućnosti uvođenja sopstvenog rešenja koje ne umanjuje funkcionalnost uvodeći i drugi princip pravila bezbedne autentifikacije nečim što korisnik ima. Kroz rad je predstavljen detaljan proces razvoja i karakteristika rešenja koja obezbeđuje ova načela i time povećava pouzdanost SSL/TLS protokola višestrukim kanalima autentifikacije.

Abstract

Users of modern electronic systems believe that the level for protection of personal data and resources is at a high level. Their trust is based on the assumption that the party providing the service has done everything that a failure does not occur. The complicated system of protection and user administration discouraged users from using services or services offered on the market. In order to achieve the highest level of professional care communication systems have developed their own systems of protection, resistant to all known types of attacks. However, professional systems are forced to balance between functionality and user protection. Their service are set to the familiar infrastructure and in most cases protected with the SSL / TLS protocol with the use of data that are known to user (username and password). In this way, there has been a balance between functionality and security, ignoring safety rules and authentication for something a user has and something the user is. This paper discusses the possibility of introducing its own solution that does not diminish the functionality of system, and introducing the principle of other rules of safe authentication, something the user has. This paper presents detailed process of development and characteristics of solutions provided by these principles and thus increases the reliability of SSL / TLS authentication with multiple channels.

SADRŽAJ

1. UVOD	9
1.1. Motivacija i opšta razmatranja	9
1.2. Predmet istraživanja	12
1.3. Cilj istraživanja	15
1.4. Naučne hipoteze	16
1.5. Metode istraživanja	17
1.6. Naučni i stručni doprinosi	18
2. PREGLED U OBLASTI ISTRAŽIVANJA	19
2.1. Trenutna situacija	20
2.1.1. Podaci o napadima na elektronsku i ličnu svojinu	21
2.1.2. E-Bankarstvo i E-Trgovina	24
2.2. Kriptografski protokoli IEEE 802.11	25
2.2.1. Bežične računarske mreže štićene WEP tipom zaštite	27
2.2.2. Bežične računarske mreže štićene WPA tipom zaštite	29
2.2.3. Bežične računarske mreže štićene WPA2 tipom zaštite	30
2.3. Istraživanje zaštite koju nudi SSL, primena i napadi u bežičnim računarskim mrežama	33
2.4. Nedostaci HTTPS uspostave veze	36
2.4.1. Osnove uspostavljanja SSL veze	37
2.4.2. Analiza nedostataka	38
2.5. Napad na HTTPS saobraćaj	40
2.5.1. Napad prisluškivanjem saobraćaja	40
2.5.2. Napad zaobilaženjem sigurne veze	42
2.6. DOS napadi na bežične mreže	46
2.6.1. Tipovi DoS napada	46
2.6.2. DoS napadi na bežičnu mrežu	47
2.7. Analiza podataka prikupljenih nakon održanih savetovanja	49
2.8. Rešavanje problema	52
3. TEORIJSKE OSNOVE ISTRAŽIVANJA	56
3.1. Mehanizmi obezbeđivanja stabilne računarske mreže, sistemi za otkrivanje upada	57
3.1.1. SDU zasnovan na potpisu	59

3.1.2. SDU zasnovan na izuzetku	60
3.2. Sistemi za izviđanje i prikupljanje informacija	62
3.2.1. Horizontalno izviđanje – Izviđački alati	63
3.2.1.1. Sakupljanje opštih informacija.....	63
3.2.1.2. Veb sajтови	63
3.2.1.3. Pretraživači.....	64
3.2.1.4. Google „napadi“	64
3.2.1.5. Detaljno pretraživanje	65
3.2.1.6. Whois	66
3.2.1.7. DNS.....	68
3.2.1.8. Metapodaci.....	70
3.2.1.9. Metagoofil.....	70
3.2.1.10. Exiftool.....	71
3.2.1.11. Strings.....	72
3.2.1.12. Maltego.....	73
3.2.1.13. Odbrana od horizontalnog izviđanja	74
3.2.2. Vertikalno izviđanje - Alati za skeniranje	75
3.2.2.1. Nmap	75
3.2.2.2. Nessus	78
3.2.2.3. Odbrana od vertikalnog izviđanja	82
3.2.3. Izviđanje po dubini, sistemi za pristup i iskorišćavanje propusta, Alati za pristup i eskalaciju	82
3.2.3.1. Alati za šifru.....	83
3.2.3.2. Projekat Metasploit	86
3.2.3.3. Odbrana od izviđanja po dubini, sistema za pristup i iskorišćavanja propusta	90
3.3. Zaštita servisa računarskih mreža	91
3.3.1. Mrežne bezbednosne pretnje.....	92
3.3.2. Pristupi bezbednosti mrežnog saobraćaja	94
3.3.3. Sloj sigurnosnog adaptera i bezbednosni sloj prenosa.....	95
3.3.3.1. Arhitektura SSL-a	95
3.3.3.2. SSL protokol evidencije.....	98
3.3.3.3. Specijalni protokol za promenu šifrata	102
3.3.3.4. Protokol upozorenja	103
3.3.4. Protokol sinhronizacije	104
3.3.4.1. Faza 1. Uspostaviti sigurnosne mogućnosti	106
3.3.4.2. Faza 2. Autorizacija servera i razmena ključeva.....	108
3.3.4.3. Faza 3. Autorizacija klijenta i razmena ključeva	110
3.3.4.4. Faza 4. Kraj	111
3.3.5. Kriptografski proračuni.....	113
3.3.5.1. Generisanje kriptografskih parametara	114
3.4. Sigurnosni sloj prenosa	115

3.5.	HTTPS	120
3.5.1.	Iniciranje veze	121
3.5.2.	Zatvaranje veze	122
3.6.	Sigurnosni omotač (SSH – secure shell).....	122
3.6.1.	Transportni Protokol	124
3.6.2.	Protokol autorizacije korisnika/User Authentication Protokol	131
3.6.3.	Protokol veze	133
3.6.3.1.	Kanalni mehanizam.....	133
4.	<i>NOVI SISTEMI ANALIZE I ZAŠTITE RAČUNARARSKE MREŽE</i>	140
4.1.	Analiza WPA2 PSK računarske mreže.....	141
4.1.1.	Prvi korak napada na zaštićenu bežičnu računarsku mrežu.....	142
4.1.2.	Drugi korak napada na zaštićenu bežičnu računarsku mrežu	143
4.1.3.	Treći korak napada na zaštićenu bežičnu računarsku mrežu	144
4.1.4.	Četvrti korak napada na zaštićenu bežičnu računarsku mrežu	145
4.2.	Analiza prikupljenih podataka	146
4.2.1.	WPA i WPA2 napad korišćenjem rečnika (CPU)	148
4.2.2.	WPA i WPA2 napad korišćenjem rečnika (GPU)	150
4.2.3.	WPA i WPA2 brute-force napad	152
4.2.4.	WPA i WPA2 CPU klaster	153
4.3.	Napad po principu čovek u sredini	155
4.4.	Višekanalna autentifikacija - realizacija	163
4.4.1.	Struktura baze podataka	163
4.4.1.1.	Tabela users.....	163
4.4.1.2.	Tabela tokens	165
4.4.1.3.	Tabela logins	166
4.4.2.	Način generisanja tokena	167
4.4.3.	Aktivacija naloga	168
4.4.4.	Način prijave na portal.....	169
4.4.5.	Bezbednost pristupa portalu.....	169
4.4.5.1.	Implementacija aplikacija sa strane korisnika.....	169
4.4.5.2.	Razmena informacija između aplikacije i servera	170
4.4.6.	Izvorni kod aplikacije	172
4.5.	Hardverski uređaji za realizaciju napada	175
4.5.1.	Tehnički detalji o uređaju:	175
4.5.2.	Korišćenja hardverskog rešenja baziranog na WiFi Pineapple Mark IV	179
5.	<i>ZAKLJUČAK</i>	181
5.1.	Sumirani ciljevi istraživanja	181
5.2.	Ostvareni rezultati i doprinos.....	183
5.3.	Predlog daljeg rada	185

6.	<i>PRILOZI I DOKUMENTACIJA</i>	186
6.1.	Prilog 1. Android aplikacija	186
6.1.1.	Activate.java	186
6.1.2.	Pin.java.....	190
6.1.3.	Main.java	191
6.1.4.	Data.java	196
6.1.5.	Storage.java.....	196
6.1.6.	Factory.java.....	197
6.2.	Prilog 2. Serverska strana aplikacije	200
6.2.1.	Activate.php	200
6.2.2.	Config.php	201
6.2.3.	Index.php	202
6.2.4.	New_token.php	208
6.2.5.	Panel.php.....	211
6.3.	Prilog 3. MySQL upiti korišćeni na strani servera	215
7.	<i>BIBLIOGRAFIJA</i>	217
8.	<i>OBJAVLJENI RADOVI U NAUČNIM ČASOPISIMA I NA KONFERENCIJAMA</i>	227

SADŽAJ SLIKA I ILUSTRACIJA

Slika 1: Učestalost napada na elektronsku i ličnu svojinu	22
Slika 2: Finansijski troškovi zavisno do tipa napada	23
Slika 3: Prosečno vreme koje je potrebno da se napad otkrije	23
Slika 4: Kriptografski mehanizmi IEEE 802.11	25
Slika 5 : Inicijalizacija ključa u AES algoritmu.....	32
Slika 6: Uspostavljanje SSL veze	37
Slika 7: Ilustracija prisluškivanja HTTPS saobraćaja.....	41
Slika 8: Ilustracija preusmeravanja HTTP i HTTPS saobraćaja.....	43
Slika 9: Izgled ekrana Internet pretraživača, bez napada i sa napadom na klijenta.....	44
Slika 10: Rezultati ankete održane tokom jednodnevnih seminara	49
Slika 11: Rezultati ankete učesnika seminara po pitanju zaštite bežičnih računarskih mreža.....	50
Slika 12 : Upućenost i interesovanje korisnika o različitim zaštitama	51
Slika 13: Uticaj predavanja na polaznike.....	52
Slika 14: Whois upit.....	67
Slika 15: Nslookup komanda	68
Slika 16: Prikaz komande Dig	69
Slika 17: Metagoofil skripta.....	71
Slika 18: Maltego prikaz ekrana	73
Slika 19: Komanda Nmap osnovno skeniranje.....	76
Slika 20: Korišćenje komande Nmap za mapiranje portova	77
Slika 21: Prikaz primene jednog od mnogih dodatak Nessus programskog paketa	79
Slika 22: Prikaz konfiguracije koja omogućava podešavanje Nessusa za skeniranje mrežnog sistema.....	79
Slika 23: Rezultati nakon skeniranja i specifikacija određenog uređaja.....	80
Slika 24: Nessus izveštaj, sa detaljnim o osetljivosti skeniranog sistema	81
Slika 25: Prikaz alata Hydra	85
Slika 26: Prikaz po pokretanju msfconsole Metasploit okvira.....	87
Slika 27: Grafičko okruženje Metasploita kroz Armitage	88
Slika 28: Relativna lokacija sigurnosti objekata u TCP/IP protokolima	94
Slika 29 : Slojevi SSLa	96
Slika 30: Operacija dodavanja SSL zapisa	99
Slika 31: Format SSL zapisa.....	102
Slika 32: Paket SSL evidentacionog zapisa	103
Slika 33: Inicijalna razmena koja je neophodna kako bi se uspostavila veza između klijenta i servera.....	106
Slika 34: TLS funkcija P_hash (tajna, dodatak)	117
Slika 35: SSH Protokol	124

Slika 36: SSH Protokol Transportnog sloja	126
Slika 37: Formiranje SSH transportnog paketa	127
Slika 38: SSH Transportni sloj šifarski algoritmi	129
Slika 39: Primer razmene poruka na protokolu veze	135
Slika 40: SSH transportni sloj, razmena podataka	137
Slika 41: Prebacivanje kartice u mod za osluškivanje	143
Slika 42: Prikaz rezultata skeniranja okolnih mreža	144
Slika 43: Prikaz otpočinjanja prisluškivanja	144
Slika 44: Prikaz rezultata prisluškivanja	145
Slika 45: Prikaz deautentifikacije korisnika	145
Slika 46: Prikaz uspešnog hvatanja hendšejk poruka	146
Slika 47: Prikaz komande za izvoz podataka iz poruka hendšejka	146
Slika 48: Prikaz podataka sadržanih u hendšejk porukama	147
Slika 49: Prikaz komande za pokretanje pretrage	148
Slika 50: Prikaz procesa pronalaženja lozinki	149
Slika 51: Prikaz hešket alata u toku rada	151
Slika 52: Prikaz brute-force napada	153
Slika 53: Pokretanje ARPspooft-a i pregled opcionih parametara	156
Slika 54: Utvrđivanje putanje kretanje korisnika kroz mrežu ka udaljenoj lokaciji	157
Slika 55: Preusmeravanje saobraćaja korisnika ka napadaču metodom arpspoof	158
Slika 56: Alat SSLstrip, pregled mogućih opcija	159
Slika 57: Usmeravanje saobraćaja klijenta nakon pregledavanja od strane napadača	159
Slika 58: Konfigurisan alat SSL strip tako da beleži aktivnosti korisnika	160
Slika 59: Prikaz alata Wireshark , analiza prikupljenih podataka korisnika	161
Slika 60: Analiza saobraćaja, kretanja i podataka korisnika uprkos SSL zaštiti	162
Slika 61: Grafički prikaz tabele users	164
Slika 62: Grafički prikaz tabele tokens	165
Slika 63: Grafički prikaz tabele logins	166
Slika 64: Dijagram generisanja tokena	167
Slika 65: Dijagram prijave korisnika na sistem	171
Slika 66: Fizička reprezentacija uređaja WiFi Pineapple Mark IV	175
Slika 67: Korisnički interfejs WiFi Pineapple Mark IV	176
Slika 68: Instalacija dodatnih servisa WiFi Pineapple Mark IV	177
Slika 69: Konfiguracija i parametra WiFi Pineapple Mark IV	178
Slika 70: Priprema SSL Strip napada korišćenjem WiFi Pineapple Mark IV	179

SADRŽAJ TABELA

Tabela 1: Poređenje pretnji na mreži	93
Tabela 2: Enkripcijski algoritmi dozvoljeni pri korišćenju SSLa.....	101
Tabela 3: Parametri korišćeni u protokolu uspostave/sinhronizacije	105
Tabela 4: Referentni računar.....	141
Tabela 5: Prosečna vremena pretrage po rečniku	150
Tabela 6: Pretraga po rečniku korišćenjem grafičkih procesora.....	151
Tabela 7: Poređenje laptop i destktop grafičkih procesora.....	152
Tabela 8: Poređenje Hashcat i sopstvenog rešenja	154
Tabela 9: Komponente android aplikacije i njihova uloga	173
Tabela 10: Komponente serverske strane aplikacije i njihova uloga.....	174

1. UVOD

1.1. Motivacija i opšta razmatranja

Sa pojavljivanjem prvog standarda bežičnih računarskih mreža teško je bilo pretpostaviti u kojoj meri i kojom brzinom će se razvijati bežične mreže. Standard 802.11 a/b odobren je 1999. godine i od tada se proizvodi bazirani na bežičnoj tehnologiji pojavljuju na tržištu. Ubrzo po pojavljivanju otkrivena je mana u sistemu zaštite bežičnih mreža. Greška se nalazila u WEP kriptografskom metodu što je detaljno opisano i analizirano u radu. Zbog naglog rasta broja kućnih i poslovnih korisnika bežičnih računarskih mreža, kao i zbog lakoće instalacije i korišćenja, problem je postajao sve izraženiji. Usled toga, predloženo je novo rešenje u vidu WPA2 bezbednosnog standarda, koji koristi AES šifarski mehanizam umesto ranjivog RC4 TKIP sistema. Predloženo novo rešenje nije kompatibilno sa prethodnim. Na taj način, vlasnicima starih sistem bila je uskraćena bezbednost ukoliko se nisu odlučivali za nabavku novih uređaja koji su podržavali WPA2 standard. U 2009. godini pojavili su se prvi napadi na WPA2 standard, a odnosili su se samo na kratke šifre i šifre koje se zasnivaju na opšte poznatim rečima (slabe šifre). U ovom radu razmatra se problem pridruživanja korisnika (uključivanje klijenata) modernim bežičnim računarskim mrežama. Pored toga, razmatraju se problemi koje „legitimni” korisnici mogu izazvati u bežičnim računarskim mrežama baziranim na najnovijem WPA2 sigurnosnom standardu. Analize u radu sprovedene su nad aktuelnom metodama zaštite bežičnih računarskih mreža kako bi se istakao problem koji se javlja i u poslednjoj generaciji zaštitnih sistema. Ovakve sisteme korisnici često smatraju potpuno bezbednim iako to kao što je analiza pokazala ne mora biti slučaj.

Nakon analize bežičnih računarskih mreža i njihove bezbednosti, predstavljen je i analiziran najčešće korišćeni metod zaštite korisnika sistema elektronske trgovine i elektronskog bankarstva. Koncept zaštite elektronske trgovine i elektronske zaštite obezbeđene SSL/TLS vezom doveden je u korelaciju budući da se najčešći napadi na moderne sisteme zaštite ovih usluga dešavaju upravo u bežičnim računarskim mrežama koje danas čine veliki procenat kućnog ali i poslovnog okruženja računarskih mreža. Korisnici ovih sistem svoju bezbednost i resurse poveravaju sistemima elektronske trgovine i elektronskog bankarstva, podrazumevajući da je sistem koji pruža usluge pravilo obezbeđen.

Sveobuhvatna analiza je pokazala da ovo najčešće nije slučaj i da zaštita bazirana isključivo na implementaciji SSL protokola najčešće nije dovoljna.

U radu se razmatra više međusobno povezanih problema:

- Analiza aktuelnih napada na elektronsku i ličnu svojinu
- Istraživanje zaštite koju nudi SSL, primena i napadi u bežičnim računarskim mrežama
- Nedostaci HTTPS uspostavljanja veze
- Napadi na HTTPS saobraćaj
- Analiza anketa sa održanih savetovanja
- Predlog rešenja za odbranu od napada
- Problem hardverskih uređaja za sprovođenje napada na SSL protokol u bežičnim računarskim mrežama
- Predlog povećanja pouzdanosti SSL protokola implementacijom sopstvenog rešenja za višekanalnu autentifikaciju

U periodu izrade rada organizovano je više predavanja o problemima u zaštiti bežičnih računarskih mreža kao i o problemu pravilne implementacije SSL protokola. Predavanja i naučno istraživanje vezani za ove probleme podržani su od strane Ministarstva za nauku i tehnološki razvoj Republike Srbije kroz projekat TR32054 – Digitalna obrada signala u sintezi sistema za zaštitu informacija. Predavanja su organizovan je za srednjoškolce i studente, ali i za postdiplomske studenata, kao i za zaposlene u javnom i privatnom sektoru. Cilj seminara i predavanja bio je edukacija, ukazivanje na probleme, ali i uzorkovanje grupe koja je učestvovala u samom seminaru. Rezultati su prikupljeni anketiranjem nakon predavanja, a isti su predstavljeni i analizirani u delu rada koji se bavi analizom anketa učesnika konferencija.

Rad se bavi analizom bezbednosti, ispitivanjem koje se klasifikuje prema tehnikama skeniranja. Analize penetracijskog ispitivanja prema tehnici skeniranja dele se na:

- ispitivanje bazirano na računarskim mrežama
- ispitivanje bazirano na ostalim načinima komunikacije,
- fizičko penetracijsko ispitivanje i
- socijalni inženjering

Korišćenje bežične mreže podrazumeva prenos putem medijuma koji je dostupan svima na mnogo većem prostoru od onog koji neophodna samom korisniku. Prema tome, svako može da presretne pakete podataka koji su u etru. Dovoljan je bežični adapter i odgovarajući softver.

Ispitivanje bazirano na računarskim mrežama je standardan način izvođenja penetracijskog ispitivanja sigurnosti u računarskim sistemima. Analizira se mrežna infrastruktura poslovnog okruženja i računarskih stanica na njoj. Za skeniranje koristi se TCP/IP (*Transmission Control Protocol / Internet Protocol*) protokol.

Ispitivanje bazirano na ostalim načinima komunikacije primenjuje se za ispitivanje ostalih načina komunikacije unutar poslovnog okruženja. Analiziraju se telekomunikacione mreže, različite bežične mreže za komunikaciju (*bluetooth*) i druge.

Fizičkim penetracijskim ispitivanjem analizira se poslovno okruženje sa aspekta fizičke sigurnosti.

Ispitivanje bazirano na socijalnom inženjeringu, koristi se za iskorišćavanje sigurnosne nesvesnosti zaposlenog u poslovnom okruženju u cilju sprovođenja aktivnog, pasivnog napada ili se može koristiti kao pogodna metoda za analiziranje procesa poštovanja i sprovođenja sigurnosne politike u poslovnom okruženju.

1.2. Predmet istraživanja

Sa pojavljivanjem prvog standarda bežičnih računarskih mreža teško je bilo pretpostaviti u kojoj meri i kojom brzinom će se razvijati bežične mreže. Bežične računarske mreže danas čine jedan od glavnih medijuma razmene podataka. Dobro je poznato da sve više poslova i delova našeg života zavise od računara i obavljanja zadataka preko računarskih mreža. Zbog toga sve više istraživača obavlja analize i istraživanja vezana za bezbednost računarskih sistema i poslovnih transakcija. S obzirom na to da je broj mrežnih servisa, novih korisnika usluga svakodnevno u porastu, povećava se i vrednost razmenjenih informacija. Informacije koje se prenose preko mreže i čuvaju na hardveru mogu biti zloupotrebljene ili ukradene ukoliko nisu zaštićene od neovlašćenog pristupa. U mrežnim komunikacijama prisutni su razni bezbednosni problemi kod kontrole pristupa, uglavnom zbog lažnog predstavljanja ili krađe identiteta.

Razvoj e-trgovine i e-bankarstva vezan je za sve širu primenu računarske i telekomunikacione tehnologije u procesu obrade, prenosa podataka i informacija. Ove tehnologije dovode do krupnih promena u funkcionisanju banaka, finansijskog sektora i trgovine. Pokrenuti procesi dovode do uvođenja novih postupaka i tehnologija, koje sa sobom nose nove izazove i mogućnosti. Tehničko - finansijske inovacije, sa svoje strane, vode intenziviranju konkurencije u svim segmentima. Dolazi, ne samo do reinženjeringa poslovnih procesa u bankama, finansijskom sektoru, trgovini, već i do reinženjeringa u poslovnim procesima i odnosima između samih banaka, banaka i njihovih klijenata i između klijenata i trećih lica međusobno.

Glavni cilj e-trgovine i e-bankarstva je da se bez obzira na geografsku udaljenost što brže i efikasnije povežu klijenti i tokovi informacija. Tajna uspešnog savremenog informacionog sistema je fleksibilna infrastruktura bazirana na personalnim računarima, modernim računskim mrežama i Internetu pre svega. Razvoj informacione i telekomunikacione tehnologije stvorio je uslove za globalizaciju poslovanja banaka tj. do uvođenja novih kanala distribucije bankarskih usluga, odnosno elektronskog bankarstva. Nova revolucija u oblasti tehnoloških promena, međutim, bazirana je na ideji da se za automatizaciju transakcija u e-bankarstvu, e-trgovini ne formira nova infrastruktura, već da se iskoristi postojeća infrastruktura javnih računskih mreža, pre svega Interneta. Internet spada u

javne tj. otvorene mreže, kojima svako ima neograničeni pristup. Današnje globalno tržište i oštra konkurencija nameću takve uslove organizacijama, da se one suočavaju sa velikim brojem rizika kojima treba upravljati. Upotreba Interneta za automatizaciju transakcija plaćanja na malo, povoljna je zbog niskih troškova po jednoj transakciji. Veliki problem, međutim, predstavlja bezbednost transakcija koje se obavljaju preko Interneta.

Nivo zaštite podataka kod razmene elektronskih poruka izuzetno je značajan element na osnovu kojeg se korisnici opredeljuju za ili protiv primene e-transakcija. To je posebno karakteristično u slučajevima kada gubitak, izmena, neovlašćeni uvid ili neke druge pretnje elektronskim dokumentima mogu da izazovu značajne pravne i finansijske posledice po učesniku u on-line transakciji. Naime, u e-poslovanju i razmeni elektronskih dokumenata učestvuju brojni partneri i svi moraju da imaju odgovarajući nivo zaštite.

Sa brzim razvojem mrežnih aplikacija, problem bezbednog prenosa postaje veoma važan. Zbog toga je SSL (*Secure Sockets Layer*) protokol sve više u širokoj upotrebi u različitim mrežnim servisima. Ovaj protokol sam po sebi nije savršen i u praksi sa njim često postoje problemi. Ovaj rad ukazuje na problem koji postoji u procesu SSL uspostavljanja veze. Prvi nedostatak kod uspostavljanja veze je što se u početnoj fazi koristi otvorena komunikacija, pa postoji mogućnost neovlašćenog praćenja sadržaja. Drugi nedostatak je implementacija SSL-a u aplikaciju. Zbog razmatranja faktora o radu mrežne veze, obično se koristi prekidač veze zasnovan na HTTP (*HyperText Transfer Protocol*) protokolu koji preusmerava saobraćaj na HTTPS (*Hypertext Transfer Protocol Secure*). Napadač je u mogućnosti da na relativno lak način iskoristi ove nedostatke. Kroz izradu doktorske disertacije obavljeni su eksperimenti koji su pokazali da se na više načina mogu izazvati značajni bezbednosni rizici u HTTPS komunikaciji. Pokazalo se da SSL nije savršen za svaki tip primene ukoliko se koristi kao jedini faktor bezbednosti. Još 2003. realizovani su napadi na SSL protokol i dokazane su greške u uspostavljanju veze koje su omogućavale napad preuzimanja sesije. Proizvođači Internet pretraživača unapređivali su zaštitu svojih klijenata koja se zasniva na upozorenjima na nevalidne sajtove kao i na lažne sertifikate. Kompanije za antivirusna rešenja razvile su rešenja za zaštitu od ARP (*Address Resolution Protocol*) prevara, koja se najčešće koriste pri napadima. Međutim, pored svih zaštita i inicijativa proizvođača programa često su nastajali bezbednosni propusti koji se i danas koriste.

Danas u savremenom poslovanju, dosta autora predlaže sopstvena rešenja. Postoji nekoliko protivmera koje sprečavaju napade na SSL/TLS (*Secure Sockets Layer / Transport*

Layer Security), bazirane na aplikaciji za internet poslovanje, e-trgovinu i e-bankarstvo. Većina autora predlaže rešenja u kojima je važno shvatiti da SSL/TLS protokol može zaštititi od napada, ali samo ako korisnik i server poseduju važeće sertifikate i ako se istovremeno identifikuju koristeći ove sertifikate. Drugim rečima, samo rešenje zavisi od elektronske pismenosti korisnika. Današnji sajtovi za e-trgovinu i e-bankarstvo često rešavaju problem, tako što zaobilaze isti. U praksi savremenog elektronskog poslovanja, naročito u Evropi, najčešće se autentifikacija korisnika vrši slanjem SMS poruke koja sadrži verifikacioni kod. Kod mora da se unese pri prijavi. Slučaj slanja SMS poruke je primer korišćenja dva komunikaciona kanala za autentifikaciju sa dva faktora (mobilni uređaj je drugi faktor). Često se tvrdi da ovaj mehanizam štiti od napada, nažalost, to nije uvek slučaj. Ukoliko je napadač između korisnika i servera, napadač nema potrebu za presretanjem SMS poruke. Sve što treba da uradi, jeste da ubedi korisnika da kod koji je dobio prosledi njemu, kako bi nesmetano uspostavio komunikaciju sa serverom. Ukoliko postoji potreba za korišćenjem autentifikacije putem SMS poruke, bolje je ovaj mehanizam koristiti tako da se poruka šalje za svaku transakciju, a ne samo za prijavu korisnika. Svaka zatražena transakcija od strane korisnika, mora biti ispraćena SMS porukom sa kodom i zatražena potvrda korisnika. Za potvrdu transakcije, korisnik mora ukucati kod. Naravno, postoje i nedostaci ovog koncepta. Nekoliko kriptografskih tehnika i protokola je tokom godina predloženo kao prevencija i zaštita od napada. *Rivest i Shamir* predložili su *Interlock* protokol, ali daljim istraživanjem pokazano je da je ranjiv kao servis autentifikacije. Od strane *Jakobssona i Mayersa* predložena je tehnika nazvana odloženo otkrivanje lozinke DPD (*Digital Pre-Distortion*). DPD se može koristiti kao dodatak autentifikaciji zasnovan na lozinci i protokolima za razmenu ključeva, kako bi korisnika zaštitili od specifičnih napada. DPD zahteva autentifikaciju zasnovanu na lozinci i protokolima za razmenu ključeva, ali korisnika ne štite u potpunosti od napada koji kontroliše interfejs internet pretraživača. Kroz rad je predloženo sopstveno rešenje višekanalne autentifikacije zasnovano na Android mobilnoj aplikaciji i kodovima koji su validni samo pri inicijalnom korišćenju.

1.3. Cilj istraživanja

Problem bezbednosti u žičnim i bežičnim računarskim mrežama svakako postoji. Ovaj problem posebno se ogleda u primeni e-trgovine i e-bankarstva. Cilj ovog rada je sveobuhvatna analiza problema, sumiranje rezultata mogućih rešenja, primena mogućih rešenja, ali i razvoj sopstvenog rešenja za prevazilaženje problema. Kroz rad će biti prikazano da i u najpouzdanijim metodama koje se koriste u najvećim sistemima, za koje većina smatra da su bezbedni, postoje nedostaci koje zlonamerni korisnici mogu iskoristiti. Od situacije u kojoj su napade sprovodili retki, izuzetno upućeni i teorijski dobro potkovani pojedinci, došlo se do situacije da današnji alati za penetracijske testove ne zahtevaju previše znanja od strane samog napadača. Ovakve alate koji su prevashodno napravljeni za penetracijsko testiranje i dijagnostikovanje računarskih mreža zlonamerni pojedinci koriste u cilju dobijanja lične koristi. Zbog svega navedenog bitno je obraditi tematiku bezbednosti i podizanje svesti o propustima koji mogu biti zlonamerno iskorišćeni. U radu je data sveobuhvatna analiza problema, prezentovan je i analiziran sopstveni koncept zaštite od ovakve vrste napada, kao i metodologija sprovođenja istog.

Elektronsko poslovanje nesporno je donelo mnoge „blagodeti“, ali i brojne opasnosti. Što je njegov rast i zastupljenost veća, to su i posledice zloupotrebe teže. U zemljama u kojima se ovaj oblik poslovanja nalazi u stalnoj ekspanziji, preduzimaju se mnogobrojne aktivnosti u zaštiti, ali i hvatanju i kažnjavanju onih kojima zloupotrebe postaju „zanimanje“.

Aktuelnost obrazložene teme i potreba za istraživanjem u ovoj oblasti potvrđena je od Ministarstva za nauku i tehnološki razvoj Republike Srbije. Šifra projekta: TR32054, naziv projekta: “Digitalna obrada signala u sintezi sistema za zaštitu informacija“, prijavljenog u kategoriji projekata tehnološkog razvoja za period 2011- 2014 na kom autor i metor učestvuju kao naučnici istraživači.

1.4. Naučne hipoteze

Hipoteze koje čine osnovu ovog rada su:

1. U uslovima globalizacije poslovanja, jake konkurentnosti i promenljivosti okruženja, elektronsko poslovanje sve više dobija na važnosti i neophodnosti. Obezbeđivanje sigurne komunikacije nameće se kao imperativ uspešnog obavljanja elektronskih transakcija.
2. Primena i usvajanje elektronskog poslovanja u Republici Srbiji, direktno će zavisiti od stepena poverenja klijenata u sigurnost i bezbednost obavljenih transakcija.
3. Uspešnost primene elektronskog poslovanja i elektronskih transakcija u direktnoj je korelaciji sa kvalitetom procesa njegovog projektovanja, sprovođenja i bezbednosti.

Posebne hipoteze:

1. Bankarsko tržište kao najreprezentativniji korisnik elektronskih transakcija predstavlja jak kontrolni mehanizam zaštite sigurne komunikacije, intenzivno promoviše upotrebu elektronskih transakcija radi ostvarenja boljih poslovnih i finansijskih performansi.
2. Uspešnost obezbeđivanja sigurnog elektronskog poslovanja je u direktnoj korelaciji sa redovnom kontrolom i revizijom informacionih sistema i sistema zaštite komunikacije.
3. Od ekonomske snage privrednih subjekata direktno zavisi uvođenje i pravilna primena novih tehnologija.

1.5. Metode istraživanja

Složenost predmeta istraživanja zahteva primenu:

- od analitičkih osnovnih metoda: metod analize, metod apstrakcije, metod specijalizacije i metod dedukcije;
- od sintetičkih osnovnih metoda: sintezu, konkretizaciju, generalizaciju i indukciju;
- od opšte naučnih metoda: hipotetičko-deduktivnu, analitičko-deduktivnu, komparativnu, statističku i metodu modelovanja.

Primenom ovih metoda, kako govore dosadašnji rezultati istraživanja, moguće je validno ostvarenje naučnog i društvenog cilja istraživanja. Pristup istraživanju je integrativan i sintetički, u tom smislu što se ni jednom metodološkom postupku ne daje isključiva prednost.

U prikupljanju podataka primeniće se: ispitivanje, anketu i metodu analize sadržaja dokumenata.

Analiza će biti ostvarena na dva nivoa:

- na nivou eksperimentalne analize realnih subjekata
- na nivou sekundarne analize rezultata ranijih istraživanja i adekvatne literature.

1.6. Naučni i stručni doprinosi

Stručni doprinosi

- pregled svetskih iskustava u oblasti zaštite i obezbeđivanja sigurne komunikacije,
- pregled tehnologija koje se koriste za projektovanje i implementaciju sistema za zaštitu i obezbeđivanje sigurne komunikacije,
- primer korišćenja standardne metodologije i alata za softversko modelovanje radi proširivanja funkcionalnosti postojećih aplikacija i implementacija sigurne komunikacije,
- razvoj sopstvenog rešenja distribucije zaštićenih podataka višestrukim kanalima distribucije, verifikovanog sa aspekta teorijsko informacione analize, opisanim procesom razvoja, upotrebe, analize performansi i poređenja sa ostalim srodnim rešenjima.

Naučni doprinosi

- Sveobuhvatna analiza sistema zaštite u bežičnim računarskim mrežama,
- Predlog sopstvenog rešenja analize sistem zaštite bežičnih računarskih mreža sistemom distribuiranog računarstva,
- Sveobuhvatna analiza sistema zaštite baziranih na SSL/TLS protokolu,
- Pregled i analiza dosadašnjih istraživanja i dostignuća u oblasti kreiranja sigurne komunikacije korišćenjem SSL/TLS protokola,
- Predlog sopstvenog rešenja poboljšanja i upotrebe višekanalne, višefaktorske autentifikacije korišćenjem postojećih SSL/TLS protokola,
- Predlog smernica za dalji razvoj.

2. PREGLED U OBLASTI ISTRAŽIVANJA

2.1. Trenutna situacija

Fokus disertacije su bežične računarske mreže i bezbedna komunikacija unutar istih. U upotrebi su sledeći tipovi bežičnih računarskih mreža:

- Nezaštićene bežične računarske mreže – Najčešća primena je za hotspotove, besplatne pristupne tačke po restoranima, hotelima – osnovna karakteristika ovih mreža je da iste ne zahtevaju lozinku da bi postali njihov član.
- Bežične računarske mreže štice WEP tipom zaštite – prema poslednjim istraživanjima 25% bežičnih računarskih mreža današnjice još uvek koristi ovaj prevaziđeni standard. Često se koristi u kućnim mrežama, kablovski operateri i internet provajderi ga još uvek stavljaju kao preporučeni tip zaštite¹.
- Bežične računarske mreže štice WPA tipom zaštite – moderniji i sve češće korišćeni tip zaštite. Znatno teži za nelegalno pridruživanje².
- Bežične računarske mreže štice WPA2 tipom zaštite – trenutno najmoderniji tip zaštite bežičnih računarskih mreža. Koristi se u poslovnom okruženju i bolje konfigurisanom kućnom i hotelskom okruženju.

Važno je napomenuti da gotovo svi aktuelni modeli bežičnih pristupnih uređaja podržavaju sve navedene standarde. Sporadično se može desiti da neki od starijih klijentskih uređaja ne podržava WPA2 tip zaštite ali je to zaista redak slučaj.

Bez obzira na tip bežične računarske mreže, korisnici imaju potrebu i želju za bezbednom komunikacijom. Bezbedna komunikacija unutar računarske mreže često se poverava SSL/TLS protokolu. Kroz disertaciju ukazano je na nedostatke SSL/TLS protokola i predstavljene su mogućnosti za unapređenje istog.

Predstavljani su mogući načini priključenja bežičnim računarskim mrežama koji se mogu koristiti radi dalje analize ovih mreža. Jedini zvanični i zakonski dozvoljeni način

¹ S. Fluhrer, I. Mantin, A. Shamir. Weaknesses in the key scheduling algorithm of RC4. Proceedings of the 8th Workshop on Selected Areas in Cryptography. 2001

² Wi-Fi Alliance. Wi-Fi Protected Access. http://www.wi-fi.org/white_papers/whitepaper-042903-wpa/

priključenja bežičnoj računarskoj mreži je ukoliko je ona ne zaštićena ili ukoliko su pristupni parametri dobijeni od vlasnika mreže. Primeri koji slede rađeni su za potrebe analize i na sopstvenoj bežičnoj računarskoj mreži. Svrha rada i korišćenih programskih paketa je analiza a ne zloupotreba i nelegalno korišćenje bežičnih računarskih mreža.

Nezaštićene bežične računarske mreže, kao što im i samo ime kaže, ove mreže ne zahtevaju autentifikaciju tj. posedovanje lozinke za prijavu na ovu bežičnu računarsku mrežu. Saobraćaj u ovakvim mrežama se ne šifrjuje a do klijentskih podataka koji su razmenjeni kroz ovakvu bežičnu računarsku mrežu se relativno lako dolazi dostupnim i komercijalnim programima za skeniranje etra³. Ovakve mreže popularne su na javnim lokacijama, tržnim centrima, restoranima, kafićima. Bitno je napomenuti da kod većine ADSL operatera nezaštićena bežična mreža predstavlja preporučeno stanje što svakako nije dobro⁴. Gotovo svi moderni operativni sistemi upozoravaju vlasnike na posledice korišćenja ovakvih mreža, tj. na mogućnost zloupotrebe i vidljivost razmenjenih podataka⁵.

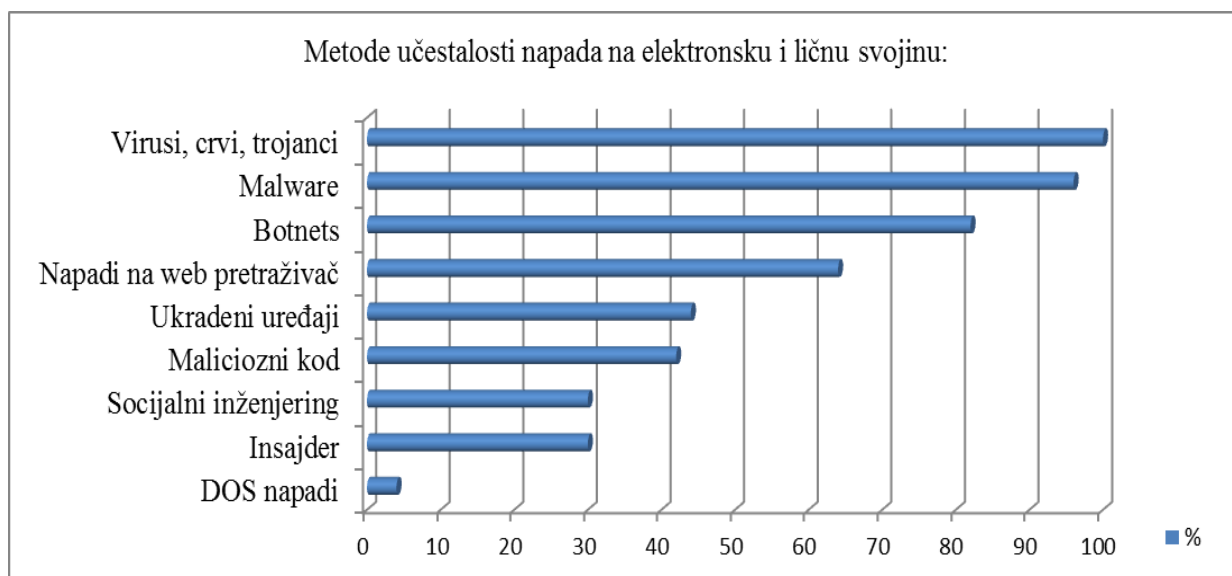
2.1.1. Podaci o napadima na elektronsku i ličnu svojinu

Dobro je poznato da sve više poslova i delova života zavise od računara i obavljanja zadataka preko računarskih mreža. Zbog toga sve više istraživača obavlja analize i istraživanja vezana za bezbednost računarskih sistema i poslovnih transakcija. Kroz naredne primere biće prikazane globalne statistike najčešćih napada na računarske sisteme, zastupljenost različitih napada, koliku finansijsku štetu organizaciji nanose, kao i koliko je dana prosečno potrebno organizaciji da postane svesna da je napadnuta.

³ W. Arbaugh, N. Shankar, J. Wan, K. Zhang. Your 802.11 network has no clothes. IEEE Wireless Communications Magazine, 9(6):44-51, 2002.

⁴ Frank Stajano and Ross J. Anderson. The resurrecting duckling: Security issues for ad-hoc wireless networks. 7th International Workshop on Security Protocols, pages 172-194, London, UK, 2000. Springer-Verlag.

⁵ Jari Arkko and Pekka Nikander. Weak authentication: How to authenticate unknown principals without trusted parties. In Bruce Christianson, Bruno Crispo, James A. Malcolm, and Michael Roe, editors, Security Protocols Workshop, volume 2845 of Lecture Notes in Computer Science, pages 5-19. Springer, 2002.



Slika 1: Učestalost napada na elektronsku i ličnu svojinu

Na ilustraciji (Slika 1) je predstavljena klasifikacija po učestalosti napada. Klasifikacija je predstavljena po istraživanju koje je obavio Ponemon institut u avgustu 2012. godine.⁶ Kao što se iz same ilustracije vidi najveći broj napada se čini posredstvom virusa, crva, trojanaca, malwera, dok su maliciozni kodovi i insajderi ređe zastupljeni tip napada. Međutim, upravo ti ređi napadi čine najveću finansijsku štetu organizaciji.

Prema podacima sa ilustracije (Slika 2), a po istraživanju Ponemont instituta maliciozni kod identifikovan je kao najveći uzročnik finansijskih troškova u okviru organizacije, dok su virusi, crvi i trojanci koji su identifikovani kao najčešći tip napada odneli svega 8% ukupnih finansijskih troškova organizacije u oblasti zaštite i napada⁷.

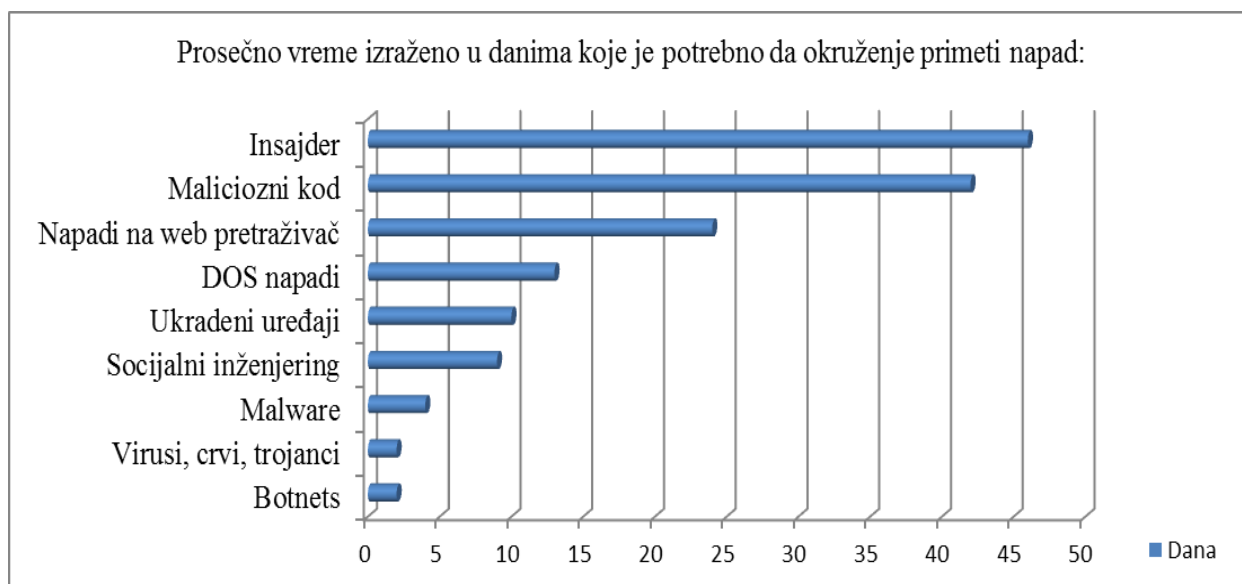
⁶ Ponemon Institute© Research Report, Cost of Cyber Crime Study, August 2012

⁷ <http://www.ponemon.org/about-ponemon-research>



Slika 2: Finansijski troškovi zavisno do tipa napada

Organizacija često smatra da uspešno funkcioniše i pravilno radi i sa insajderom u svojim redovima. Ovakav tip napada je teško otkriti i oduzima najviše vremena, a odgovarajući podaci su dati na narednoj ilustraciji (Slika 3). U proseku je potrebno više od 45 dana za identifikaciju insajdera. Maliciozni kod takođe je čest uzrok problema koji je teško identifikovati i najčešće je potrebno preko 40 dana dok organizacija postane svesna da ima problema sa malicioznim kodom. Ovaj tip napada je najveći uzročnik finansijskih gubitaka organizacije.



Slika 3: Prosečno vreme koje je potrebno da se napad otkrije

2.1.2. E-Bankarstvo i E-Trgovina

Razvoj e-bankarstva i e-trgovine vezan je za sve širu primenu računarske i telekomunikacione tehnologije u procesu obrade i prenosa podataka i informacija. Ove tehnologije dovode do krupnih promena u funkcionisanju banaka. Pokrenuti procesi dovode do uvođenja novih postupaka i tehnologija, koje sa sobom nose nove izazove i mogućnosti. Tehničko - finansijske inovacije, sa svoje strane, vode intenziviranju konkurencije u svim segmentima. Dolazi, ne samo do reinženjeringa poslovnih procesa u bankama, već i do reinženjeringa u poslovnim procesima i odnosima između samih banaka, banaka i njihovih klijenata i između klijenata i trećih lica međusobno.

Glavni cilj e-bankarstva i e-trgovine je da se bez obzira na geografsku udaljenost što brže i efikasnije povežu klijenti i tokovi informacija. Tajna uspešnog savremenog informacionog sistema je fleksibilna infrastruktura bazirana na personalnim računarima, modernim računskim mrežama i Internetu pre svega. Razvoj informacione i telekomunikacione tehnologije stvorio je uslove za globalizaciju poslovanja banaka tj. do uvođenja novih kanala distribucije bankarskih usluga, odnosno elektronskog bankarstva⁸.

Nova revolucija u oblasti tehnoloških promena, međutim, bazirana je na ideji da se za automatizaciju transakcija u bankarstvu ne formira nova infrastruktura, već da se iskoristi postojeća infrastruktura javnih računskih mreža, pre svega Interneta. Internet spada u javne tj. otvorene mreže, kojima svako ima neograničeni pristup. Upotreba Interneta za automatizaciju transakcija plaćanja na malo, povoljna je zbog niskih troškova po jednoj transakciji. Veliki problem, međutim, predstavlja bezbednost transakcija koje se obavljaju preko Interneta⁹.

Nivo zaštite podataka kod razmene elektronskih poruka izuzetno je značajan element na osnovu kojeg se korisnici opredeljuju za ili protiv primene e-transakcija. To je posebno karakteristično u slučajevima kada gubitak, izmena, neovlašćeni uvid ili neke druge pretnje elektronskim dokumentima mogu da izazovu značajne pravne i finansijske posledice po

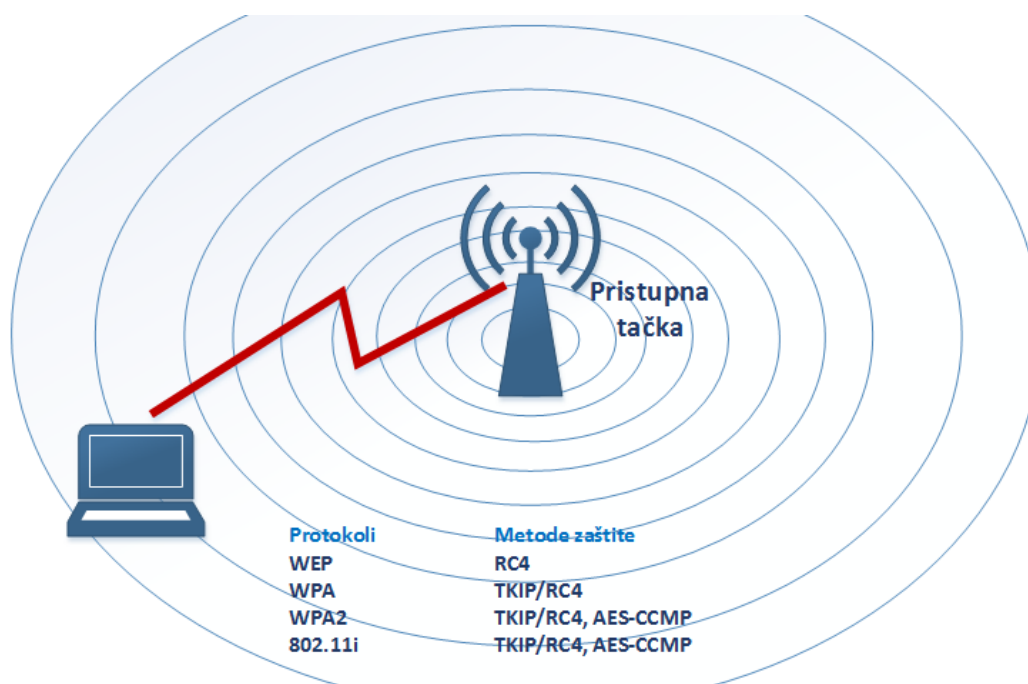
⁸ E-Banking Management: Issues, Solutions, and Strategies; Mahmood Shah and Steve Clarke (Apr 23, 2009)

⁹ The Silver Bullet for E-Commerce Security.(Public Key Infrastructure; digital evidence of online transactions); Don Waugh (Jul 28, 2005)

učesniku u on-line transakciji. Naime, u e-poslovanju i razmeni elektronskih dokumenata učestvuju brojni partneri i svi moraju da imaju odgovarajući nivo zaštite.

2.2. Kriptografski protokoli IEEE 802.11

Svi kriptografski protokoli bežičnih uređaja u osnovi koriste RC4 kriptografski algoritam. RC4 je šifra koja pripada grupi sekvencijalnih šifara. Sam algoritam je zasnovan na promenljivim tabelama ili samo modifikujućim tabelama. Tabela uvek sadrži jednu permutaciju brojeva od 0 – 255. Inicijalna permutacija algoritma je unutrašnji ključ. U svakom koraku RC4 vrši zamenu elemenata u tekućoj tabeli i selektuje byte radnog ključa iz tabele po principu one-time pad-a. Ovaj algoritam u svakom koraku produkuje po jedan byte šifrata u odnosu na A5/1 koji produkuje jedan bit. Zbog dobrih performansi efikasno se može softverski realizovati.



Slika 4: Kriptografski mehanizmi IEEE 802.11

Wired Equivalent Privacy (WEP) je algoritam za sigurnu komunikaciju putem IEEE 802.11 bežičnih mreža. Slabost algoritma WEP 64 poznat kao i WEP 40 je dosta kratak radni ključ veličine 40 bita dok se preostala 24 bita uzimaju za inicijalni vektor ili kao unutrašnji

radni ključ za RC4 algoritam¹⁰. Pored kratkog ključa druga slabost ovog protokola je sistem za autentifikaciju. Upotrebom brute force metoda za pretraživanje ključa za nekoliko minuta može se dobiti ključ i postati legitiman korisnik mreže.

WPA je napravljen i uveden u upotrebu od strane Wi-Fi Alijanse nakon što je uočena ranjivost starijeg WEP algoritma. Bitna karakteristika WPA algoritma je da radi na uređajima koji mogu koristiti WEP. Poboljšanja koja su uvedena u WPA tiču se šifrovanja komunikacije i autentifikacije korisnika.

Šifrovanje komunikacije je poboljšana korišćenjem TKIP protokola (engl. Temporal Key Integrity Protocol). Za autentifikaciju se koristi EAP (engl. Extensible Authentication Protocol).

WPA2 protokol (engl. Wi-Fi Protected Access) je protokol za sigurnu komunikaciju IEEE 802.11i bežičnih mreža koje su vrlo ranjive na prisluškivanje. Prva verzija WPA ima svoje poboljšanje koje se ogleda u WPA2 protokolu čije se poboljšanje ogleda u uvođenju novog algoritma koji se bazira na AES-u¹¹.

¹⁰ J. Walker. Unsafe at any key size: An analysis of the WEP encapsulation. IEEE 802.11-00/362, 2000.

¹¹ Dissecting the Hack: The F0rb1dd3n Network, Revised Edition; Jayson E. Street, Kent Nabors and Brian Baskin (Jul 15, 2010)

2.2.1. Bežične računarske mreže štićene WEP tipom zaštite

WEP (Wired Equivalent Privacy) je najstariji vid zaštite definisan još davne 1999 godine. Zasnivao na unapred poznatom ključu i obično se koristi u 64-bitnoj i 128-bitnoj varijanti. Pokazao se kao veoma slab i samo daje lažni osećaj sigurnosti jer se uz odgovarajući softver praćenjem saobraćaja između stanica moguće kompromitovati ključ u roku od nekoliko minuta. Kao što smo već pomenuli WEP predstavlja najstariji tip zaštite bežičnih računarskih mreža i u današnje vreme korišćenjem besplatno dostupnih alata relativno lako i brzo dolazimo do pristupnih parametara ovakve mreže. Proces penetracije WEP zaštite predstavimo kroz par koraka komercijalno dostupnih programskih paketa. Oko 25% bežičnih mreža je „obezbeđeno“ WEP protokolom, čije se ključevi vrlo lako mogu kompromitovati uz pomoć besplatnih alata na internetu¹².

Kada klijent komunicira sa pristupnom tačkom (engl. *Access Point*), on šalje i prima pakete u obliku kontrolnih okvira u kojima zaglavlja nisu šifrovana, što omogućava napadaču da vidi inicijalizacioni vektor (IV) koji je korišćen za šifrovanje. Kada napadač "uhvati" dva paketa koji su šifrovani istim IV-om, on može doći do informacija o samim porukama unutar paketa:

$$C_1 \text{ XOR } C_2 = (P_1 \text{ XOR } RC4(v, k)) \text{ XOR } (P_2 \text{ XOR } RC4(v, k)) = P_1 \text{ XOR } P_2.$$

gde je:

- v – inicijalizacioni vektor,
- k – ključ.

Ono što je bitno za obezbeđivanje integriteta poruke je metod koji se koristi kako bi se dobio kontrolni zbir okvira. Naime, integritet se obezbeđuje korišćenjem algoritma CRC kojim se računa kontrolni zbir okvira i dodaje se na kraj podataka koji se štite. U osnovi ovog algoritma se nalazi 32-bitni polinom, 0x04C11DB7 u heksadecimalnom zapisu¹³. Namena

¹² How to hack Wifi or a protected Wireless Network; Markus Smith (May 20, 2012)

¹³ Network Security First-Step (2nd Edition); Thomas M. Thomas and Donald Stoddard (Jan 9, 2012)

ovog algoritma je da očuva integritet poruke u komunikacionim kanalima sa smetnjama razne vrste i kanalima sa šumom. CRC-32 nije najpogodnija varijanta algoritma za šifrovanje jer ne pruža dovoljno dobru kriptografsku zaštitu. U ovu svrhu je boji izbor neka od heš funkcija koja bi bila dovoljno dobro rešenje za ovakve scenarije iz razloga što se baziraju na jednosmernim funkcijama pa presretanje paketa od strane napadača i menjanje sadržaja poruke i kontrolnog zbira radi očuvanja "nevidljivosti" napadača, ne bi bila uspešna. Korišćenjem algoritama kao što su SHA1 ili MD5 se obezbeđuje znatno veća sigurnost od izmene poruke i samim tim jači integritet poruke. Blok podatka koji se dovodi na ulaz za šifrovanje u WEP sistemu, dobija se na sledeći način:

$$P = (M, \text{CRC-32}(M))$$

gde je:

- P – podatak za šifrovanje (otvoreni tekst za šifrovanje),
- M – originalni podatak,
- $\text{CRC-32}(M)$ – kontrolni zbir originalnog podatka.

MD5 se inače koristi za izračunavanje heš vrednosti lozinke na operativnim sistemima. Kada se korisnik loguje na sistem, operativni sistem računa heš vrednost lozinke koju korisnik unosi i računa njen heš kako bi uporedio sa heš vrednošću koja se nalazi u sistemu. Na linuxu u datoteci `/etc/shadow`.

Šifrovanje

Osnova WEP šifrovanja je algoritam RC4. Ovaj algoritam generiše niz pseudoslučajnih bitova kao funkciju ključa k i inicijalizacionog vektora IV ($\text{RC}(IV, k)$). Nad bitovima otvorenog teksta se vrši operacija ekskluzivno ILI (XOR) i nad nizom bitova pseudoslučajne sekvence kako bi se dobio šifrat. Predstavljeno funkcijom, ovaj proces bi mogao da se zapiše na sledeći način:

$$C = P \text{ XOR } \text{RC}(IV, k)$$

- C – šifrat,
- P – blok otvorenog teksta za šifrovanje

Na osnovu dobijenih podataka dalje se može nesmetano kao legitimni korisnik priključiti ovakvoj bežičnoj računarskoj mreži¹⁴.

2.2.2. Bežične računarske mreže šćićene WPA tipom zaštite

WPA tehnički predstavlja sertifikaciju. Ne predstavlja sigurnosni standard obzirom da uključuje samo jedan sigurnosni protokol TKIP. Kombinacija WPA i TKIP protokola nije najbolje rešenje, ali pruža zadovoljavajuću sigurnost. WPA dozvoljava dve vrste autentifikacije: WPA-802.1x (WPA-Enterprise) i WPA-PSK. WPA-802.1x (RADIUS) podrazumeva da postoji RADIUS (Remote Authentication Dial-in User Service) server u mreži. RADIUS server vrši autentifikaciju na osnovu sertifikata koji su ranije dodeljeni klijentskim stanicama i ne dozvoljava pristup nikome ko ne poseduje validan sertifikat. Ova vrsta WPA protokola se koristi uglavnom u većim organizacijama, dok je za SOHO i kućno okruženje zadužen WPA-PSK. WPA-PSK za autorizaciju koristi lozinku dužine od 8 do 63 karaktera koja mora biti uneta i na klijentu i na pristupnoj tački (access point). Ako je lozinka dovoljne dužine (preporučuje se barem 13 karaktera) i složenosti, da bi se izbegao napad sa pogañanjem lozinke iz rećnika ili ispitivanjem svih mogućih kombinacija¹⁵.

TKIP je definisan pod 802.11i standardom i predstavlja omot za WEP šifrovanje, koristi isti algoritam za šifrovanje (RC4), ali ključ mu je dužine 128 bita što rešava problem kratkog ključa kod WEP-a. Bitna karakteristika TKIP-a je to što menja ključ za svaki paket što objašnjava reć "temporal" u imenu protokola. Ključ se kreira mešanjem bazićnog ključa (engl. Pairwise Transient Key), MAC adrese stanice koja šalje podatke i serijskog broja paketa. Mešanjem ove tri stavke se pristupna taćka i stanice opterećuju u granicama normale a kriptografska zaštita je na zadovoljavajućem nivou.

Svaki poslati paket ima jedinstven serijski broj koji se inkrementuje svaki put kada se šalje sledeći paket, takoćde, serijski broj paketa se koristi i kao IV (inicijalizacioni vektor) i kao deo ključa. Smešćanjem ovog broja u ključ se obezbećuje da se svaki paket šifruje

¹⁴ 802.11 Wireless Networks: The Definitive Guide; Matthew Gast (Oct 11, 2011)

¹⁵ Stealing the Network: How to Own the Box; Ryan Russell, Ido Dubrawsky, FX and Joe Grand (May 29, 2003)

različitim ključem. Na ovaj način se rešava još jedan problem kod WEP-a, "Napad kolizijom" kada dođe do šifrovanja dva različita paketa jednim ključem.

Korišćenjem serijskog broja paketa kao inicijalizacioni vektor (IV) rešava se problema napada ponavljanjem jer bi napadač čekao da se pojave dva ista IV-a, međutim, inicijalizacioni vektor sada ima 48 bitova tako da je vreme koje je potrebno da se ponovi isti IV, dosta veće. Kada bi napadač uspeo da dođe do istog IV-a na dva različita paketa i kada bi pokušao ponavljanje, detektovala bi se greška jer se sada za ključ koriste IV kao serijski broj paketa, MAC adresa i bazični ključ. Tako da se ponavljanjem paketa ne narušava bezbednost bežične mreže¹⁶.

2.2.3. Bežične računarske mreže štćene WPA2 tipom zaštite

WPA2 je nadogradnja na WPA koja umesto RC4 algoritma za šifrovanje, koristi algoritam AES koji radi u režimu CBC. WPA2 koristi iste metode za razmenu ključeva, jedino je algoritam za šifrovanje drugačiji. Promenom algoritma je sigurnost podataka povećana, pristup mreži je daleko rigorozniji tako da onemogućava lažno prijavljivanje na mrežu nekim od hakerskih metoda i za sada ne postoje poznati načini za razbijanje WPA2 zaštite. Ono što je neophodno objasniti je način na koji algoritam AES šifrue podatke.

Algoritam RIJNDAEL postao je novi standard za šifrovanje – AES (*Advanced Encryption Standard*). Karakterističan je po tome da ne koristi Feistelove mreže (radi drugačije, runde se razlikuju od Feistelovih) kao i da prilikom konstrukcije supstitucijskih kutija koristi operacije u konačnom polju $GF(2^8)$. Osnovne karakteristike RIJNDAEL algoritma su sledeće:

- veličina bloka za šifrovanje je promenljiva (128, 192 ili 256 bitova),
- dužina ključa je promenljiva (128, 192 ili 256 bitova),
- broj rundi je promenljiv i zavisi od dužne ključa i veličine bloka.

Matrica stanja je matrica bajtova koja opisuje trenutno stanje bloka za šifrovanje; dimenzije matrice zavise od veličine bloka za šifrovanje. Svaka ćelija predstavlja jedan bajt,

¹⁶ Build Your Own Security Lab: A Field Guide for Network Testing; Michael Gregg (Apr 28, 2008)

tj. oktet. Broj redova matrice je fiksna (četiri reda), što znači da svaka kolona ima ukupno 32 bita, tj. 4 okteta. Broj kolona matrice (N_w) zavisi od veličine bloka za šifrovanje i dobija se tako što se veličina bloka podeli sa 32. Na primer, blok podataka dužine 128 bita može se prikazati matricom stanja 4×4 , 192-bitni blok matricom 4×6 a 256-bitni matricom 4×8 .

Ključ za šifrovanje prikazuje se pomoću tabele sa četiri reda i N_k kolona, slično stanju bloka. Za 192-bitni ključ, tabela ključa će imati sledeći oblik 4×6 .

Broj rundi (N_r) je promenljiv, i zavisi od veličine bloka i dužine ključa. Konkretno, broj rundi je određen onim što je duže¹⁷.

Sve operacije algoritma RIJNDAEL izvode se na matrici stanja.

Nakon kopiranja ulaznog podatka, u matricu stanja se inicijalno dodaje potključ (AddRoundKey). Zatim se matrica stanja transformiše 10, 12 ili 14 puta, zavisno od dužine ključa, s tim da se poslednja runda transformacija razlikuje od prethodnih (izostavlja se transformacija MixColumn). Svaka runda algoritma predstavlja funkciju koja se sastoji od četiri transformacije nad oktetima:

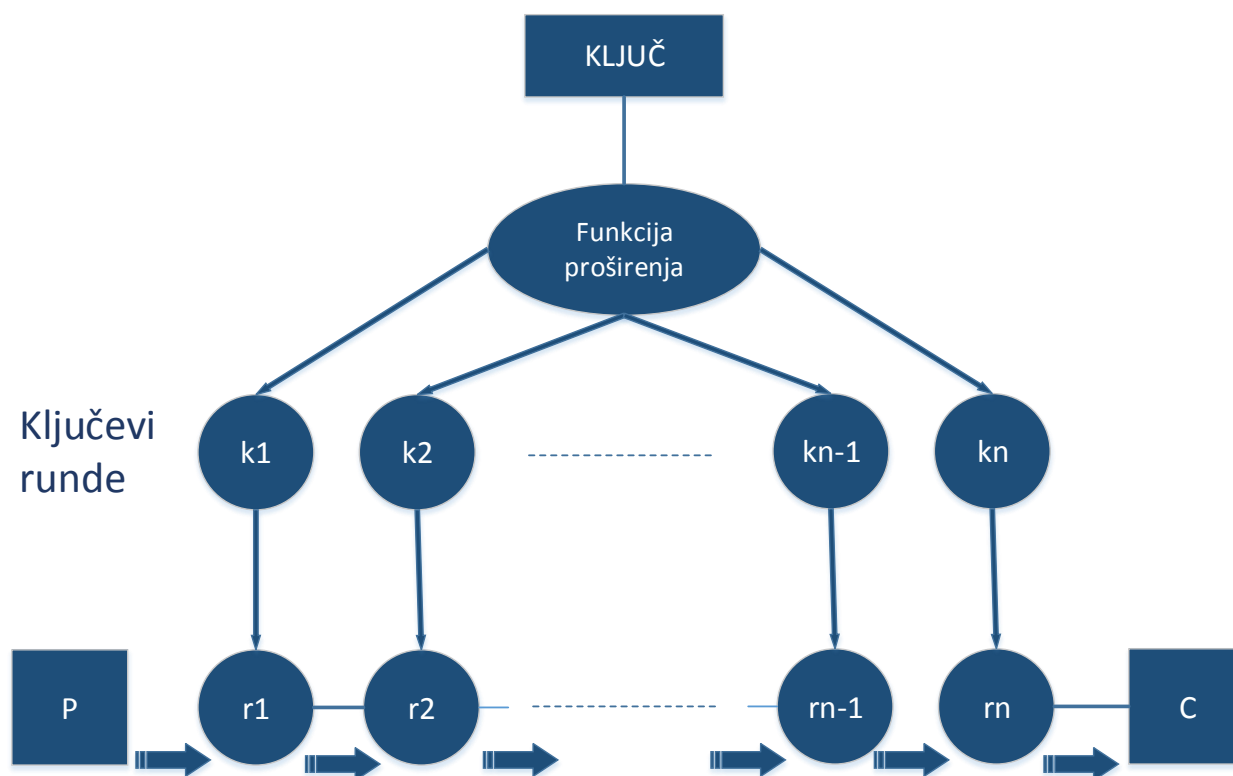
- **ByteSub.** Zamena okteta na osnovu tabele supstitucije. Funkcija ByteSub je jedina nelinearna transformacija.
- **ShiftRow.** Pomeranje (rotiranje ulevo) okteta u redovima matrice stanja.
- **MixColumn.** Transformacije u svakoj koloni matrice stanja. Kolone se posmatraju kao polinomi četvrtog stepena sa koeficijentima iz $GF(2^8)$ i množe sa konstantnim polinomom četvrtog stepena. ShiftRow i MixColumn formiraju linearni sloj koji obezbeđuju veliku difuziju bitova nakon nekoliko rundi.
- **AddRoundKey.** Dodavanje potključa u matricu stanja pomoću operacije ekskluzivno ILI.

Ključ se proširuje kako bi se generisali odgovarajući potključevi za operaciju AddRoundKey. Prve četiri reči proširenog ključa predstavlja ključ k . Svaka sledeća reč se

¹⁷ Stealing the Network: The Complete Series Collector's Edition, Final Chapter; Johnny Long, Ryan Russell and Timothy Mullen (May 18, 2009)

dobija izvođenjem operacije ekskluzivno ILI nad prethodnom reči i reči koja se nalazi N_k pozicija pre tekuće. Proširenjem ključa generiše se $(N_r+1)N_w$ 32-bitnih reči¹⁸.

Potključevi se biraju iz proširenog ključa tako da se prvi međuključ sastoji od prve četiri reči, drugi od sledeće četiri. Dešifrovanje se obavlja pomoću istog algoritma, s tim što se u rundama koriste funkcije inverzne funkcijama ByteSub, ShiftRow i MixColumn, i nepromenjena funkcija AddRoundKey.



Slika 5 : Inicijalizacija ključa u AES algoritmu

Ključ se proširuje kako bi se dobilo n ključeva koji bi opslužili isti broj rundi i to u operaciji *AddRound Key*. Nakon unetog prvog bloka podataka, on prolazi kroz tri operacije u rundi i u poslednjoj (*Add Round Key*) se meša sa podključem, odnosno ključem runde. Nakon poslednje runde dobija se blok šifrata C .

AES algoritam je detaljno objašnjen u radu jer ga koristi WPA2 mehanizam zaštite, koji jeste najpouzdaniji protokol zaštite bežičnih računarskih mreža.

¹⁸ Stealing the Network: The Complete Series Collector's Edition, Final Chapter; Johnny Long, Ryan Russell and Timothy Mullen (May 18, 2009)

2.3. Istraživanje zaštite koju nudi SSL, primena i napadi u bežičnim računarskim mrežama

Sa brzim razvojem mrežnih aplikacija, problem bezbednog prenosa postaje veoma važan. Zbog toga je SSL protokol sve više u širokoj upotrebi u različitim mrežnim servisima. Ovaj protokol sam po sebi nije savršen i u praksi sa njim često postoje problemi. Ovaj deo rada ukazuje na problem koji postoji u procesu SSL uspostavljanja veze. Prva mana kod uspostavljanja veze je što se u početnoj fazi koristi otvorena komunikacija pa postoji mogućnost neovlašćenog praćenja sadržaja¹⁹. Druga mana je implementacija SSLa u aplikaciju²⁰. Zbog razmatranja faktora o radu mrežne veze, obično se koristi prekidač veze zasnovan na HTTP protokolu koji prebacuje saobraćaj na HTTPS. Nedostatke napadač relativno lako koristi, ovaj deo rada predstaviće dva načina falsifikovanja uverenja i konvertovanje podataka u obrnutom smeru od onog koji je inicijalno zamišljen tj. sa HTTPS na HTTP nakon čega se izvršava napad²¹. Eksperimenti u ovom delu rada će pokazati da se sa više od tri metode mogu izazvati značajne bezbednosni rizici za HTTPS komunikacije.

Na primenu SSL-a značajno utiču i razvoj e-trgovine i rad u „oblaku“. Njegovom primenom garantuje se bezbednost komunikacionog procesa, a koriste se mehanizmi autentifikacije od kraja do kraja, šifrovanje poruka, provera integriteta poruke i drugi mehanizmi zaštite. Najveći svetski proizvođači koriste SSL kako bi štitili svoje klijente. Primera radi, Google štiti sigurnost razmene mailova kao i prijave svojih korisnika preko SSL protokola. Amazon štiti svoje transakcije i račune preko SSL protokola. Poslednjih par godina značajan rast beleži i korišćenje sistema rada u „oblaku“. Razmena podataka se u najvećem broju slučajeva obavlja putem SSL protokola. VeriSign, kao najveće autorizaciono telo za izdavanje SSL sertifikata, čak garantuje sigurnu uslugu svim korisnicima nove

¹⁹ El-Hajj, W. (2011). The most recent SSL security attacks: origins, implementation, evaluation, and suggested countermeasures. John Wiley & Sons, Ltd.

²⁰ Bradley, T. (2011, March 24.) Google, Skype, Yahoo Targeted by Rogue Comodo SSL Certificates. Maj, 2011. pcworld.com

²¹ Encryption Is Not Enough: Why It's Time for General Counsel To Weigh In On The Authentication Practices Associated With Secure Communications. Maj, 2011. Redsmith.com

Microsoft Windows Azure platforme u „oblaku“.²² Kada se radi u „oblaku“ svi podaci korisnika i sva obrada podataka obavlja se na udaljenoj lokaciji pa je samim tim neophodno da veza funkcioniše savršeno, ali i da ostane bezbedna kako ne bi došlo do kompromitovanja podataka. Kao što je već više puta pomenuto, bankarski sektor je posebno osetljiv na napade. Ovo je sektor koji je napadačima najzanimljiviji ali i sektor na koji su korisnici usluga najosetljiviji i gde očekuju najveću bezbednost. Savremeno elektronsko bankarstvo se uglavnom oslanja na SSL protokol kao odgovarajući model zaštite.

Ipak pokazalo se da SSL nije savršen za svaki tip primene. Još 2003. realizovani su napadi na SSL protokol i dokazane su greške u uspostavljanju veze koje su omogućavale napad preuzimanja sesije²³. U 2009. godini Michael Howard je uspešno realizovao napad korišćenjem alata Webmitm – alat za Linux okruženje koji uspešno dešifruje podatke iz SSL protokola. Iste 2009. godine na internacionalnoj bezbednosnoj konferenciji predstavljeno je rešenje koje prosleđuje saobraćaj sa HTTPS na HTTP saobraćaj koji se dalje mnogo lakše obrađuje i prisluškuje. Kod ovih napada klijenti ne primećuju razliku u uslugama i servisima koje koriste. Daljim usavršavanjem, rešenje napada na SSL protokol iz 2009. godine postalo je jedno od najčešće korišćenih i biće detaljno opisano i analizirano u ovom delu rada.

Proizvođači Internet pretraživača unapređivali su zaštitu svojih klijenata koja se zasniva na upozorenjima na nevalidne sajtove kao i na lažne sertifikate. Kompanije za antivirusna rešenja razvile su rešenja za zaštitu od ARP prevara, koja se najčešće koriste pri napadima²⁴. Međutim, pored svih zaštita i inicijativa proizvođača programa često su nastajali bezbednosni propusti koji se i danas koriste.

Ovaj deo disertacije kroz analizu bezbednosnih propusta koji se javljaju pri SSL uspostavljanju veze ukazuje na moguće nedostatke. Analiza je rađena primenom dva metoda. Prvi metod zasniva se na karakteristikama otvorenog teksta u fazi uspostavljanja veze kao nedostatu klijent sertifikacione autentifikacije, preuzimanju sesije preko ARP obmane i DNS poplave u okviru mrežnog segmenta²⁵. Napadač podmeće falsifikat serverskog

²² VeriSign Authentication Services - The leading Provider of SSL, <http://www.verisign.com/>

²³ Adi Shamir. Cryptography: State of the science. ACM A. M. Turing Award Lecture, June 2003.

²⁴ Mozilla CA Certificate Policy (Version 1.2). www.mozilla.org/projects/security/certs/policy/

²⁵ Moxie Marlinspike. sslsniff, Jul 2009. www.thoughtcrime.org/software/sslsniff/

sertifikata i predstavlja se kao proksi ili gateway nakon čega dolazi u mogućnost za napad. Posledica je da napadač može da dešifruje sve podatke prenete putem https-a. Drugi metod zasniva se na korićenju defekta SSLa i zaobilaženja istog u praktičnoj primeni²⁶. Na početku se ARP prevarom napadač predstavlja kao lažni proksi ili gateway. Napadač uspostavlja HTTPS vezu sa pravim serverom i ta komunikacija se dalje obavlja sa pravim sertifikatom, dok se veza za obmanutim klijentom obavlja preko HTTP saobraćaja čiji je saobraćaj lako pratiti i obrađivati. Napadač direktno šalje otvoren tekst klijentu sa HTTP strane dešifrovanjem šifrovanih podataka, tako da on može da pristupi svim komunikacijama podataka.

U eksperimentalnoj analizi obrađeno je u odnosu na specifičnosti nekoliko napada po prethodno opisanim scenarijima. Analizirani su verovatnoća uspeha napada i mogućnost klijenta da primeti da je do napada uopšte i došlo. Nakon eksperimenata obavljene su i obuke, nakon kojih su rađene analize obučanih klijenata za detekciju napada i predstavljeni su rezultati grupe pre i nakon obuke.

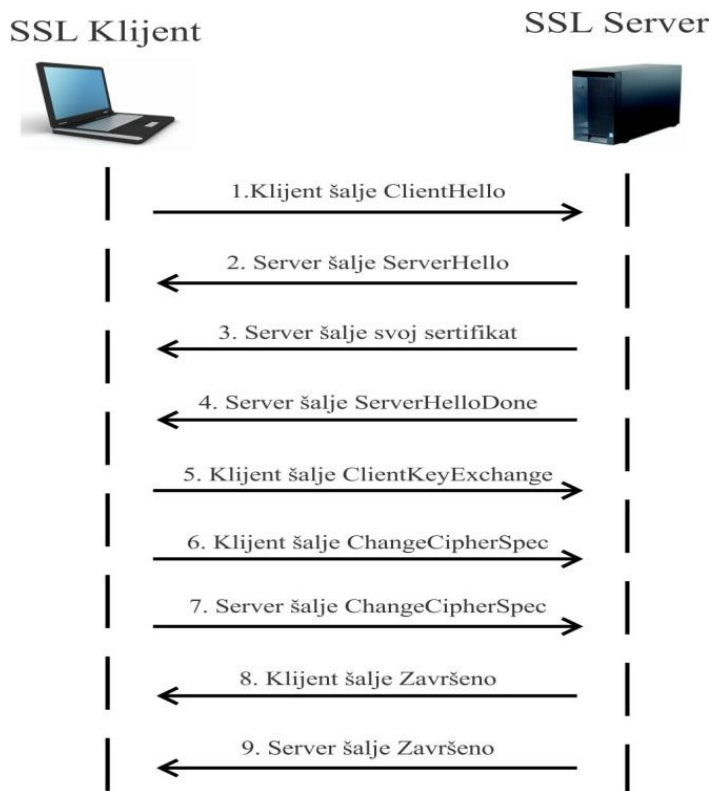
²⁶ Moxie Marlinspike. sslstrip, Decembar 2009. www.thoughtcrime.org/software/sslstrip

2.4. Nedostaci HTTPS uspostave veze

Secure Sockets Layer (SSL) je protokol za sigurno slanje poruka (komuniciranje) putem Interneta. Omogućava slanje poverljivih podataka (npr. broj kreditne kartice) putem Interneta u šifrovanom i sigurnom obliku. Prva verzija SSL protokola je SSL 3.0. Koristi se za širok spektar aplikacija. Uključena je u Internet standardima IETF. Trenutna, verzija SSL protokola u širokoj upotrebi je TLS (SSL 3.1). Proces SSL uspostave veze je jasno definisana u RFC2246, kao što je prikazano na slici. SSL od aplikativnog sloja prima poruku koju treba poslati, rastavlja je u manje delove pogodne za šifrovanje, dodaje kontrolni broj, šifrjuje, eventualno vrši kompresiju, a zatim te delove šalje. Primalac prima delove, vrši se dekompresija, dešifrovanje, proveru kontrolnih brojeva, sastavljaju se delovi poruke i na kraju se predaju aplikativnom sloju. Na taj način se putem SSL-a ostvaruje zaštićeni kanal kroz nebezbednu mrežu. Ukoliko su klijent i server neaktivni duže vreme ili razgovor sa istim atributima zaštite potraje predugo, atributi se menjaju.

2.4.1. Osnove uspostavljanja SSL veze

Na primeru klijent-server komunikacije prikazan je proces uspostavljanja SSL veze.



Slika 6: Uspostavljanje SSL veze

1. Klijent šalje ClientHello: Klijent šalje SSL pozdravnu poruku i pokazuje na početak uspostavljanja SSL veze. ClientHello poruka uključuje veličinu paketa, SSL verziju, slučajni broj koji generiše ključ klijenta, broj sesije, sve kriptografske pakete koji klijentski pretraživač može da podrži, metod kompresije koji može biti usvojen.
2. Server šalje ServerHello: Na primljenu ClientHello poruku poslatu od strane klijenta server odgovara sa porukom ServerHello. Poruka uključuje SSL verziju koju server najviše može da podrži, slučajni broj koji generiše server, broj sesije postavljen od strane servera, metode kompresije koje mogu biti usvojene.
3. Server šalje svoj sertifikat: Server šalje svoj autorizacioni lanac, koji uglavnom uključuje i sertifikat servera kao i viši sertifikat. Sertifikat uključuje svoju verziju, serijski broj, potpis algoritama, informacije potvrde izdavaoca, efikasno vreme, informacije subjekta sertifikata, javni ključ, proširen domen, potpisanu vrednost svih sertifikata.

4. Server šalje ServerHelloDone: Server pokazuje da je sve autentifikacione informacije poslao klijentu u potpunosti.
5. Klijent šalje ClientKeyExchange: Nakon što klijent prođe proveru sertifikata koji je dobio od servera, klijent će generisati pre-master ključ, i korišćenjem javnog ključa sertifikata servera šifrovati pre-master ključ. Zatim se šifrovani pre-master šalje serveru u obliku klijent ključa pomoću razmene paketa.
6. Klijent šalje ChangeCipherSpec: Klijent pokazuje da će sledeći mrežni paketi biti šifrovane od strane uspostavljenih ključeva za šifrovanje, a klijent generiše master ključ koji se koristi u komunikaciji kroz pre-master ključ i informacije koje su se razmenile ranije.
7. Klijent šalje Završeno: Klijent pokazuje da završava poruku u fazi uspostave veze.
8. Server šalje ChangeCipherSpec: Server pokazuje da će sledeća poruka biti šifrovana ključem koji je uspostavljen u toku kriptološke sinhronizacije. Server generiše master ključ ove komunikacije putem pre-master ključa i prethodno razmenjenih podataka.
9. Server šalje Završeno: Server pokazuje da je završio poruke u fazi potvrde uspostave veze.

Nakon ovih koraka, komunikacija obe strane je šifrovana simetričnim šifarskim sistemom primenom uspostavljenog simetričnog tajnog ključa. Šifrovana komunikacija obezbeđuje sigurnost prenetih podataka između klijenta i servera.²⁷

2.4.2. Analiza nedostataka

Sadržaj podataka je otvoren pri SSL uspostavi veze, tako da napadač neovlašćenim pristupom i praćenjem može da dobije paket podataka. Kod SSL-a je moguć napad preuzimanja sesije. Klijent potvrđuje verodostojnost servera u koraku tri koji prenosi serverski sertifikat, ali sertifikat se prenosi u otvorenom tekstu i lako se može presretnuti i neovlašćeno koristiti. Kada napadač presretne poruku i dobije serverski sertifikat, može ga iskopirati i napraviti sopstveni lažni sertifikat, a zatim popuniti informacije sa podacima servera. Pri tom postavlja sopstveni javni ključ i potpis u sertifikat i dalje šalje falsifikovani sertifikat. Kada korisnici prihvate lažni sertifikat tada napad postaje uspešan. Napadač na

²⁷ Jeff Muir, SSL Handshake, <http://citrixblogger.org/2010/09/08/ssl-handshake/>

kraju može da dešifruje premaster ključ koji klijent šalje i više ne postoji tajna za napadača. Dakle, postoji nedostatak u koraku tri pri SSL uspostavi veze (Slika 6)²⁸.

Kao još jedna mana može se iskoristiti i samo poverenje korisnika u SSL i bezbednost sistema. Nakon brojnih analiza došlo se do zaključka da preko 95% korisnika ne obraća pažnju pri ukucavanju Web adrese i ne unosi samostalno HTTPS zahtev. Čak i kada korisnici koriste HTTPS zaštićeni pristup sajtu oni najčešće u svom pretraživaču ne unose adresu sa HTTP ili HTTPS delom, već jednostavno unose adresu u obliku: www.sajt.com. Svi Internet pretraživači će prvo pokušati da uspostave HTTP vezu kao pretpostavljeni tip veze. Tek nakon što se uspostavi veza sa serverom, serveri podešeni za sigurnu vezu će poslati uputstvo sa redirekcijom na sigurni sajt u obliku https://www.sajt.com. Takođe čest je slučaj i da sam sajt u određenim delovima funkcioniše na osnovnom HTTP nivou bez zaštite (ukoliko se radi o prezentacionim materijalima koji ne zahtevaju interakciju korisnika), da bi tek nakon potrebe za interakcijom korisnika zahtevali sigurnu vezu za potrebu autentifikacije. Ovi slučajevi najčešće su rešeni postavljanjem istaknutih linkova koji iniciraju sigurnu vezu²⁹.

²⁸ The Rootkit Arsenal: Escape and Evasion in the Dark Corners of the System; Bill Blunden (May 4, 2009)

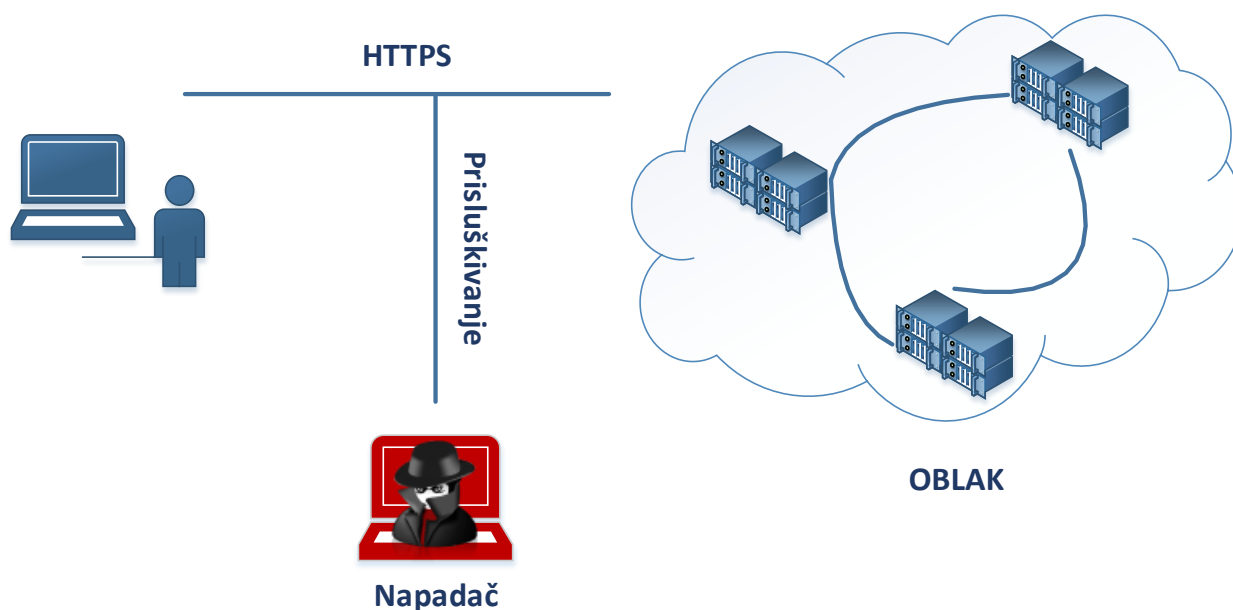
²⁹ Information Security The Complete Reference, Second Edition; Mark Rhodes-Ousley (Apr 3, 2013)

2.5. Napad na HTTPS saobraćaj

Većina saobraćaja funkcioniše preko običnog HTTP saobraćaja iz razloga što HTTPS saobraćaj dodatno opterećuje servere šifrovanjem informacija. Zavisno od slučaja do slučaja usporenja koja korisnik primećuje iznose od 2 do 100 puta. Iz ovog razloga sistemi su automatizovani pa samo bitni podaci koriste HTTPS saobraćaj. Zbog ovakvih i sličnih automatizovanih rešenja koja olakšavaju svakodnevno korišćenje Interneta i njegovih servisa sve manje korisnika obraća pažnju da li je uspostavljena obična HTTP ili sigurna HTTPS veza. Sa strane korisnika sve izgleda isto bez obzira na to da li je veza uspostavljena unosom cele adrese ili samo dela adrese. Sa strane mrežnog saobraćaja stvari ipak funkcionišu drugačije.

2.5.1. Napad prisluškivanjem saobraćaja

Kako se inicijalni prenos podataka odvija u otvorenom tekstu pri SSL uspostavi veze, napadač može da dostavi podatke za ARP prevaru između klijenta i servera. Napadač presreće HTTPS zahtev od klijenta, i povezuje se sa samim serverom. Kada server pošalje sertifikat za autentifikaciju, napadač može da napravi sertifikat koji je sam potpisao, a zatim ga prosledi klijentu. Korisnici imaju tendenciju da izaberu prihvatanje lažnog sertifikata čak i ukoliko ih programski paketi upozoravaju na moguću prevaru. Ovo dodatno olakšava napad, pa napadač lako uspostavlja SSL komunikacioni link. Napad sadrži sledeće korake: lažiranje ARP saobraćaja omogućava umetanje napadača na relaciju server klijent. Tada napadač može da prati saobraćaj, tj. pakete između klijenta i servera. Nakon ARP napada sledi DNS napad i prisluškivanje porta 443 preko koga se uspostavlja SSL saobraćaj. Kada napadač primeti SSL zahtev od strane klijenta, napadač tada prihvata TCP zahtev i inicira SSL vezu sa serverom. Nakon uspešne konekcije na server napadač sprema lažni sertifikat koji sam potpisuje i podmeće klijentu kako bi dalje mogao da dešifruje sav saobraćaj koji prisluškuje. Kada klijent primi lažni sertifikat, može se desiti situacija u kojoj će Internet pretraživač upozoriti da se radi o lažnom sertifikatu ili na mogućnost napada. Kao što je već rečeno u radu istraživanja su pokazala da i pored upozorenja klijenti u ovom koraku najčešće prihvataju lažni sertifikat ukoliko će ih odvesti do željenog sajta. Nakon prihvatanja lažnog sertifikata napad se smatra uspešnim jer i klijent i napadač šifruju sadržaj istim sertifikatom i imaju uspostavljenu SSL konekciju tako da se saobraćaj dalje može lako prisluškivati. Saobraćaj se može pratiti i pregledati u realnom vremenu.



Slika 7: Ilustracija prisluškivanja HTTPS saobraćaja

DNS spoof napad svodi se na lažiranje DNS zapisa, kako bi se korisnik preusmerio na željeni sadržaj koji se nalazi na drugom serveru. Najčešće se koristi u svrhu preusmerenja korisnika na drugi web server. Korisniku će se zatim prikazati lažna web stranica i zabeležiti svi korisnikovi unosi, na ovaj način sve informacije ostaće dostupne potencijalnom napadaču.

Na ovaj način moguće je prikupiti korisnikov korisnički račun za neki web servis ili čak bankovne podatke. Sami tip podataka koje je moguće prikupiti zavisi od web stranice koju je korisnik posetio, a napadač lažirao.

Kako se lažiranjem web sadržaja može u potpunosti preuzeti identitet neke web stranice, sam tim postoji mala je verovatnoća da korisnik primeti nepravilnosti pre nego što unese svoje korisničke podatke, a tada je već kasno. DNS spoof napad moguće je konfigurisati tako da se on prekine po prihvatanju prvih informacija, nakon čega se korisnik preusmerava na pravu web stranicu. Korisnik će verovatno primetiti samo „refresh“ stranice, i eventualno biti zbunjen činjenicom da mora ponovno da unese svoje informacije, koje je nekoliko trenutaka pre toga već uneo i potvrdio.

DHCP exhaust metodom napadač može zameniti postojeći DNS server vlastitim, te na taj način kompromitovati DNS zapise koji će se prosledivati korisniku. Ovakvim napadom se u većini slučajeva napadač izlaže manjem riziku, te može trajnije održavati napad uz minimalne resurse.

Danas postoji dosta specijalizovanih programa za ovu vrstu napada čija rešenja u sebe uključuju DHCP, ARP i DNS napad. Radi se o grafičkim rešenjima koja ne zahtevaju previše poznavanja računarskih mreža i protokola. Ova činjenica omogućava i slabije upućenim pojedincima da postanu napadači, ali omogućava i potencijalno veći broj žrtava. Većina proizvođača Internet pretraživača se protiv ove vrste napada bori primarno detekcijom i sve upadljivijim upozorenjima koja od klijenata zahtevaju veću interakciju, osim do sada uobičajne interakcije od jednog klika.

2.5.2. Napad zaobilaženjem sigurne veze

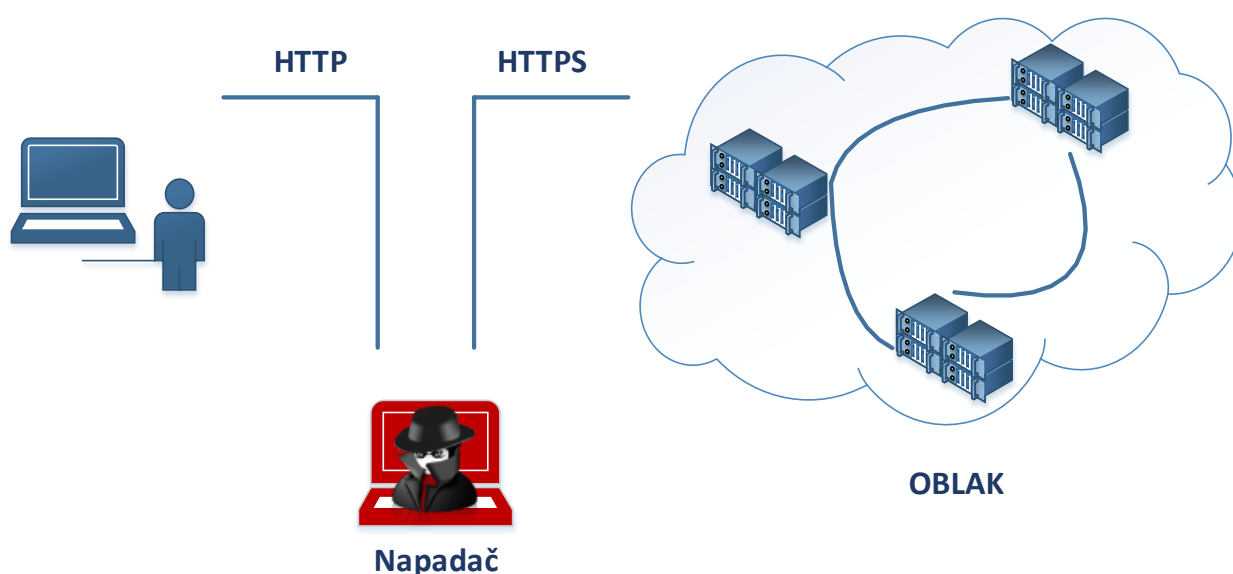
Slično prvom napadu, napadač može da prosleđuje pakete i prati HTTP saobraćaj ARP varanjem između klijenta i servera. Kada napadač primeti da će saobraćaj sa običnog HTTP biti preusmeren na HTTPS saobraćaj, prosleđuje lažni saobraćaj klijentu. Sa druge strane, uspostavlja se lažna HTTPS veza sa serverom. Veza se uspostavlja sa lokacijom koju je klijent stvarno i želeo da poseti, a klijentu se prosleđuje sadržaj željene stranice kako ju je napadač primio, ali se stranica prosleđuje u HTTP obliku bez sertifikata i šifrovanja tako da je sadržaj dalje lako pratiti. Kako su podaci koji se razmenjuju sa klijentom u HTTP obliku i ne postoji provera sertifikata, nema načina da Internet pretraživači uoče bilo kakvu nepravilnost i upozore klijenta, tako da se u ovom napadu od klijente i ne zahteva nikakva interakcija. Samim tim uspešnost ovog napada je znatno izvesnija.

Kao što je već rečeno detekcija ovog napada je znatno teža u odnosu prethodno opisani napad. Ipak ovaj napada zahteva veće poznavanje sigurnosnih propusta i teže ga je implementirati. Napad je grafički predstavljen na ilustraciji (Slika 8) i odvija se kroz sledeće korake:

- Sprovođenjem ARP napada napadač se umeće između servera i klijenta tako što i serveru i klijentu šalje lažne ARP pakete koji obmanjuju i klijenta i server. Nakon toga napadač može serveru i klijentu slati lažne ili validne podatke lažno se predstavljajući.
- Napadač prati HTTP saobraćaj između servera i klijenta, prisluškujući i analizirajući saobraćaj.
- Napadač postaje aktivan tek kada primeti zahtev za HTTPS saobraćajem `<a href="https://...">`, nakon čega zahtev menja sa `<a href="http://...">`. Ukoliko je zahtev predstavljen na sledeći način: `"location:https://..."`, biće zamenjen

sa: "location:http://...". Takođe se pravi lista svih adresa koje su zamenjene radi dalje lakše analize i manipulacije.

- Ukoliko napadač kasnije primi HTTP zahtev od klijenta na osnovu zapisa zna da treba da uspostavi HTTPS komunikaciju sa serverom.
- Nakon uspostavljanja HTTP zahteva sa klijentom, saobraćaj je otvoren (nezaštićen), a jedina manifestacija napada koju klijent može da primeti, ukoliko obrati pažnju, je sam zapis URL adrese u Internet pretraživaču. Pri normalno korišćenju sigurni saobraćaj bi u svojoj URL adresi sadržao https, dok će u slučaju napada u URL adresi sadržati samo http.

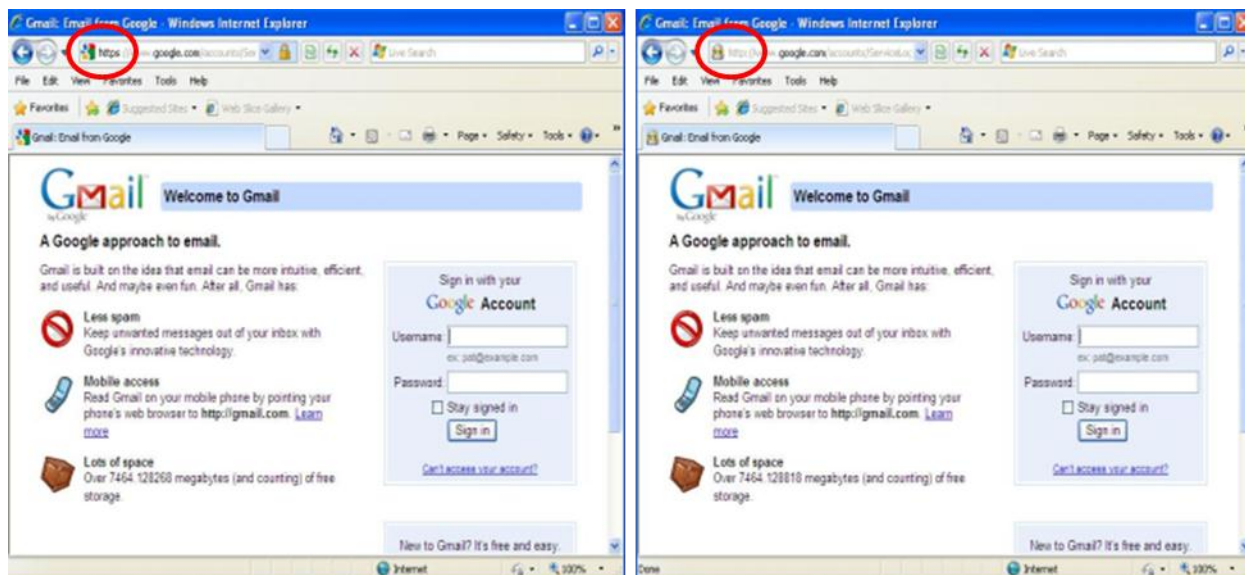


Slika 8: Ilustracija preusmeravanja HTTP i HTTPS saobraćaja

Programi koji su korišćeni u drugoj metodi za APR i DNS napad su isti kao i u prvoj metodi, dok je za uklanjanje HTTPS saobraćaja na strani klijenta korišćen sslstrip i Open SSL biblioteka.

SSL strip je relativno novija metoda MITM napada i postao je prvi izbor u dekriptovanju HTTPS saobraćaja. Za razliku od klasičnog SSL spoof-a koji je korišćen u prvoj metodi napada rešava nas moguć neugodnosti i detekcije. Za razliku od SSL spoof napada, jako je mala verovatnoća da će korisnik uopšte primetiti da se nalazi pod MITM napadom. U slučaju SSL strip-a korisnik dobija čisti HTTP saobraćaj i ne dolazi do poznate SSL greške koja bi korisnika mogla zbuniti ili upozoriti. Iako postoje metode zaobilazjenja SSL greške u slučaju korišćenja SSL spoof-a, njih je najčešće vrlo teško izvesti, pa u krajnjem slučaju i nepotrebno ukoliko SSL strip može obaviti napad kako treba.

Na sledećoj ilustraciji (Slika 9) predstavljena je izgled pristupa Google mail servisu iz Internet pretraživača. Na levoj strani predstavljena je konekcija koja nije pod napadom dok je na desnom ekranu predstavljena konekcija koja se prisluškuje i pod napadom je.



Slika 9: Izgled ekrana Internet pretraživača, bez napada i sa napadom na klijenta

Kao što se iz ilustracije primećuje jako je teško utvrditi razliku između levog i desnog ekrana. Osim vizuelne razlike u jednom slovu klijenti u samom radu neće primetiti razliku u radu. Sa strane klijenta sve funkcionalnosti će ostati na istom nivou dok će brzina učitavanja stranice biti gotovo identična u oba slučaja. Sve ovo utiče na to da je ovakav napad jako teško primetiti pa samim tim i preduprediti.

U slučajevima kada je potrebno "dekriptovati" SSL saobraćaj bez interakcije krajnjeg korisnika, SSL strip će biti prvi izbor. Slabost ove vrste napada je što postoji mogućnost da napredniji korisnici primete kako im se sesija ne otvara putem HTTPS protokola. Ovo bi eventualno mogao biti znak za uzbunu. Ipak generalno gledano, većina korisnika ne razmišlja o tome da li nam je uključen SSL, ako i zna da treba biti uključen. Tako će u slučaju Google maila koji važi za jednog od najvećih mail provajdera, a koji ima uključen SSL za sav saobraćaj biti moguće primetiti da nešto „fali“ u adres baru. Generalno krajnji korisnici neće obraćati pažnju na takve „sitnice“.

Situacija koja je najbolja za napadača je kada se putem HTTPS protokola izvršava isključivo autentifikacija. Ovo je čest slučaj jer na ovaj način sajtovi „garantuju“ sigurnu autentifikaciju a bez daljeg šifrovanja saobraćaj se odvija brže. Nakon autentifikacije saobraćaj se dalje odvija po HTTP protokolu. Primer ovakvih sajtova predstavljaju

www.youtube.com www.facebook.com. Verovatnoća da će korisnik u ovim slučajevima nešto primetiti je svedena na minimum. U svakom slučaju, SSL strip nam ne omogućava samo dekriptovanje lozinki pri autentifikaciji korisnika, već nam ostavlja mogućnost dekriptovanja kompletnog HTTP saobraćaja, odnosno mogućnost generisanja kompletnog HTML sadržaj neke stranice. Na ovaj način može se doći do informacija čak i u slučajevima kada za autentifikaciju nije dovoljno imati samo username i password.

Prikazan napad je posebno interesantan u bežičnim mrežama, jer napadač ne mora imati fizički pristup lokalnoj mreži već se može nalaziti na bilo kojoj lokaciji u okviru dometa bežične mrežne stanice. Kako je analiza dobijana anketom predstavljena u sledećem segmentu rada pokazala, većina klijenata ne pravi razliku u radu u okviru žične i bežične računarske mreže. Analiza je takođe pokazala da većina klijenata ne vodi računa da li se saobraćaj u okviru bežične računarske mreže šifruje, čime dodatno olakšavaju posao potencijalnim napadačima.

2.6. DOS napadi na bežične mreže

Među najčešćim napadima na mrežu su DoS napadi (Denial-of-Service Attacks). Tipičan DoS napad podrazumeva obasipanje mreže nekom vrstom zahteva, kako bi se mreža opteretila i postala nedostupna za legitimne korisnike. Ovi napadi mogu da budu usmereni na različite slojeve mreže, a odbijanje napada ima za cilj izolovanje mrežnih resursa od onih koji je ugrožavaju³⁰.

2.6.1. Tipovi DoS napada

Aplikacija (OSI sloj 7)

DoS napad na aplikacionom OSI sloju se vrši slanjem većeg broja (legitimnih) zahteva aplikaciji u odnosu na broj za koji je ona dimenzionisana (npr. slanje enormno velikog broja zahteva za stranicama na Web serveru). Server obično pokušava da ispuni zahteve korisnika, ali i da održi servise u životu (kako bi oni bili dostupni). U jednom momentu DoS napada, server ne može da odgovori na sve zahteve, pa odbija izvršenje usluga. Mreža i dalje može da se koristi, ali kako serverski procesi ne mogu da odgovore na upite korisnika, pristup servisima je zabranjen.

Transport (OSI sloj 4)

DoS napad na OSI sloju 4 se odnosi na zahteve za uspostavljanjem veze ka hostu. Ovaj tip napada je usmeren protiv TCP/IP steka u operativnom sistemu računara. Najčešće se radi o tzv. SYN poplavi. (SYN paketi su prvi korak pri uspostavljanju veze TCP protokolom.) Napadač šalje veliki broj TCP zahteva za vezom da bi smanjio mogućnosti operativnog sistema da održava TCP sesije. Najveći broj operativnih sistema je limitiran po pitanju broja uspostavljenih veza u jedinici vremena koje može da obradi. DoS napad teži prekoračenju limita, što za posledicu ima zabranu pristupa servisima koji se izvršavaju na hostu. Kao i u prethodnom slučaju, mreža funkcioniše, ali ciljni host ne odgovara.

³⁰ An Investigation into the Detection and Mitigation of Denial of Service (DoS) Attacks: Critical Information Infrastructure; S.V. Raghavan and E Dawson (Sep 27, 2011)

Mreža (OSI sloj 3)

Mrežni sloj se onesposobljava slanjem velike količine podataka na mrežu. Žrtva napada ovog tipa je mrežna infrastruktura. (Na primer, napadač šalje podatke preko 100Mb/s linka na mrežu koja je u stanju da prenese samo 10Mb/s, koja onda odbacuje pakete.) Ovaj saobraćaj takođe može da izazove prevelika opterećenja procesora i da isprovocira dodatne probleme. Napadač stvara ogromnu količinu ICMP saobraćaja na ciljanom mrežnom uređaju (ICMP se koristi za upravljanje radom mrežnih uređaja). Izolovanjem mrežnog segmenta na kome se nalazi žrtva od ostatka Interneta, eliminiše se mogućnost proširenja napada.

Data-Link (OSI sloj 2)

Data-link DoS napad ima za cilj ili pojedinačni računar ili deo mreže. Ovakav tip napada pokušava da onemogući hostu pristup lokalnoj mreži čak i ako su hostovi povezani. Primer za ovo je poplava nesvičovane Ethernet mreže nekorektno formatiranim paketima (frejmovima). Napadač (ponekad to može da bude i loša mrežna kartica) šalje ponovljena zaglavlja frejmova, koji se dalje reemituju po mreži. Ovaj tip napada se ne sreće kod žičnih mreža.

Fizički (OSI sloj 1)

Fizički napadi, takođe, nisu uobičajeni u žičanim mrežama, jer njihov glavni cilj je ugrožavanje prenosnog medijuma mreže.

2.6.2. DoS napadi na bežičnu mrežu

Kritične razlike između napada na žične i bežične mreže manifestuju se kod fizičkog i Data-link DoS napada³¹.

³¹ BackTrack 5 Wireless Penetration Testing Beginner's Guide; Vivek Ramachandran (Sep 9, 2011)

802.11b fizički napadi

Kod bežičnih mreža prenosni medijum je svuda oko nas, te napadač može da preduzme napad i sa udaljenih lokacija. Za razliku od žičanih mreža gde bi posledice ovakvog napada bile evidentne, ovde nema vidljivih tragova napada.

Specifikacija 802.11 PHY definiše ograničeni broj frekvencija za komunikaciju. Meta napadača je uređaj koji radi po standardu 802.11, a cilj da se on zasiti RF šumovima, odnosno smanji odnos signal-šum do nivoa neupotrebljivog. Ako napadač emituje potrebnu količinu RF šuma, bežični uređaj postaje "ućutkan", te više nije u stanju da prihvati validni mrežni signal. U tom slučaju bi i validan mrežni signal generisao šum.

Pravljenje ometača na osnovnoj frekvenciji od 2,4GHz je relativno jeftino i lako. Takođe je moguće kupiti i komercijalne uređaje koji mogu daljinski da ugase bilo koju mrežu. Na nesreću, mnogi bežični telefoni koji rade na 2,4 GHz mogu da dođu u interferenciju sa bežičnom mrežom.

Takođe postoje i problemi različitih mrežnih protokola. Bluetooth na primer, koristi isti ISM pojas kao i 802.11b i 802.11g. DSSS modulacija 802.11b protokola je podložna interferenciji sa modulacijom koja se koristi u Bluetooth mrežama.

802.11b Data-link DoS napadi

Mogućnost direktnog pristupa Data-link sloju daje nove šanse napadačima. Naime, napadač može da pristupi ovom sloju čak i ako je WEP protokol uključen, a ako nije uključena onda ima mogućnost potpune manipulacije akcija između stanica i AP-ova. Ako klijent ne koristi WEP autentifikaciju (ili je napadaču poznat WEP ključ), onda je klijent izložen DoS napadu.

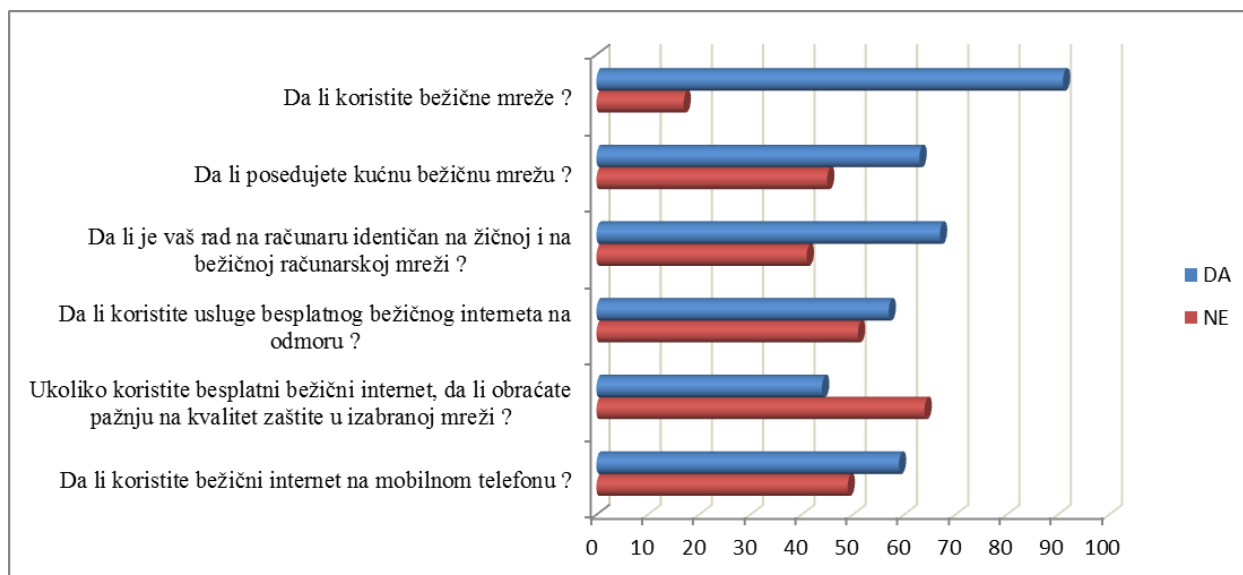
Kod AP-ova sa dva antenska priključka može da se desi da AP ne podesi antene kako treba, pa se opet javlja mogućnost direktnog pristupa Data-link sloju. Uobičajeno je da se dve antene postave u situacijama kada prenos želi da se zakloni zidom, pa antene mogu da budu podešene i različito.

802.11b mrežni DoS napadi

Ako mreža dozvoljava bilo kom klijentu da se asocira, ona postaje ranjiva na napad na mrežnom nivou. Pošto je 802.11 mreža podeljeni medijum, napadač može da poplavi mrežu saobraćajem i tako onemogući pristup drugim uređajima asociranim na dati AP. Primer je sledeći: napadač šalje veliku količinu saobraćaja na AP - ovaj je zagušen samo zbog poplave saobraćaja od strane jednog korisnika - ostali korisnici ne mogu da pristupe preostalom delu mreže jer sa taj AP bori sa velikom količinom saobraćaja koji treba da obradi.

2.7. Analiza podataka prikupljenih nakon održanih savetovanja

Na osnovu predstavljenih analiza i testova obavljena su edukativna predavanja i treninzi koji su imali preko 150 učesnika. Među učesnicima skupova našli su se studenti završnih godina fakulteta ali i poslovni ljudi. Održana su ukupno dva skupa u maju 2011 godine, jedan na teritoriji Srbije i drugi na teritoriji Bosne i Hercegovine. Objedinjeni rezultati anketa koje su učesnici popunjavali predstavljeni su na sledećoj slici.

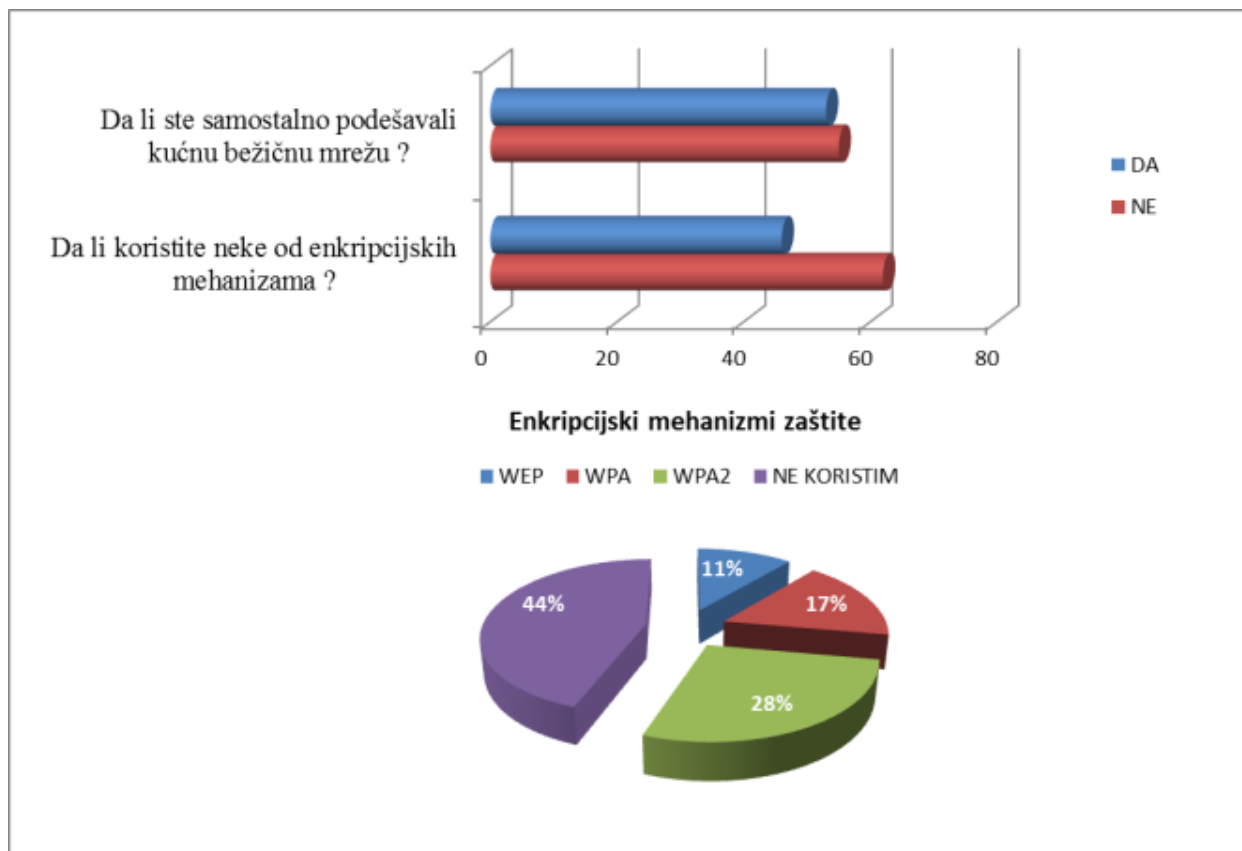


Slika 10: Rezultati ankete održane tokom jednodnevnih seminara

Ono što je u uvodu ovog rada i predstavljeno, anketa učesnika skupa je potvrdila. Preko 90 % učesnika skupa odgovorilo je da koristi bežične računarske mreže. Preko 65% učesnika poseduje kućnu bežičnu računarsku mrežu i ne pravi razliku u radu na bežičnoj i

žičnoj računarskoj mreži. Ono što je istaknuto u ovom radu i njegovim analizama je faktor bezbednosti na koji su posebno ranjivi klijenti bežičnih računarskih mreža. Istina je da se napadi na SSL koji su predstavljeni u radu mogući i u žičnim računarskim mrežama, ali se identitet napadača i postojanje napada mnogo teže detektuje u bežičnim računarskim mrežama. Istraživanje anketom pokazalo je da oko 60% ispitanika koristi usluge besplatnog interneta za vreme svog odmora. Prema mnogim analiza tada se najčešće dešavaju napadi na korisnike. Analiza je takođe pokazala da preko 60% ispitanika neće obratiti pažnju na činjenicu da koristi nepoznatu bežičnu računarsku mrežu, što će ih učiniti posebno ranjivim na napade i kompromitovanje podataka.

Kroz seminare je utvrđeno da preko 65% učesnika seminara poseduje kućnu bežičnu računarsku mrežu i da je oko 50 % njih samostalno vršilo konfiguraciju. Međutim poražavajuća je bila činjenica da 44% učesnika koji su samostalno konfigurisali mrežu izabralo da ne štiti istu od mogućih napada.

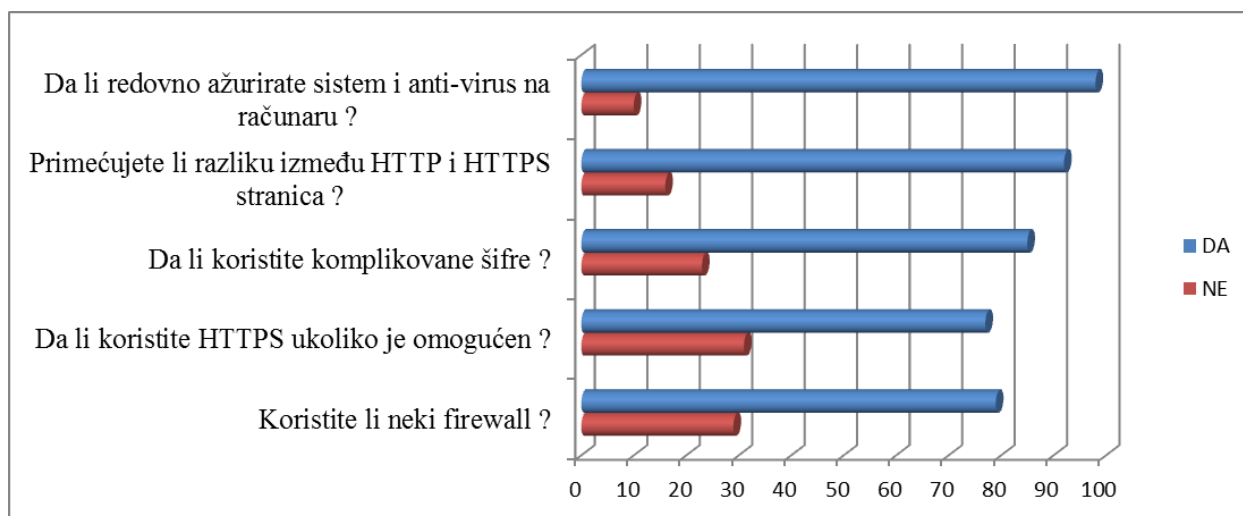


Slika 11: Rezultati ankete učesnika seminara po pitanju zaštite bežičnih računarskih mreža

Kao što je već argumentovano i predstavljeno kroz praktičnu analizu u prethodnom radu autora pod nazivom: Analysis of the wireless network security IEEE 802.11 - The city

of Belgrade³². WEP predstavlja zastareli mehanizam zaštite koji ne nudi dovoljan nivo zaštite i vrlo se lako kompromituje. Pozitivan utisak analize ankete ostavila je činjenica da samo 11% anketiranih koristi stari i prevaziđeni WEP mehanizam koji pruža lažni osećaj sigurnosti.

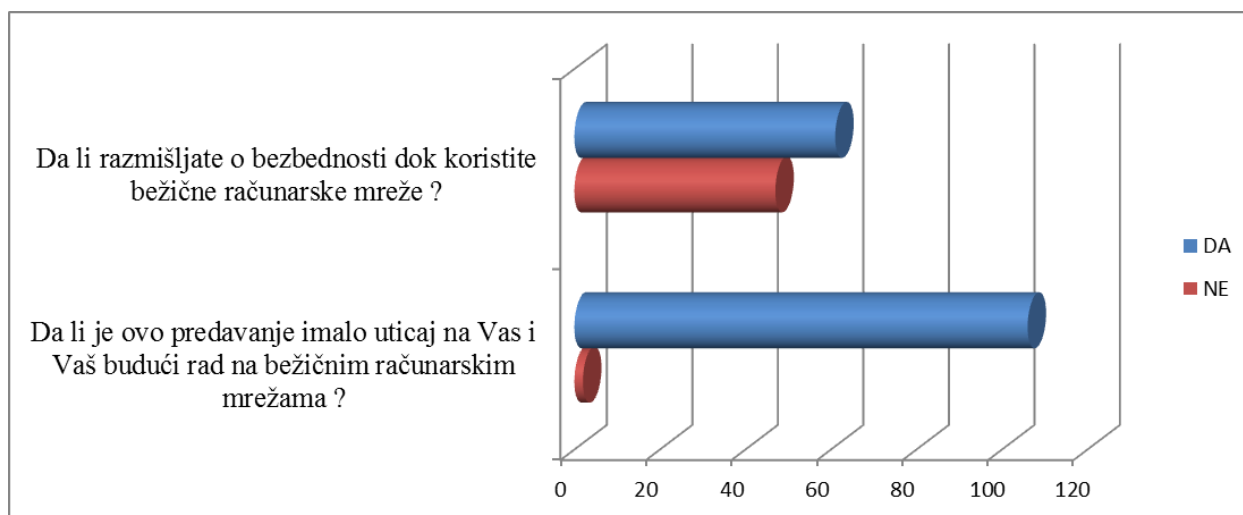
Na ilustraciji (Slika 10) prikazani su rezultati ankete vezani za ažurnost korisnika po pitanja odgovornosti pri korišćenju sigurnosnih proizvoda. U ovom delu ankete veliki broj polaznika seminara odgovorio je da koristi sve savremene metode zaštite od napada. Većina polaznika je upoznata sa sigurnosnim tipovima veza i prepoznaje razliku između HTTP i HTTPS konekcije. Veliki procenat korisnika vodi računa o sigurnosnim zakrpama i nadogradnjama operativnog sistema i anti-virusnog sistema ličnih računara. Dobar pokazatelj podizanja svesti o bezbednosti je svakako i svest korisnika o korišćenju komplikovanih šifri. Komplikovane šifre u toku seminara podrazumevale su šifre duže od 8 karaktera sa uključenim alfa numeričkim karakterima malih i velikih karaktera (case-sensitive) ali i simbolima.



Slika 12 : Upućenost i interesovanje korisnika o različitim zaštitama

Savetovanje je uključivalo i samoevalvaciju, pa je početno pitanje savetovanja bilo: „Da li razmišljate o bezbednosti dok koristite bežične računarske mreže ?“ . Dok je savetovanje završeno pitanjem o korisnosti istog. Rezultati odgovora učesnika dati su na ilustraciji (Slika 13).

³² Analysis of the wireless network security IEEE 802.11 -The city of Belgrade, Saša Adamović, Marko Šarac, Dalibor Radovanović - Univerzitet Singidunum, <http://www.infoteh.rs.ba/rad/2011/B-III/B-III-1.pdf>



Slika 13: Uticaj predavanja na polaznike

Cilj savetovanja je svakako bio da se ova situacija promeni i da se podigne svest učesnika o potrebi za zaštitom lokalnih žičnih i bežičnih računarskih mreža. Cilj savetovanja takođe je bio podizanje svesti polaznicima o prednostima ali i problemima elektronske trgovine i bankarstva. U toku savetovanja urađene su praktične demonstracije napada tako da su polaznici mogli i sami da se uvere koliko su neke od metoda, za koje su smatrali da su potpun bezbedne, podložne kompromitovanju. Takođe, značajno je podignuta svest za potrebom korišćenja kriptografskih mehanizama u bežičnim računarskim mrežama.

2.8. Rešavanje problema

Postoji nekoliko protivmera koje sprečavaju MITM napade na SSL/TLS bazirane aplikacije za internet poslovanje. Veći deo njih je razvijen, kao što je SSL/TLS, pre pojave pecanja (eng. phishing), postoje i tehnike trovanja (eng. poisoning). Na primer, stručnjaci iz oblasti sigurnosti pretpostavljali su da će autentifikacija sa dva faktora sprečiti MITM (kao i druge probleme), ali nedavni napadi su dokazali suprotno.

Istina je da ranjivost na MITM napade, nije zapravo problem autentifikacija korisnika – već je to problem autentifikacije servera. Drugim rečima, ako korisnik na ispravan način autentifikuje server sa kojim uspostavlja SSL/TLS sesiju, u tom slučaju bi bio zaštićen od MITM napada.

Nažalost, ovo nije slučaj i diskutabilno je da li je to uopšte moguće. Napadač koji se služi MITM napadom, može upotrebiti dosta trikova kako bi korisnika zavarao da je

uspostavio vezu sa pravim serverom (npr. visual spoofing). U određenim situacijama, MITM napadač može kontrolisati grafički interfejs korisnikovog internet pretraživača.

Da stvari budu gore, postoje (phishing) sajтови koji imaju važeće sertifikate. Razmotrite situaciju u kojoj napadač zakupljuje važeće sertifikate za www.mountain-america.com i www.mountainamerica.net kako bi spremio napad za web sajt Mountain America Credit Union (www.mtnamerica.org). U takvoj situaciji, većina korisnika ne bi uspela da prepozna da je meta napada.

U prilog ovome, važno je shvatiti da SSL/TLS protokol može zaštititi od MITM napada, ali samo ako korisnik i server poseduju važeće sertifikate i ako se istovremeno identifikuju koristeći ove sertifikate.

Rešenje ovog problema zahteva istraživanje i pronalaženje drugih načina zaštite, daljim razvijanjem SSL/TLS protokola dodatnim načinima autentifikacije korisnika. Jedno rešenje, provera već poslatih podataka, lako se može automatizovati, a i korisniku je bliža za razumevanje. IETF radi na razvoju šifrata koji bi proširio i dodao podršku SSL/TLS protokolu da kontroliše autentifikaciju zasnovanu na već podeljenim ključevima (npr. [S-HOL ET+()5. T-H05, BH06]). Ovakve nadogradnje su važne, ali potrebno je još vremena kako bi ove TLS nadogradnje postale dostupne, implementirane i masovno korišćene.

U praksi, rastući broj aplikacija za elektronsko poslovanje – naročito u Evropi – autentifikacija korisnika se vrši slanjem SMS poruke koja sadrži verifikacioni kod. Kod mora da se unese pri prijavi. Slučaj slanja SMS poruke je primer korišćenja dva komunikaciona kanala za autentifikaciju sa dva faktora (mobilni uređaj je drugi faktor). I ako se tvrdi da ovaj mehanizam štiti od MITM napada, nažalost, to nije uvek slučaj. Ukoliko je MITM napadač između korisnika i servera, napadač nema potrebu za presretanjem SMS poruke. Sve što treba da uradi, jeste da ubedi korisnika da kod koji je dobio prosledi njemu, kako bi nesmetano uspostavio komunikaciju sa serverom. Ukoliko postoji potreba za korišćenjem autentifikacije putem SMS poruke, bolje je ovaj mehanizam koristiti tako da se poruka šalje za svaku transakciju, a ne samo za prijavu korisnika. Svaka zatražena transakcija zatražena od strane korisnika, mora biti ispraćena SMS porukom sa kodom i zatražena potvrda korisnika. Za potvrdu transakcije, korisnik mora ukucati kod.

Ovaj pristup ima svoje loše strane, primetno je skuplji. Dalje gledano, nije naročito pogodan za korisnika, a čak i nije toliko siguran – MITM napadač i dalje uspešno može

zloupotrebiti transakcije koje ne zahtevaju verifikaciju. Osim toga, mora se voditi računa da napadač ne zameni originalni broj telefona svojim, čime bi kontrolisao sve transakcije.

Nekoliko kriptografskih tehnika i protokola je tokom godina predloženo kao prevencija i zaštita MITM napada. Rivest i Shamir³³ predložili su Interlock protokol, ali daljim istraživanjem pokazano je da je ranjiv kao servis autentifikacije. Od strane Jakobssona i Mayersa³⁴ predložena je tehnika nazvana odloženo otkrivanje lozinke (DPD). DPD se može koristiti kao dodatak autentifikaciji zasnovan na lozinci i protokolima za razmenu ključeva, kako bi korisnika zaštitili od specifičnih MITM napada.

Nažalost, DPD zahteva autentifikaciju zasnovanu na lozinci i protokolima za razmenu ključeva, ali korisnika ne štite u potpunosti od MITM napada koji kontroliše interfejs internet pretraživača. Kaliski i Nystrom³⁵ su predložili korišćenje modula za zaštitu lozinke (PPM) kako bi omogućili zaštitu od MITM napada. PPM je aplikacija koja uliva poverenje kada je upitanju autentifikacija zasnovana na lozinci, naime on na osnovu lozinke generiše kod jedinstven svakom korisniku i delu aplikacije. Već pomenuto, PPM ne štiti od MITM napada koji kontrolišu ponašanje internet pretraživača. Dalje gledano, PPM je težak za realizaciju u praksi, kako za postavljanje, tako i za održavanje.

Parno et Al³⁶ predložio je sistem Phoolproof kao prevenciju napada zvanog „pecanje“, ključnu ulogu igra uređaj kome se veruje (mobilni telefon). Asokan et Al predložio je mehanizam zaštite, prošireni autentifikacioni protokol (EAF). Ovaj protokol ima zadatak da korisnike zaštiti on MITM napada na bežičnim mrežama.

Neki istraživači veruju u to, da ako PKI sertifikat bude poboljšán ili ako bi krajnji korisnik mogao sa sigurnošću da potvrdi krajnju adresu do koje želi da stigne, ne bi došlo do situacije da posete lažni sajt i tako budu meta MITM napada. Ali iz iskustva, korisnici jednostavno neće da obrate više pažnje na adresu ili prozore koju im iskaču sve dok njihov put ka krajnjem putu nije zaustavljen. Umesto kriptografskih sistema i protokola ili menjanja

³³ <http://searchsecurity.techtarget.com/definition/RSA>

³⁴ Phishing and Countermeasures Understanding the Increasing Problem of Electronic Identity Theft

³⁵ <http://www.ietf.org/rfc/rfc2986.txt>

³⁶ http://www.intechopen.com/source/pdfs/12460/InTechDistributed_detection_of_node_capture_attacks_in_wireless_sensor_networks.pdf

korisničkog doživljaja internet pretraživača, neki istraživači preporučuju korišćenje više komunikacionih kanala i kombinovanje istih u zaštiti od MITM napada. Kako uvezivanje više kanala za komunikaciju iziskuje veću složenost i dosta smanjuje performanse. Takođe ima veliki negativan uticaj na današnje aplikacije zasnovane na web-u, polako se odustaje od komplikovanih rešenja. Složena i komplikovana rešenja najčešće se koriste samo u slučajevima kada je sigurnost apsolutni prioritet.

Sve predložene protivmere do sada ne rešavaju probleme na adekvatan način ili imaju ozbiljne nedostatke.

3. TEORIJSKE OSNOVE ISTRAŽIVANJA

3.1. Mehanizmi obezbeđivanja stabilne računarske mreže, sistemi za otkrivanje upada

Primarni fokus kompjuterske bezbednosti teži da bude ka prevenciji upada, gde je krajnji cilj da se svi uljezi ovoga sveta zadrže van sistema ili mreže. Autorizacija može da se posmatra kao jedno od sredstava da se spreče upadi, a „zaštitini zidovi“ su sigurno jedna od formi prevencije upada, jer predstavljaju i zaštitu od virusa. Prevencija upada kao informaciona bezbednost je analogna zaključavanju vrata na kolima. Ali čak i ako su zaključana pomenuta vrata, kola i dalje mogu da budu ukradena. Kada je reč o informacionoj bezbednosti, bez obzira na trud koji se ulaže da se sprovede prevencija upada, obično loši momci budu uspešniji i desi se upad³⁷. Šta uraditi kada prevencija upada ne uspe? Sistemi za detekciju upada ili SDU, (IDS - Intrusion detection systems) su relativno skorije otkriće u informacionoj bezbednosti. Svrha ovakvih sistema je da otkriju upade pre, u toku istih ili odmah nakon što se isti dogode. Osnovni pristup koji SDU primenjuje je traženje „neobičnih“ aktivnosti. U prošlosti, administrator bi prošao kroz fajlove logovanja tražeći znake neobičnih aktivnosti – automatska detekcija upada je prirodna nadogradnja ručne analize fajlova logovanja. Nije od velike važnosti reći da je detekcija upada trenutno aktuelna tema istraživanja.

Kao i sa svakom relativno novom tehnologijom, postoje mnoge tvrdnje u ovoj oblasti koje tek moraju da budu dokazane. U ovoj fazi, daleko je od jasnog koliko uspešne ili korisne neke od ovih tehnika mogu biti, posebno kada se iste suoče sa sofisticiranim upadima koji su u porastu. Pre nego što se pređe na diskusiju o osnovama SDU-a, na kratko je bilo pomenuto da je odgovor na upade povezana tema od praktičnog značaja. Dakle, jednom kada dođe do napada, javlja se želja da se odgovori na isti. U nekim od slučajeva dobiju se specifične informacije tako da je razuman odgovor sasvim jasan. Na primer, može se otkriti upad u smislu pogađanja šifre koji je usmeren ka određenom nalogu, a u tom slučaju moglo bi se odgovoriti zaključavanjem naloga. Međutim, isto nije uvek tako očigledno. Dalje će biti navedeno kako u nekim od slučajeva SDU pruža malo specifičnije informacije o samoj prirodi napada. U takvim slučajevima, određivanje pravog odgovora nije lako jer same

³⁷ The Tao of Network Security Monitoring: Beyond Intrusion Detection; Richard Bejtlich (Jul 22, 2004)

specifičnosti napada nisu poznate. U bilo kom slučaju, dalje neće biti reči o odgovoru na upade.

A ko su zapravo „uljezi“ koje SDU pokušava da otkrije? Uljez može biti haker koji je uspeo da se probije kroz mrežne odbrane i koji sada pokušava da izvrši napad na internu mrežu. Ili još podmuklije, upad može da se izvrši od strane „zlog insajdera“, koji može biti neki nezadovoljan zaposleni. Koje vrste napada uljez može sprovesti? Uljez sa ograničenim veštinama najverovatnije će pokušati da sprovede dobro poznati napad ili blagu varijaciju istog. Malo veštiji napadač može da bude u mogućnosti da sprovede znatnu varijaciju u odnosu na dobro poznati napad, ili manje poznati napad ili čak kompletno novi napad. Često će napadač samo iskoristiti sistem kao bazu odakle će sprovoditi napade na druge sisteme. Široko govoreći, postoje 2 pristupa otkrivanja upada³⁸.

- SDU zasnovan na potpisu otkriva napadnu osnovu specifično poznatih potpisa ili šema. Ovo je analogno sa otkrivanjem virusa na osnovu potpisa.
- SDU zasnovan na izuzetku pokušava da definiše osnovu za normalno ponašanje i pruža upozorenje svaki put kada sistem odstupa od nekog normalnog ponašanja.

Naknadno će biti detaljnijih objašnjenja za oba pristupa. Postoje takođe i dve osnovne arhitekture za SDU:

- SDU zasnovan na „domaćinu“ primenjuje metodu/e otkrivanja aktivnosti koje se dešavaju na „domaćinu“. Ovi sistemi imaju mogućnost da otkriju svaki napad koji je vidljiv na „domaćinu“ (npr, prekoračenje tampon-zone ili probijanje privilegija). Međutim, sistemi koji su zasnovani na „domaćinu“ nemaju skoro nikakav uvid u mrežne aktivnosti.
- SDU zasnovan na mreži primenjuje metode otkrivanja u mrežnom saobraćaju. Ovi sistemi su dizajnirani kako bi otkrili napade kao što su odbijanje izvršenja usluge, skeniranje portova, pokušaji uključivanja „zlonamernih“ paketa, itd. Ovakvi sistemi se mahom preklapaju sa zaštitnim zidovima. Sistemi koji su zasnovani na mreži nemaju skoro nikakav direktan uvid u aktivnosti koje se dešavaju na „domaćinu“.

³⁸ Practical Intrusion Analysis: Prevention and Detection for the Twenty-First Century: Prevention and Detection; Ryan Trost (Jul 4, 2009)

Naravno, moguće su razne kombinacije ovih kategorija SDU-a. Na primer, sistemi koji su zasnovani na „domaćinu“ mogu da koriste i tehniku zasnovanu na potpisu kao i tehniku zasnovanu na izuzetku, a sistemi zasnovani na potpisu mogu da iskoriste oba aspekta otkrivanja zasnovanih na „domaćinu“ kao i na mreži.

3.1.1. SDU zasnovan na potpisu

Neuspešni pokušaji logovanja mogu da budu indikativni da se napad sa napadom na šifre upravo odigrava, tako da SDU može da smatra da UN neuspešni pokušaj logovanja nakon M sekundi predstavlja naznaku ili potpis samog napada. Dakle, u bilo kom trenutku kada imamo N ili više pokušaja logovanja koji su se odigrali u roku od M sekundi, SDU izdaje upozorenje da se napad sa napadom na šifre možda upravo odigrava. Ukoliko bi napadač znao da korisnikov SDU izdaje upozorenje kada imamo N ili više pokušaja logovanja koji su se odigrali u roku od M sekundi, onda bi napadač mogao pouzdano da pogodi $N - 1$ šifre svake M sekunde. U ovom slučaju, otkrivanje potpisa bi usporilo napad sa pogađanjem šifre, ali ne bi u potpunosti sprečilo sam napad. Još jedna briga kod ovakve šeme jeste da i N i M moraju da budu podešeni tako da broj lažnih alarma nije prevelik. Mnoge tehnike se koriste kako bi se otkrivanje na osnovu potpisa učinilo malo otpornijim, gde je uobičajen pristup otkriti „skoro“ pa potpise. Na primer, ukoliko se N pokušaja logovanja dese nakon M sekundi, sistem može da ukaže na potencijalni napad napadom slučajnim uzorkovanjem šifre, možda sa određenim nivoom uverenja na osnovu broja pokušaja i vremenskih intervala. Ali nije uvek lako odrediti razumne vrednosti za to „skoro pa“. Statistička analiza i heuristika se mogu pokazati korisnim, ali mnogo veća pažnja treba se posvetiti kako bi se umanjila stopa lažnih alarma. Lažni alarmi će jako brzo oslabiti pouzdanost u bilo koji sigurnosti sistem, bezbednosni sistemi koji stalno upozoravaju na napade kada niko nije zaista prisutan će uskoro biti sasvim ignorisani. Prednosti otkrivanja zasnovanog na potpisu uključuju jednostavnost, efikasnost (pod uslovom da broj potpisa nije prekomeran), kao i odličnu sposobnost da se poznati napadi otkriju. Još jedna prednost je da je upozorenje koje se izdaje specifično, jer se potpis uklapa sa posebnom šemom specifičnih napada. Uz određeno upozorenje, administrator može brzo da odluči da li je potencijalni napad stvaran ili lažan alarm, i u slučaju da je isti stvaran, administrator obično može da odgovori na pravi način. Mane otkrivanja zasnovanog na potpisu uključuju činjenicu da fajl

sa potpisom mora da bude aktuelan, broj potpisa može da se poveća čime se efikasnost smanjuje, i što je najvažnije, sistem može da otkrije samo poznate napade. Sistem zasnovan na potpisu će verovatno propustiti svaki napad sa čak i najmanjom varijacijom poznatog napada. SDU zasnovan na izuzetku pokušaće da prevaziđe nedostatke šema sistema zasnovanog na potpisu. Ali nijedna šema zasnovana na izuzetku ne može biti u današnje vreme kompletna zamena za otkrivanje zasnovano na potpisu. Dakle, sistem zasnovan na izuzetku može da doprinese određenim performansama sistema zasnovanog na potpisu ali ne može u potpunosti zameniti otkrivanje potpisa.

3.1.2. SDU zasnovan na izuzetku

SDU zasnovan na izuzetku traži neobično ili nesvakidašnje ponašanje. Postoji nekoliko velikih izazova koji su svojstveni takvom pristupu. Kao prvo, mora se odrediti šta sačinjava normalno ponašanje sistema, a isto se mora dešavati kada se sistem ponaša uobičajeno. Kao drugo, definicija normalnog se mora prilagođavati raznim promenama i poboljšanjima sistema, inače će se broj lažnih alarma povećati. Treće, tu postoje razni problemi statističkih pragova koji se pojavljuju. Na primer, mora se znati koliko je to nešto neuobičajeno daleko od uobičajenog ponašanja. Statistika je suštinski neophodna pri razvoju SDU-a zasnovanog na izuzetku. Podsećanja radi, sredstvo definiše statističku normu dok varijacija doprinosi načinu merenja distribucije podataka o sredstvu. Sredstvo i varijacija pomažu da se odredi normalno ponašanje. Kako se normalno sistemsko ponašanje može izmeriti? Koja god karakteristika da se izabere za merenje, ta ista merenja moraju se dešavati u trenucima reprezentativnog ponašanja. Posebno je važno da se ne uspostavi ta granica merenja u toku napada jer će se onda napad smatrati normalnim ponašanjem. Merenje neuobičajenih aktivnosti ili, još preciznije, određivanje kako da se odvoji normalna varijacija u ponašanju od napada, predstavlja jednako izazovan problem. Neuobičajeno se mora meriti u odnosu na neku specifičnu vrednost normalnog. Neka bude da neuobičajeno ponašanje predstavlja sinonim za napad, iako u stvarnosti postoje i druga potencijalna objašnjenja za neuobičajeno ponašanje, što dodatno komplikuje situaciju. Tehnike statističkog razlikovanja se koriste kako bi se razdvojilo normalno od neuobičajenog. Primeri takvih tehnika uključuju Bayesian analizu, linearnu diskriminacionu analizu (LDA), kvadratnu diskriminacionu analizu (KDA), neuralne mreže, kao i između ostalog, Sakrivene Markov modele (SMM). Pored toga, neki istraživači otkrivanja izuzetaka koriste tehnike višeg modelinga iz oblasti veštačke inteligencije i veštačkih imunih sistema. U daljem radu dva pojednostavljena

primera otkrivanja izuzetaka će biti uzeta u obzir. Prvi primer će biti jednostavan ali ne i realan, dok će drugi biti malo komplikovaniji ali stoga i realističniji. Pretpostavka je da se nadzire korišćenje tri komande koje su otvori, pročitaj, zatvori. Isto bi predstavljalo normalno korišćenje, dok korisnik koristi seriju komandi koje su otvori, pročitaj, zatvori, otvori, otvori, pročitaj, zatvori. Zbog statističkih podataka, uzeće se u obzir konsektivne komande kako bi se pronašla mera normalnog ponašanja za korisnika. Iz korisničkih serija komandi može se primetiti da, od šest mogućih naručenih parova komandi, četiri para su normalna za korisnika a to su (otvori, pročitaj), (pročitaj, zatvori), (zatvori, otvori), (otvori, otvori), dok preostala dva para (pročitaj, otvori), (zatvori, pročitaj) korisnik obično ne koristi. Ovo opažanje se može iskoristiti kako bi se identifikovalo potencijalno neuobičajeno ponašanje od strane korisnika, koje bi naznačilo da se uljez predstavlja kao korisnik. Onda se može nastaviti nadzor ove tri komande koje korisnik koristi. Ukoliko je odnos neuobičajenih i normalnih aktivnosti isuviše veliki, potrebno je upozoriti administratora da je možda napad u toku. Ovo jednostavno otkrivanje izuzetka može da se poboljša. Na primer, očekivana frekvencija svakog normalnog para može se uključiti u računicu, a ukoliko se posmatrani parovi znatno razlikuju od očekivane distribucije, upozorenje o potencijalnom napadu se može objaviti. Poboljšanje u smislu otkrivanja izuzetka se takođe može postići korišćenjem više od dve uzastopne komande, ili obuhvatanjem drugog modela ponašanja korisnika, ili korišćenjem sofisticiranih tehnika statističkog razlikovanja. Za malo verovatniju šemu otkrivanja izuzetka, potrebno je fokusirati se na pristupu fajlu.

3.2. Sistemi za izviđanje i prikupljanje informacija

Predstavljaju set alata koji se koriste za izviđanje i istraživanje računarskih mreža kao i sistema za koje postoji interesovanje. Takođe se koriste kao prethodnica napada različitih ciljeva, sredstva koja se koriste u svakodnevnom penetracijskom testiranju aplikacija, sistema kao i računarskih mreža.

Česti su slučajevi kada su ispitivač penetracijskog testa i testirana strana povezani, u nekim slučajevima čak i ugovorno. Ovo se radi kako bi se izbegli alati ili podešavanja u alatima koja se označavaju kao “opasna” zbog njihovih potencijalnih štetnih efekata na cilj, takvi efekti mogu da budu prihvatljivi. Potpisivanje ugovora pre samog penetracijskog testa ne mora uvek biti slučaj, ukoliko ne postoji ugovoreno ili plansko testiranje ispitivač svakako želi da ostane neprimećen ili oprezan u nekim scenarijima. Ovakve strategije često ostavljaju otvoreno pitanje upotrebe alata za penetracijsko testiranje na takav način da istraživači obično ne sprovode testiranje penetracije van laboratorijskog okruženja³⁹.

Ukoliko se posmatra sa aspekta vlasništva i proizvođača kategorija alata koji služe kao izvidnica, skeniranje, snimanje infrastruktura, primena sistema i operativnih sistem, možemo uzeti u obzir i izvore istih alata. Velika većina alata koji se koriste u arsenalu oružja koje koriste potencijalni napadači, ispitivači penetracije, ugovoreni testeri i teroristi su besplatni alati i/ili alati otvorenog koda, i isti se obično održavaju i unapređuju od strane baze njihovih korisnika. Postoji takođe i nekoliko komercijalnih alata ili alata koji imaju besplatne komercijalne komponente, od kojih su neki zaista jako dobri ali znaju da budu i poprilično skupi. U slučajevima pojedinaca, korporacija, kriminalnih organizacija i ostalih nedržavnih aktera, češće se u upotrebi opšti ili specijalno kreirani alati. Takvi paketi se često smatraju pretečama mnogo prodornijih formi zloćudnih softvera, napada na sajtove kao i izvorima bilo koje vrste on-line napada slabog intenziteta. Kao takvi, lako su dostupni i na dobrom su putu da učestvuju u ugrožavanju komercijalnih sajtova u širim razmerama a svakako treba da se smatraju ozbiljnom pretnjom. U istreniranim rukama besplatni alati mogu biti jako efikasni,

³⁹ CCNP Security IPS 642-627 Official Cert Guide; David Burns, Odunayo Adesina and Keith Barker (Nov 4, 2011)

ukoliko su manje automatski podešeni, a sam kod kompletno otvoren isti se koriste od strane raznih napadača.

3.2.1. Horizontalno izviđanje – Izviđački alati

Izviđački alati, kao što im sam naziv govori, su oni koji se koriste za sakupljanje informacija, obično u pasivnom stanju, o mrežama i sistemima koji se planiraju zaobići u logičkom smislu. Takvi napori mogu da obuhvate informacije sa javnih sajtova, istraživanje u fajlovima servera imena domena, sakupljanje meta podataka iz dostupnih dokumenata, ponovno uspostavljanje informacija putem upotrebe pretraživača, ili bilo koju od brojnih sličnih aktivnosti.

3.2.1.1. Sakupljanje opštih informacija

Kada istražujemo opšte informacije koje mogu da se iskoriste za pružanje podataka o nekom cilju, postoje brojni izvori koje možemo pretražiti. Mogu se “iskopati” podatci o kompanijama ili pojedincima na veb sajtovima, mogu se istražiti oglasi za posao u potrazi za raznim informacijama, mogu se potražiti licne ili tehničke informacije u rezimeima, mogu se koristiti pretraživači i u opštem i u specifičnom smislu, a mogu se takođe i koristiti specijalizovane pretraživačke alate kakav je Maltego. Po svoj prilici, najbolje je koristiti kombinaciju takvih tehnika kako bi se sačinila što kompletnija moguća slika o cilju.

3.2.1.2. Veb sajтови

Razne interesantne informacije mogu se naći na veb sajtovima pojedinaca ili organizacija. Neke od tih informacija mogu biti prikazane sa namerom, kao npr. informacije o korporativnoj organizaciji, a neke mogu da se preuzmu na nedozvoljen ili na čak na nesvestan način dostupnim mestima.

2007-e godine, Američka Uprava javnih prihoda (UJP) sprovela je internu reviziju popisnih servera sa ciljem da se postara da se sprovedene bezbednosti kao i kontrolne aktivnosti odvijaju kako treba. U toku same revizije, 1,811 neautorizovanih veb servera je otkriveno, a to je činilo 87% ukupnih otkrivenih veb servera. Od svih ovih servera, 661 se koristio u legitimne poslovne svrhe ali su se i dalje koristili van procesa koji bi obezbedili funkcionisanje u bezbednoj konfiguraciji. (Aitoro, 2008) Kada se određene neusaglašenosti

pronađu i u okruženjima koja su teoretski visoko bezbedna, implikacije o sigurnosti servera u manje regulisanim okruženjima zaista predstavljaju zastrašujuću činjenicu.

3.2.1.3. Pretraživači

Pretraživači kao npr. Google mogu da budu od velike pomoći kada se istražuje radi određenog napada. Isti se mogu koristiti kako bi se sakupile informacije u vezi sa posebnim ciljem, kako bi se potražila neka aplikacija ili detalji o određenom hardveru ili čak sakupile specifične informacije kako bi se locirala osetljiva mesta/kritične tačke u ciljanom okruženju.

Još detaljnije, pretraživači mogu da se iskoriste kako bi se sakupili podaci koji se ne pojavljuju u toku regularnog pretraživanja. Takve metode često zahtevaju jako specifično ciljane upite u standardnim pretraživačima ili upotrebu posebno podešenih pretraživača, kao sto je npr. Pipl.com, koji uzvraćaju informacije u okviru specifične oblasti fokusa.

3.2.1.4. Google „napadi“

Google „napadi“ predstavlja upotrebu poboljšanih pretraživača za pretraživanje upita kako bi se omogućila direktnija ciljana pretraga. Iako bi naziv mogao da ukaze da se takva pretraga specifično koristi na Google pretraživaču, u stvarnosti se pretrage sličnih parametara mogu koristiti na bilo kom pretraživaču. Lista takvih operatera može se generalno naći na stranici pretraživača u pitanju. Sto se Google-a tiče, poboljšani pretraživači mogu se naći na:

<http://www.google.com/intl/gn/help/operators.html>

dok se za Bing isti nalaze na:

<http://msdn.microsoft.com/en-us/library/ff795620.aspx>.

U slučaju većine pretraživača, listing unapređenih upita pretraživača može se naći ukucavanjem imena pretraživača i reci “poboljšani upiti”. Iako će se verovatno naći neke razlike u konstrukciji poboljšanih upita koje variraju od pretraživača do pretraživača, sama konstrukcija je često jako slična. Na primer, ukoliko se preko Google-a želi tražiti stranica na singidunum.ac.rs veb sajtu koje sadrži niz “poboljšanih upita” u tekstu stranice, može se sastaviti sledeći pretraživački niz:

[http://www.google.com/search?hl=en&lr=&ie=UTF-8&oe=UTF-8&q=intext:"upis na fakultet" site:singidunum.ac.rs](http://www.google.com/search?hl=en&lr=&ie=UTF-8&oe=UTF-8&q=intext:)

Ukoliko bi se napadač interesovao još detaljnije za pitanja u vezi sigurnosti, pretraga bi mogla da bude sledeća:

[http://www.google.com/search?hl=en&lr=&ie=UTF-8&oe=UTF-8&q=intext:"enable secret 5 \\$" site:singidunum.ac.rs](http://www.google.com/search?hl=en&lr=&ie=UTF-8&oe=UTF-8&q=intext:)

Takve pretrage često će locirati konfiguracione fajlove za Cisco uređaje koji sadrže enkriptovane (a lake za dešifrovanje) administrativne šifre koje se često koriste na ruterima za udaljeni pristup (hackersforcharity.org, 2010). Takvi konfiguracioni fajlovi mogu takođe sadržati i IP adrese za mreže unutar same organizacije i otkriti konfiguracionu šemu kompletne organizacije.

Samo jako mali deo specifičnih informacija koje pripadaju ovakvim vrstama pretrage dostupan je za javnu upotrebu. Knjiga *Google hakovanje za ispitivače penetracije, 2.deo* (ISBN: 978-1-59749-176-1, Syngress) koju je napisao Johnny Long, predstavlja čitavo delo posvećeno ovoj specifičnoj temi. Takođe javno dostupna i kreirana od strane istog autora je i Baza podataka Google napada (BPGH) koju je moguće naći na <http://www.hackersforcharity.org/ghdb/>. BPGH sadrži čitav spektar pretraga u vezi sa sigurnošću a iste čini dostupnim javnosti kroz samo par jednostavnih klikova mišem.

3.2.1.5. Detaljno pretraživanje

Kada pretraživači “tumaraju” Internetom kako bi kreirali indekse na kojima su zasnovani njihovi istraživački rezultati, oni zapravo samo “zagrebu” more informacija koje su dostupne. Postoji ogromna količina neistraženih informacija koje ostanu netaknute baš zbog takve prirode indeksiranja. Kada se obavlja istraživanje o nekom cilju tada bi bilo poželjno da se znaju neke od ovih informacija.

U prethodnih par godina pojavilo se nekoliko specijalizovanih pretraživača koji pružaju uvid u određene delove ovih sakrivenih informacija. Ovi pretraživači su generalno usko specijalizovani za informacije koje pružaju. Jedan od tih alata je i <http://pipl.com>, pretraživač koji je usko specijalizovan za pružanje informacija o ljudima kao i

<http://www.deeppeep.org> , pretraživač koji pruža informacije o veb formama o kojim može da otkrije veb aplikacije koje su dostupne na internetu.

3.2.1.6. Whois

Whois predstavlja alat koji se koristi za upite koji se distribuiraju globalnim bazama podataka koje sadrže informacije o domenu imena širom sveta. Te baze podataka sadrže informacije u vezi sa time kada je domen registrovan ili poslednji put ažuriran, kod kog ISP je isti registrovan, kontaktne informacije o vlasnicima domena, kao i imena servera koji se koriste kako bi se razrešili zahtevi poslati na ime domena. Na Slika 14 može se videti deo odgovora osnovnog upita sa whois. Jedna od najinteresantnijih delova informacija koje su prikazane u okviru slike predstavlja ime servera na koje je ime domena usmereno, što vodi ka dodatnim informacijama u narednom odeljku.

Informacija prikazana na ilustraciji (Slika 14) predstavlja rezultat komandne linije whois upita, alat koji se često može naći u Linux-u i Unix operativnim sistemima, ali ne tako često u ostalim sistemima, kao sto su oni koje distribuira Microsoft. Isti rezultati takođe se mogu prikazati putem upita raznim veb stranicama koje su namenjene toj svrsi, najčešća stranica za isto je svakako whois.net.

U nekim slučajevima, kontaktne informacije koje su pronađene pri whois upitu će sadržati dosta korisnih informacija kao sto su fizička adresa, broj telefona i ime nekoga ko se direktno dovodi u vezu sa domenom. Takve informacije mogu da posluže kao baza za dalje pretrage dodatnih informacija kada se istražuje neki cilj. Prethodnih godina je ipak praksa bila da se domeni registruju putem servisa koji se ponaša kao proksi za domen kontaktnih informacija, time sakrivajući realne, kontaktne informacije onih koji se dovode u vezu sa domenom.



```
root@bt: ~  
File Edit View Terminal Help  
root@bt:~# whois singipedia.com  
  
Whois Server Version 2.0  
  
Domain names in the .com and .net domains can now be registered  
with many different competing registrars. Go to http://www.internic.net  
for detailed information.  
  
Domain Name: SINGIPEDIA.COM  
Registrar: TLDS, LLC DBA SRSPLUS  
Whois Server: whois.srsplus.com  
Referral URL: http://www.srsplus.com  
Name Server: NS1.SINGIDUNUM.AC.RS  
Name Server: NS2.SINGIDUNUM.AC.RS  
Status: clientTransferProhibited  
Updated Date: 12-jan-2012  
Creation Date: 19-feb-2010  
Expiration Date: 19-feb-2015  
  
>>> Last update of whois database: Tue, 16 Apr 2013 08:50:34 UTC <<<  
  
NOTICE: The expiration date displayed in this record is the date the  
registrar's sponsorship of the domain name registration in the registry is  
currently set to expire. This date does not necessarily reflect the expiration  
date of the domain name registrant's agreement with the sponsoring  
registrar. Users may consult the sponsoring registrar's Whois database to  
view the registrar's reported date of expiration for this registration.  
  
TERMS OF USE: You are not authorized to access or query our Whois database to hear  
database through the use of electronic processes that are high-volume and  
automated except as reasonably necessary to register domain names or
```

Slika 14: Whois upit

Pored toga što se mogu obavljati whois upiti o imenima domena, takođe se mogu da obavljati upiti na IP adresama. Informacije sa ovih upita se vraćaju iz baza podataka u vlasništvu Regionalnih internet registara (RIR), koji prate IP adrese koje su dodeljene posebnim regionima. RIR se distribuiraju na sledeći način:

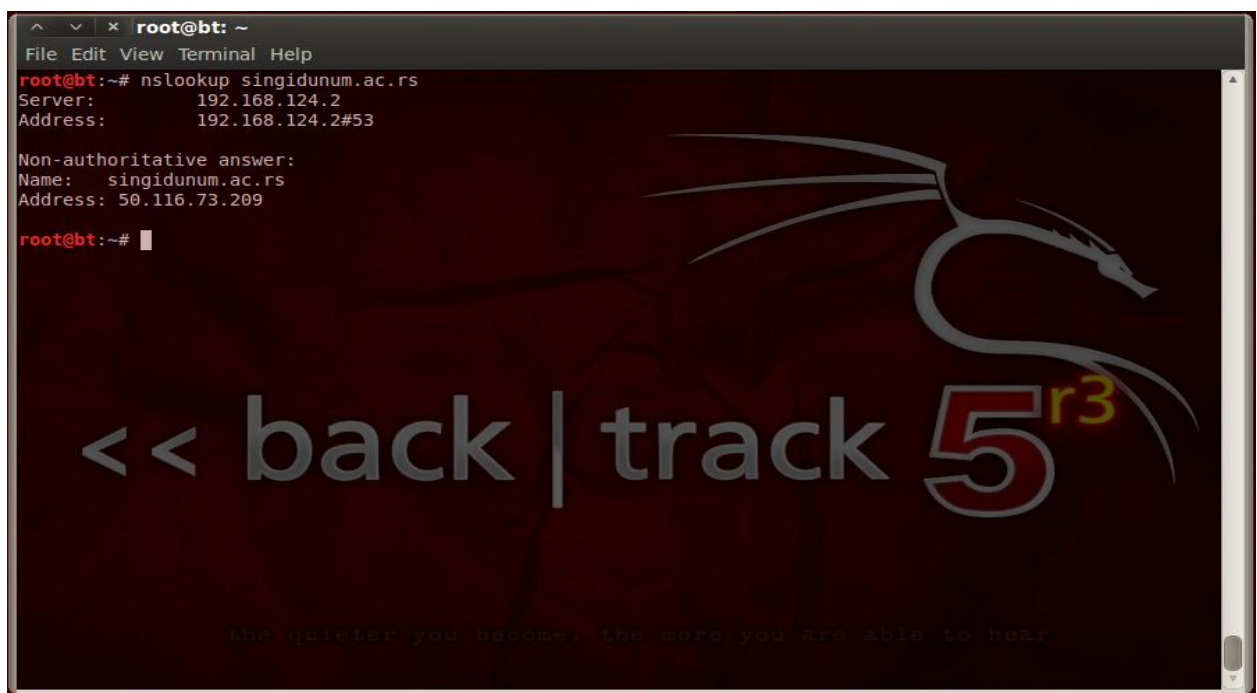
- Severna Amerika i neki od susednih regiona – Američki registar internet brojeva (ARIN).
- Evropa, Bliski Istok i jedan deo Azije – Reaeaux IP evropski centar za koordinaciju mreža (RIPE CKM).
- Azijski Pacifik – Azijsko pacifički informacioni centar mreža (APICM).
- Latinska Amerika i Karibi – Latinsko-američki i karipski registar internet adresa (LAKRIA)
- Afrika – AfriNIC

Takođe se mogu obaviti pretrage i na osnovu informacija koje nisu samo IP adresa, kao npr. mesto kontakta ili ime organizacije.

3.2.1.7. DNS

Nakon prikupljanja osnovnih informacija o cilju sa whois upita koji je pretražen, mogu se dati dodatni upiti i za još informacija. Domen servera imena (DNS) je odgovoran za popunjavanje zahteva za analizu imena za klijente koji pokušavaju da razreše ime domena sa IP adresa.

Brojni alati su dostupni u smislu pomoći oko upita za DNS informacijom. Upit se može sprovesti kroz komandnu liniju u većini operativnih sistema, koristeći nslookup komandu, koja je predstavljena u ilustraciji (Slika 15). Nslookup će u principu vratiti IP adrese sa DNS servera, koje se mogu onda naknadno ispitati u cilju traženja dodatnih informacija.



Slika 15: Nslookup komanda

U idealnom slučaju, izviđanje bi za cilj imalo sprovođenje zonskog transfera u odnosu na DNS server, izazivajući da server pošalje kompletnu kopiju fajla koji poseduje, time omogućujući da se stekne sveobuhvatan uvid u uređaje koje isti poznaje. U nekim slučajevima i dalje je moguće da se koristi komanda nslookup u ove svrhe. U Windows-u, isto se može uraditi sa interaktivnom komandom stanja kao sto je navedeno ispod:

nslookup

- > server [DNS ime servera ili IP]
 - > postavi vrstu/formu=bilo koja
 - > ls -d [ime domena]
- (1)

U većini slučajeva, ovim se neće uspjeti dobijanje informacija koje su tražene jer DNS server ne sprovodi zonski transfer svakom proizvoljnom zahtevu sa interneta. Pored toga, nslookup može da ima onemogućeno funkcionisanje zonskog transfera u nekim operativnim sistemima. Alat sličan nslookup-u, koji se zove kopanje (dig), može se naći u većini Linux sistema i u mogućnosti je da sprovede zonski transfer kada je DNS server u pitanju “voljan da sarađuje”.

Čak i da se ne dobije kompletan zonski transfer sa DNS servera, i dalje je moguće istražiti još mnogo korisnih informacija. Na ilustraciji (Slika 16), mogu se videti rezultate upita komande dig, gde se traže MX fajlovi sa domena singidunum.ac.rs. U ovom slučaju postoji polazna osnovu za dalju istragu na ovom domenu, imajući u vidu da je lociran server mejlova istog.

```

root@bt: ~
File Edit View Terminal Help
root@bt:~# dig singidunum.ac.rs -t MX

; <<>> DiG 9.7.0-P1 <<>> singidunum.ac.rs -t MX
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 44638
;; flags: qr rd ra; QUERY: 1, ANSWER: 5, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;singidunum.ac.rs.                IN      MX

;; ANSWER SECTION:
singidunum.ac.rs.                5       IN      MX      10 aspmx3.googlemail.com.
singidunum.ac.rs.                5       IN      MX      1 aspmx.l.google.com.
singidunum.ac.rs.                5       IN      MX      5 alt1.aspmx.l.google.com.
singidunum.ac.rs.                5       IN      MX      5 alt2.aspmx.l.google.com.
singidunum.ac.rs.                5       IN      MX      10 aspmx2.googlemail.com.

;; Query time: 28 msec
;; SERVER: 192.168.124.2#53(192.168.124.2)
;; WHEN: Tue Apr 16 04:55:53 2013
;; MSG SIZE rcvd: 167

root@bt:~#

```

Slika 16: Prikaz komande Dig

Kao i u slučaju whois fajlova, DNS informacije mogu da se potražuju sa različitih javnih servera na internetu. Jedan od mnogih sajtova koji pruža takvu mogućnost je i dnsquery.org. Pored toga postoji i čitav dijapazon alata koji mogu dati upit DNS serverima, a to čak ide i do sirovog forsiranja putem mogućih poddomena i host-imena, putem alata kao što su dnstenum (<http://pentestlab.wordpress.com>, 2012).

3.2.1.8. [Metapodaci](#)

Metapodaci su zapravo podaci o podacima. Na primer, ukoliko postoji fajl koji sadrži tekst u ovom odeljku, a taj isti fajl ima svoju težinu, datum poslednje posete fajlu i delove koji omogućavaju pristup fajlovima, ništa od ovih podataka nije u direktnoj vezi sa samim fajlom ali predstavlja podatke o fajlu koji je skladištio tekst. Iako takva informacija može da bude sasvim obična van krugova digitalne forenzike, neke od informacija koje se nalaze u dokumentu ili slici metapodataka mogu da budu od velikog značaja. Tu se mogu pronaći stavke poput imena korisnika koji su dopunili fajl, putanje gde je sam fajl bio skladišten, prethodne revizije teksta, koordinate koje ukazuju gde je slika uslikana, “datotekice” sa slikama, ili na još hiljade ostalih stavki informacija koje su sačuvane u fajlu sa zaista usmerenim sadržajem.

3.2.1.9. [Metagoofil](#)

Metagoofil predstavlja odličan alat za nalaženje metapodataka. Metagoofil je skripta koja sprovodi veoma usmerene Google pretrage, koristeći neke od poboljšanih operatera o kojima je bilo reči u prethodnoj sekciji Google „napadi“, kako bi locirala dokumenta koja su skladištena na veb serverima datog imena domena. Na ilustraciji (Slika 17) može se videti početak Metagoofil pretrage na singidunum.ac.rs domenu.



```
root@bt: /pentest/enumeration/google/metagoofil
File Edit View Terminal Help
root@bt: /pentest/enumeration/google/metagoofil# ./metagoofil.py -d singidunum.ac.rs -l 100 -f all -o singidu
num.html -t singidunum_data

*****
* Metagoofil Ver 2.1 -
* Christian Martorella
* Edge-Security.com
* cmartorella_at_edge-security.com
* Blackhat Arsenal Edition
*****
['singidunum_data']

[-] Starting online search...

[-] Searching for singidunum data files, with a limit of 100
    Searching 100 results...
Results: 4 files found
root@bt: /pentest/enumeration/google/metagoofil#
```

Slika 17: Metagoofil skripta

Jednom kada se ova dokumenta lociraju, ista se download-uju i raščlane na interesantne informacije u dokumentu metapodaci, koji se prikazuje u html formatu za lakšu pretragu korisnika. Kao što se može videti iz ovog isečka izveštaja, nekoliko dokumenata sadrže informaciju koja može da bude interesantna pri sakupljanju izvadačkih informacija u svrhu pojedinačnog cilja.

3.2.1.10. [Exiftool](#)

Exiftool je jos jedan koristan alat za prikupljanje metapodataka iz dokumenata i slika. Exiftool je dobio ime od jedne vrste metapodataka koji se zovu EXIF podaci a obično se “pridodaju” na fajlove sa slikama. Ovi podaci mogu da uključuju i informacije o opremi kojom je slika načinjena, uključujući i serijski broj iste, “datotekice” originala slike, koordinate gde je slika načinjena za uređaje koji poseduju GPS, kao i “matičnog domaćina” ostalih informacija. Kada se prave slike sa uređajima koji poseduju GPS, uključujući i skoro sve smart mobilne telefone kao i nekoliko kamera, informacija o lokaciji je vrlo cesto obuhvaćena EXIF podacima fajla slike. Na nekim uređajima ova aktivnost ne može biti onemogućena bez prethodnog onemogućavanja funkcionisanja čitavog GPS-a. Kada se

postavljaju slike za javni uvid, uvek je dobra ideja prvo proveriti EXIF podatke kako bi se zaista videlo šta se deli.

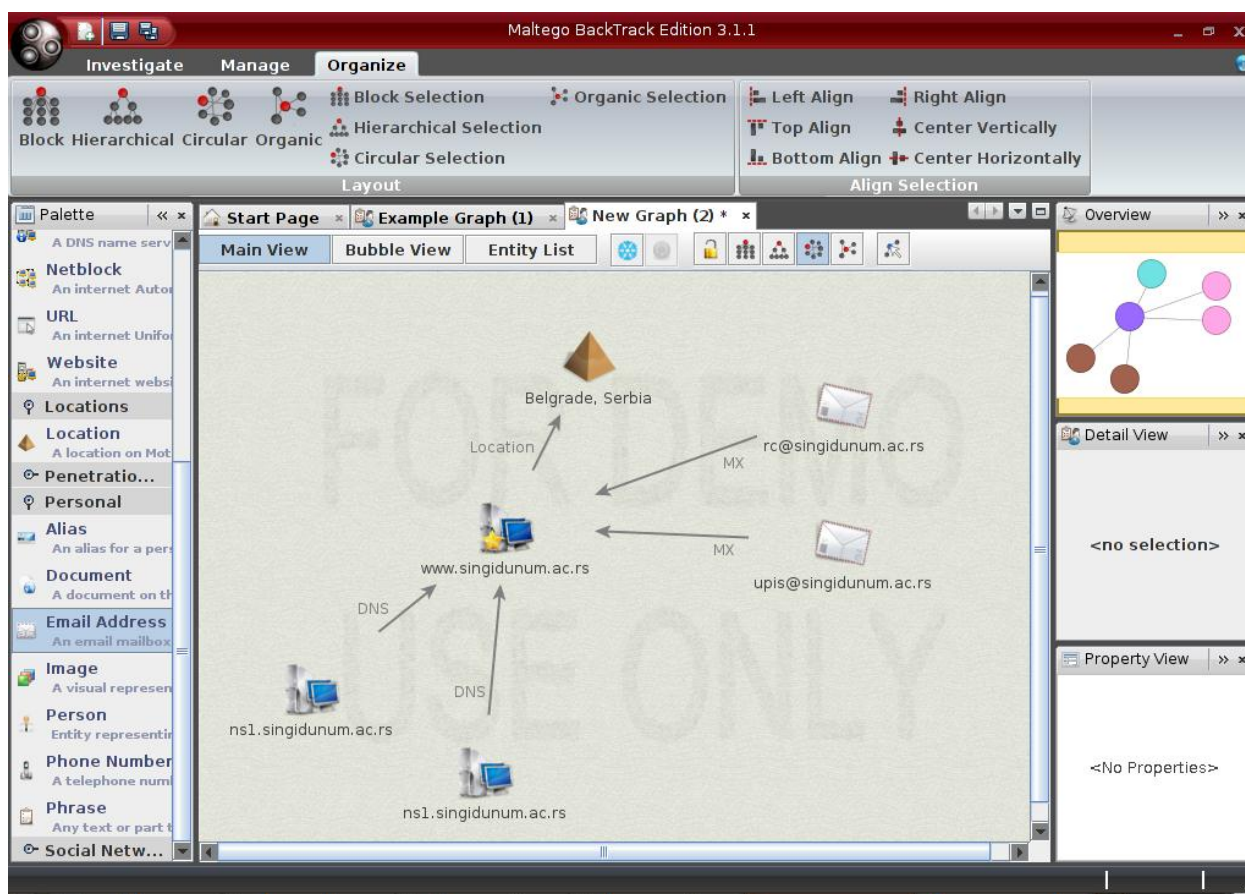
3.2.1.11. [Strings](#)

Strings predstavlja uslužni program koji će raščlaniti dati fajl na nizove teksta, koji se generalno sastoje od nekoliko karaktera u redu koji se mogu štampati. Nizovi mogu biti jako korisni u pronalaženju podataka sakrivenih u fajlovima, pa čak i podataka kao što su izbrisani sadržaji kojima se ne može pristupiti putem normalnih aplikacija koje se obično koriste za pristup i manipulaciju fajlom. Nizovi predstavljaju uobičajeni alat za Linux i Unix distribuciju a isti su dostupni za download na Microsoft operativnim sistemima.

Nizovi se mogu upotrebiti kako bismo locirali metapodatke, ne samo u dokumentima i slikama već i u raznim drugim fajlovima. Iako neki od ostalih alat za rad sa metapodacima možda predstavljaju još efikasnije sredstvo u pronalaženju metapodataka, strings će pronaći sve nizove u datom fajlu. Moguće je i verovatno da će se čak dobiti dosta nebitnih podataka ali će se zasigurno dobiti sve podaci koji postoje u fajlu u formi običnog teksta.

3.2.1.12. Maltego

Kroz prethodne alate diskutovano je o velikom broju podataka koji mogu biti od pomoći kada se sprovodi neka izvidnica o određenom cilju. Takođe je pomenuto nekoliko alata koji mogu da se iskoriste kako bi se sakupile razne stavke takvih informacija. Postoji nekoliko alata koji mogu da sakupe nekoliko stavki informacija, ali jedan od tih alata se posebno ističe u ovoj oblasti: Maltego od proizvođača Paterva. Maltego dozvoljava da se počne sa određenom stavkom informacije, kao što je na primer e-mail adresa, broj telefona ili IP adresa i da se ova informacija iskoristi kao podloga za sakupljanje ostalih informacija. U Maltegu, takvi linkovi među informacijama se nazivaju transformisani linkovi i mogu da budu jako moćno sredstvo za sakupljanje ogromnih količina informacija za veoma kratak vremenski period. Snimak ekrana rezultata takve potrage može se videti na ilustraciji (Slika 18), gde se ime domena koristi kao osnova za dalju pretragu.



Slika 18: Maltego prikaz ekrana

Maltego se može naći i u besplatnim i u komercijalnim izdanjima, varirajući dosta u smislu dostupnih informacija kao i sa time šta se može uraditi sa tim informacijama jednom kada se iste otkriju. Maltego često automatizuje informacije koje bi zahtevale dosta ručnog pretraživanja kako bi se otkrile.

3.2.1.13. Odbrana od horizontalnog izviđanja

Odbrana od različitih alata koji mogu da se koristi kao izvidnica protiv cilja se u principu može opisati jako jednostavnim konceptom: ograničiti izvore podataka kao i podatke koji su dostupni sa svakog od izvora do najvećeg mogućeg razumnog nivoa. Iako možda neće uvek biti izvodljivo da se potpuno obustavi protok izlazećih podataka, a u nekim slučajevima bi bilo čak i potpuno štetno uraditi isto, svakako bilo bi poželjno pokušati kontrolisati informacije kojima se dozvoljava da izađu.

U slučaju informacija dobijenih iz opštih podataka nađenih na veb sajtovima, mogu se ograničiti informacije do određenog nivoa, ali u slučaju poslovanja, ne može se priuštiti ostajanje bez metoda komunikacije koje mogu koristiti organizaciji. Potrebno je koristiti dovoljno opreza da se ne objavljuju previše detaljne informacije, posebno u slučajevima gde je lako odati određene informacije. Takođe je poželjno sprovesti i organizacionu politiku koja bi usmeravala i upozoravala one članove organizacije koji bi mogli da objave osetljive informacije na internim ili eksternim veb sajtovima ili sajtovima socijalnih mreža.

Što se tiče informacija koje mogu da se dobiju sa DNS servera, organizacije su na neki način ograniče u koracima koje mogu preduzeti kako ne bi previše razotkrili podatke. Mogu se odbiti zonske transferi nepoznatim uređajima kako ne bi odavali interne informacije na veliko. Takođe se preporučuje upotreba domenske registracije kao i hosting servise koji su “voljni” da se ponašaju kao proksi za aktuelne informacije tako da se ne dele mrežne ili kompanijske informacije koje mogu biti od koristi za izviđača/napadača. Neke aktivnosti možda nisu uvek odgovarajuće za poslovno okruženje jer možda se ne žele sakriti informacije od potencijalnih klijenata, zavisno naravno od linije poslovanja.

Gubitak podataka putem metapodataka je jedna od oblasti gde, makar gledano sa tehničke strane, organizacije lako mogu ograničiti šta dele sa spoljnim svetom. Aktuelne verzije mnogih alata koji se koriste da kreiraju dokumenta, kao sto su Microsoft Word ili Adobe Acrobat, imaju tu funkcionalnost ugrađenu u samu aplikaciju, da mogu da prečiste

metapodatke iz istih pre nego ih objave van organizacije. Ovi alati često obavljaju jako dobar posao ali se često i isplati proveriti isto sa nekim sekundarnim izvorom, kao što su uslužni programi nizova, dešava se da neke informacije koje organizacije ne želi da objavi promaknu. S obzirom da slike često ne podležu istom procesu kao bilo koji drugi dokument, poželjno bi bilo da organizacije takođe koriste nešto slično Exiftool-u kako bi se postarale da nisu greškom uključili i neku informaciju koju nisu nameravali da podele.

3.2.2. Vertikalno izviđanje - Alati za skeniranje

Alati za skeniranje predstavljaju kategoriju alata koji se koriste kako bi se saznalo više informacija o ciljanom okruženju sistemima, kao i detaljima tih sistema. Informacije prikupljene ovakvim alatima mogu biti jako opšte, u slučaju da se sprovodi mrežno mapiranje; specifičnije, u slučaju da se skeniraju portovi ili jako usko specifične u slučaju da se prisvajaju informacije ili informacije o korisnicima na pojedinačnim sistemima⁴⁰.

U ovom delu, grupisano je mrežno mapiranje, skeniranje portova kao i alati za brojanje u okviru jedne sekcije. Svaka od ovih oblasti može biti detaljno obrađena, ovde su prezentovane samo kroz neke od najvažnijih stvari.

3.2.2.1. Nmap

Nmap predstavlja izuzetnu alatku. U principu to je skener portova ali isti može da uradi i mnogo više. Može se iskoristiti da uspostavi IP-jeve, odredi osetljiva mesta, “uzme otiske prstiju” operativnim sistemima, sprovede rutu traga i još mnogo toga. Skoro sve upotrebe za koje nmap može da se koristi mogu da se postave na takav način da mogu da se povuku kako bi se izbegla detekcija, izmenila brzina kojom se obavljaju procesi, promene metode komunikacije i sl. Nmap je zaista svestrana alatka. Pored toga, nmap predstavlja besplatan alat koji već postoji u mnogim Linux i Unix operativnim sistemima. Nmap je takođe dostupan i za Windows. Pored verzije komandne linije koju će biti predstavljena, postoje takođe i razni GUI koji mogu da se iskoriste kao preduslov za Nmap, uključujući i Zenmap koji je kreiran od strane autora Nmap-a.

⁴⁰ Networks Security and Communication: Horizontal and Vertical HANDOFF; Mohammad Faisal (Aug 30, 2012)

Zavisno od toga šta je zapravo cilj kada se sprovedi Nmap, komanda može da se napravi na više načina. Kako bi se sprovedlo osnovno mrežno mapiranje na pordmreži, može se uraditi nešto slično sledećem:

```
nmap -sP -n 10.0.0.1-254
```

(2)

Ovaj primer sprovodi osnovnu proveru svakog IP-ja u specifičnim obimima kako bi odredio ukoliko nešto odgovara skeniranju po osnovnim portovima (-sP), a ne pokušava da razreši imena (-n), čime će pretraga biti ubrzana. Rezultat istog može se videti na ilustraciji (Slika 19). Ukoliko bi bilo potrebno da se dobije više informacija, komanda bi mogla biti izmenjena tako da sprovede mrežno mapiranje kao što sledi:

```
Nmap -sT 10.0.0.1
```

(3)



```
root@bt: /
File Edit View Terminal Help
root@bt:/# nmap -sP -n 192.168.110.1-254

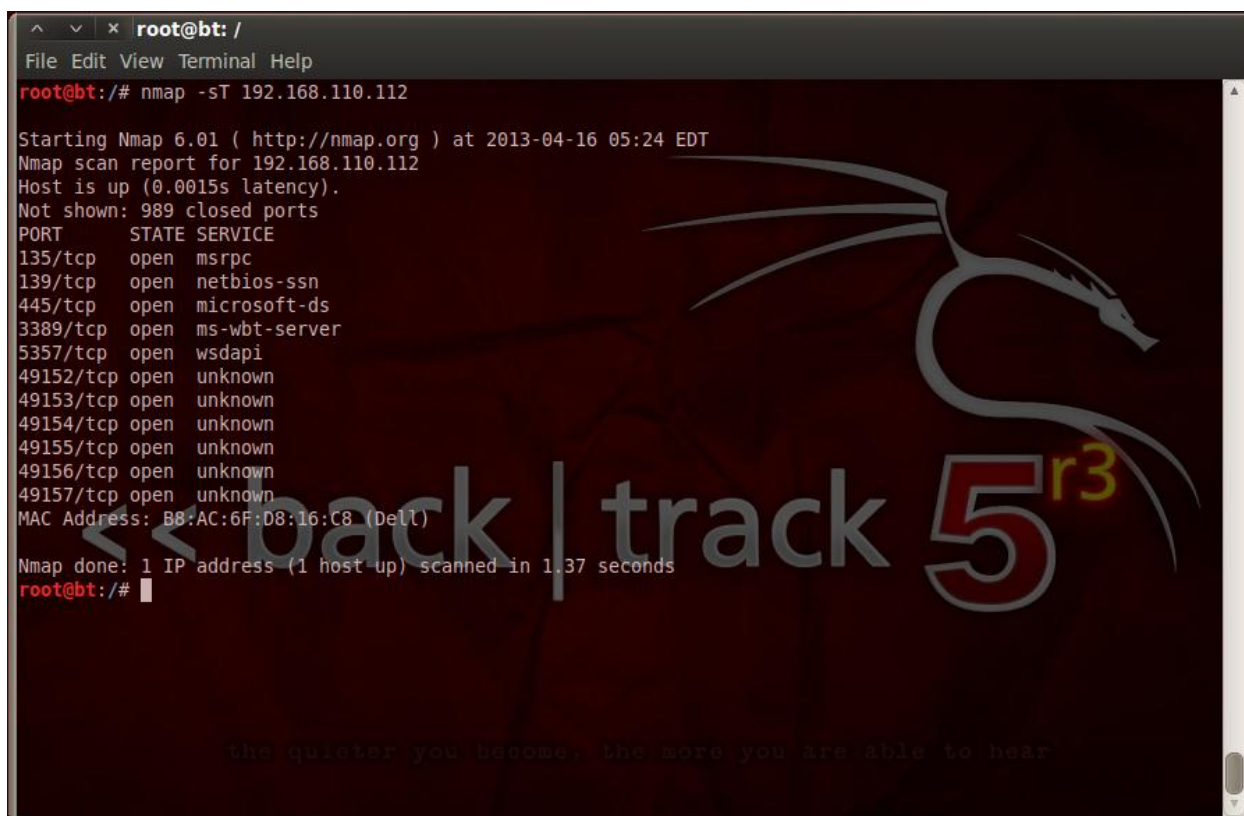
Starting Nmap 6.01 ( http://nmap.org ) at 2013-04-16 05:23 EDT
Nmap scan report for 192.168.110.7
Host is up (0.00041s latency).
MAC Address: 00:1B:11:F0:FF:D4 (D-Link)
Nmap scan report for 192.168.110.9
Host is up (0.00024s latency).
MAC Address: 00:50:8D:91:A6:EF (Abit Computer)
Nmap scan report for 192.168.110.10
Host is up (0.00054s latency).
MAC Address: 00:15:5D:73:B0:00 (Microsoft)
Nmap scan report for 192.168.110.21
Host is up (0.00069s latency).
MAC Address: 00:15:5D:73:AD:01 (Microsoft)
Nmap scan report for 192.168.110.22
Host is up (0.00020s latency).
MAC Address: 00:14:5E:45:8E:68 (IBM)
Nmap scan report for 192.168.110.56
Host is up (0.00048s latency).
MAC Address: 1C:6F:65:94:C9:F7 (Giga-byte Technology Co.)
Nmap scan report for 192.168.110.61
Host is up (0.00054s latency).
MAC Address: 50:E5:49:9A:0C:CF (Giga-byte Technology Co.)
Nmap scan report for 192.168.110.64
Host is up (0.00063s latency).
MAC Address: D0:67:E5:05:5D:79 (Dell)
Nmap scan report for 192.168.110.65
Host is up (0.00055s latency).
MAC Address: D0:67:E5:05:27:D7 (Dell)
Nmap scan report for 192.168.110.66
```

Slika 19: Komanda Nmap osnovno skeniranje

Ovim bi se i proverio svaki pojedinačni IP kako bi se videlo dali ista odgovara, ali i sprovedlo skeniranje portova uz pomoć standardnih podešavanja kada se uređaj pronade (-sT).

Nmap će standardno skenirati 1000 najčešće korišćenih portova. Kao što se može videti i na ilustraciji (Slika 20), ovakva komanda omogućava uvid u dosta korisnih informacija.

Kada se sprovodi i mrežno mapiranje i skeniranje portova, može se uočiti poprilična razlika u količini vremena koje svaka od operacija zahteva. S obzirom na mali obim IP-jeva koji je skeniran u primeru, mrežno mapiranje će vratiti informacije za minut ili nešto duže, dok skeniranje portova može da potraje i satima.



```
root@bt: /
File Edit View Terminal Help
root@bt:/# nmap -sT 192.168.110.112

Starting Nmap 6.01 ( http://nmap.org ) at 2013-04-16 05:24 EDT
Nmap scan report for 192.168.110.112
Host is up (0.0015s latency).
Not shown: 989 closed ports
PORT      STATE SERVICE
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
3389/tcp  open  ms-wbt-server
5357/tcp  open  wsddapi
49152/tcp open  unknown
49153/tcp open  unknown
49154/tcp open  unknown
49155/tcp open  unknown
49156/tcp open  unknown
49157/tcp open  unknown
MAC Address: B8:AC:6F:D8:16:C8 (Dell)

Nmap done: 1 IP address (1 host-up) scanned in 1.37 seconds
root@bt:/#
```

Slika 20: Korišćenje komande Nmap za mapiranje portova

Nmap može poslužiti i za kompleksnija skeniranja, kao što je prethodno pomenuto ona će oduzeti više vremena ali će i dati detaljnije rezultate. Jedan složeni prekidač koji uključuje i nekoliko drugih jeste A prekidač:

`nmap -A 10.0.0.1-254` (4)

Ovim će se izvršiti skeniranje lokalnog IP opsega dok se istovremeno sprovodi “uzimanje otisaka prstiju” kao i detekcija verzija operativnog sistema u odnosu na 1000 najčešće korišćenih portova.

Alat Nmap predstavljen je sa osnovnim funkcionalnostima. Postoji još mnogo prekidača koji omogućavaju razne karakteristike i funkcionalnosti, bez toga da čak idu dalje od upotrebe standardnih delova alata. Pored standardnih, može se upotrebiti Nmap masina za skriptovanje (NSE) kako bi se proširile funkcionalnosti nmap-a da obavi još dosta korisnih stvari. Autor Nmap-a (Fyodor), napisao je odličnu knjigu o širokom dijapazonu stvari koje ova alatka može sprovesti a ista se zove Nmap mrežno skeniranje: zvanični Nmap projektni vodič ka mrežnim otkrićima i bezbednosnom skeniranju – ISBN-13: 978-0979958717.

3.2.2.2. [Nessus](#)

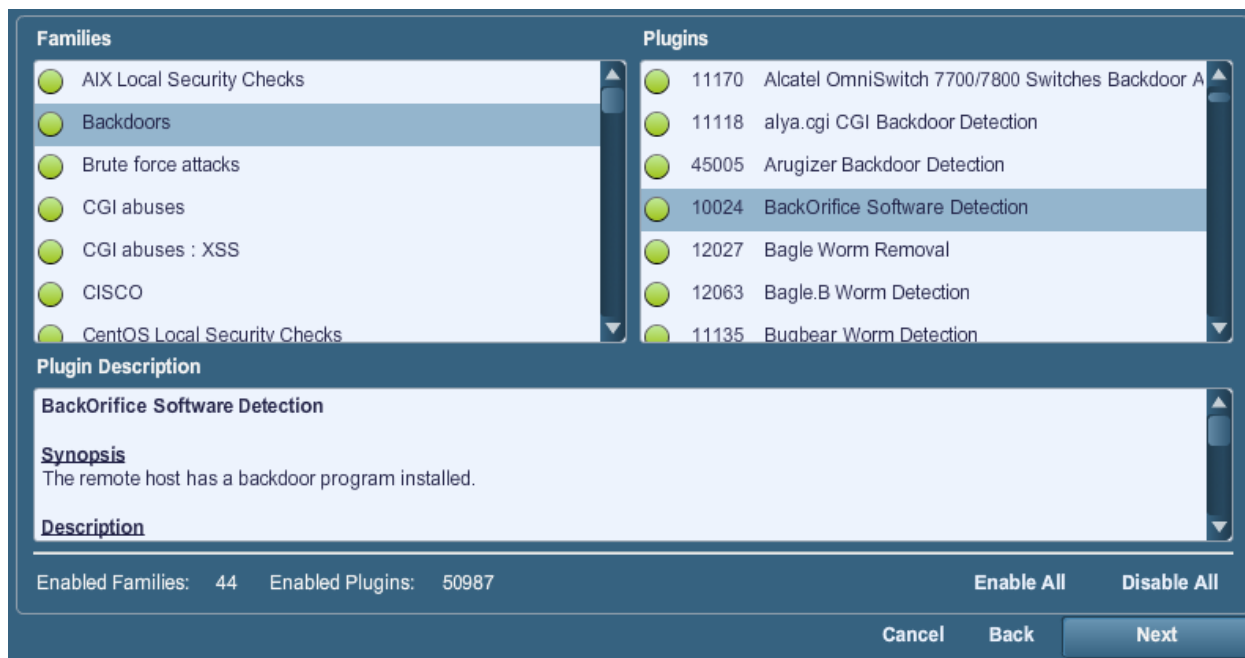
Nessus je prvenstveno predstavljao alatku za skeniranje osetljivih mesta, ali kao što je objašnjeno i kod Nmap-a, čitav set raznih karakteristika su mu se dodale tokom godina kako bi se poboljšala upotreba istog. Nessus je nekada bio apsolutno besplatna alatka otvorenog koda. 2005-e Nessus je dobio dozvolu da postane alatka zatvoreno koda, i određene karakteristike su dobile ograničen pristup u komercijalnoj verziji. Besplatna verzija istog je i dalje dostupna, ali je ograničena u smislu slučajeva u kojima može biti upotrebljena kao i u smislu listinga osetljivih mesta kojima je pristup dozvoljen. Alternativno rešenje otvorenog izvora jeste kreirano i biće pomenuto nakon predstavljanja Nessus rešenja.

Nessus grupiše osetljiva mesta u setove dodataka, sa svakom grupom dodataka koji se fokusiraju na određeni tip kritičnih tačaka. Ove grupe obuhvataju set različitih operativnih sistema, bazu podataka, protokola i usluga. Postoji profesionalni menadžer dodataka koji obuhvata brz pristup najnovijim dodacima, kao i određenim rezervisanim kategorijama dodataka, kao sto su oni za detekciju osetljivih mesta u sistemima, što je i predstavljeno na ilustraciji (Slika 21).

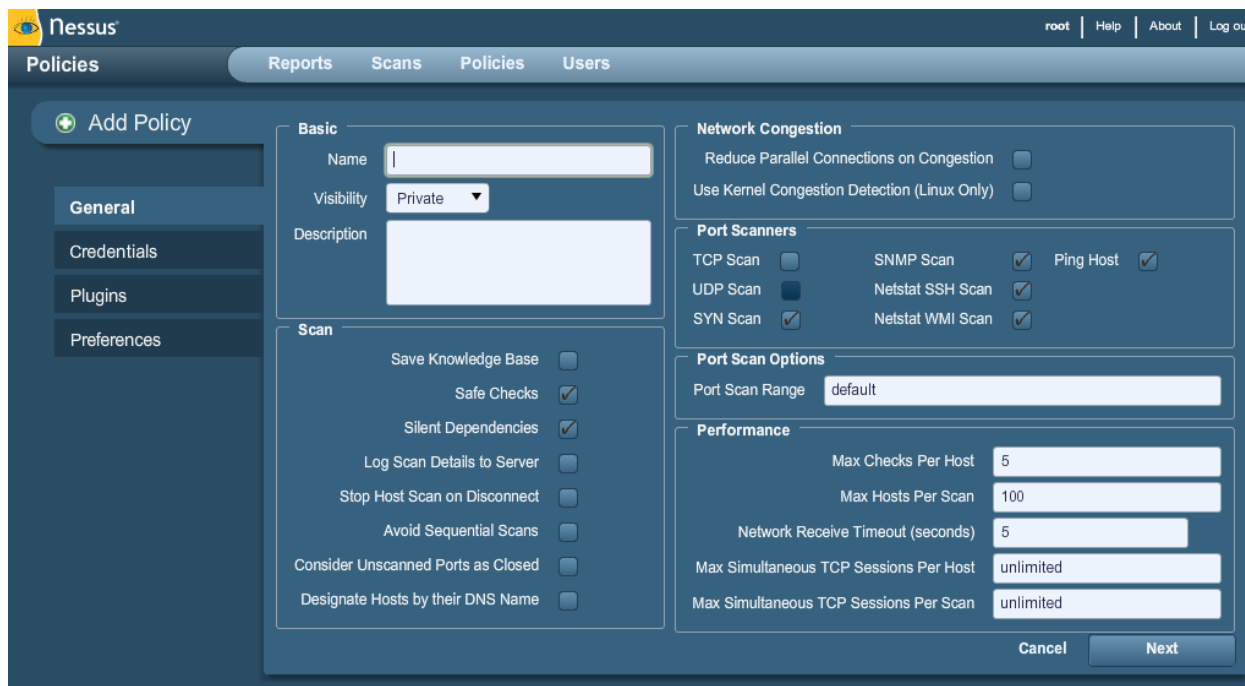
Ova posebna grupa dodataka uključuje i neka od osetljivih mesta koja koristi Stuxnet crv, koji je dospeo pod lupu javnosti 2010. zbog svojih napada širokog spektra na SCADA sisteme u mnogim zemljama.

Najlakši način da se Nessus upotrebi, zbog same kompleksnosti proizvoda, jeste putem GUI-a. Iako su ranije verzije Nessus klijenta isticali klijenta kojeg su same sadržale, trenutna verzija istog, o kojoj je ovde i reč, dostupna je putem pretraživača mreža. Na ilustraciji (Slika 22), može se videti prikaz Nessus pripreme za skeniranje koje prikazuje

locirane uređaje. Skeniranje će pokazati trenutno stanje po završetku skeniranja za svaki od uređaja, broj osetljivih mesta visokih, srednjih i niskih kategorija, kao i broj otvorenih portova na svakom uređaju.



Slika 21: Prikaz primene jednog od mnogih dodatak Nessus programskog paketa



Slika 22: Prikaz konfiguracije koja omogućava podešavanje Nessusa za skeniranje mrežnog sistema

Kako se pretraga svodi na pojedinačne uređaje, sto je predstavljeno na ilustraciji (Slika 23), mogu se videti informacije o specifičnim portovima koji su pronađeni, uslugama koje koriste ovi portovi, broj osetljivih mesta u vezi sa pojedinačnim uslugama, a to je sve grupisano u visoke, srednje i niske kategorije rizika. Na ovom nivou, može se početi dobijati jasnija slika kako određeni računar može biti primamljiv za napadača, nakon čega se može početi sa formulacijom specifičnije strategije za napad ili odbranu, zavisno od date uloge.



Port	Protocol	SVC Name	Total	High	Medium	Low	Open Port
0	udp	general	1	0	0	1	0
0	tcp	general	7	0	0	7	0
0	icmp	general	1	0	0	1	0
67	udp	bootps?	1	0	0	1	0
69	udp	tftp	1	0	0	1	0
80	tcp	www	4	0	0	3	1
137	udp	netbios-ns	1	0	0	1	0
1780	tcp	dpkeyserv?	2	0	0	1	1

Slika 23: Rezultati nakon skeniranja i specifikacija određenog uređaja

Odavde se može izvesti listing svih osetljivih mesta određene usluge za uređaj u pitanju. Ovo može dati još jedan nivo detalja gde postoje nedostaci u smislu sigurnosti, ali oni zaista interesantni detalji o osetljivosti kriju se u još jednom nivou ispod, kao sto je predstavljeno na ilustraciji (Slika 24). Ovaj detaljni listing može dati poseban opis kako se ta osetljivost može iskoristiti za napad na uređaj, ali i uputiti na druge potencijalne izvore informacija.



Slika 24: Nessus izveštaj, sa detaljima o osjetljivosti skeniranog sistema

Ukoliko napadač ili administrator želi da proširi ili automatizuje funkcionalnost Nessus-a, Jezik skriptovanja za napad korišćenjem Nessus-a (Nessus Attack Scripting Language (NASL)) može to omogućiti. Takođe omogućava i pristup Nessus-u putem komandne linije. Knjiga *Revizija Nessus mreže, drugo izdanje* (ISBN: 978-1-59749-208-9, Syngress), autora Russ Rogers-a detaljno opisuje jezik skriptovanja i korišćenje NASL usluga.

Nessus alternativa otvorenog koda, postoji Nessus verzija koja se zove Otvoreni sistem za procenu osjetljivih mesta (OpenVAS). OpenVAS je "izdanak" Nessus-a još iz vremena kada je isti bio proizvod otvorenog izvora, tako da poseduje mnoge karakteristike istog.

OpenVAS je u velikoj meri kompatibilan sa standardnim Nessus dodacima, a poseduje i mogućnost da koristi prilagođene dodatke kreirane u NASL. OpenVAS takođe nudi i sopstvene javne dodatke, koji su skoro isti ili slični dodacima dostupnim u Nessus menadžeru dodataka. Sprovedeni su testovi poređenja između Nessus-a i OpenVAS-a koji su potvrdili da Nessus jeste superiorniji proizvod ali i da je OpenVAS obavlja odličan posao i da predstavlja dostojnu alternativu Nessus-u.

3.2.2.3. Odbrana od vertikalnog izviđanja

Zaštita informacija od skenera može da bude težak zadatak. Ukoliko je skener postavljen na takav način da ima pristup mreži ili da može da “špijunira” mrežni saobraćaj, posebno ako je cilj izložen internetu, onda je velika verovatnoća da je korisnik izložen napadima skeniranja. Koncept sprečavanja curenja informacija ka skenerima može biti primenjen. Mrežni saobraćaj ne treba formulisati na način koji bi bio lako vidljiv neovlašćenim “prisluškivačima”, ne obrađivati usluge na standardnim portovima, ne slati nekriptovan saobraćaj, ili primena bilo koje od mera sličnih pomenutim.

Mnogi alati za skeniranje zavise od usluga koje postoje na opštim portovima i otvorenim pristupima informacijama kako bi se sakupili njihovi izveštaji. U mnogim slučajevima, dok se verzija skeniranja još nije odradila, alati za skeniranje će prijavljivati da se usluga obavlja na osnovu zajedničkog porta koji je otvoren. Ukoliko skener nađe otvoreni port na 21, uobičajeno će pretpostaviti da je usluga koja stoji iza toga FTP. Promenom ovih osnovnih parametara u okruženju može se jako brzo poništiti informacija koja se vraća putem alata za skeniranje, i napadač se može primorati da istroši još mnogo vremena i truda kako bi otkrio šta se dešava na datom uređaju.

3.2.3. Izviđanje po dubini, sistemi za pristup i iskorišćavanje propusta, Alati za pristup i eskalaciju

Veliki broj hakerskih i penetracionih alata za testiranje koji su dostupni, bilo otvorenog izvora ili komercijalni, fokusiraju se na dobijanje pristupa sistemima i eskalaciju na nivou privilegija jednom kada je korisnik već u mogućnosti da pristupi svom sistemu. Postoji previše alata ove namene da bi svi bili predstavljeni pojedinačno, tako da će biti predstavljeni oni koji se češće koriste i one koji su popularniji u ovoj oblasti⁴¹.

⁴¹ Product Details Deep Web Secrecy and Security - an inter-active guide to the Deep Web and beyond (Deep Web Guides); Conrad Jaeger (Jul 11, 2012)

3.2.3.1. Alati za šifru

Koliko god to loše zvučalo, kreiranje šifara je samo vrh ledenog brega kada se radi o daljim problemima sa šiframa. U današnje vreme, većini korisnika su potrebne višestruke šifre ali sami korisnici ne mogu (ili ne žele) da pamte veliki broj šifara. Ovo kao rezultat daje veliki broj šifara koje se ponavljaju, a svaka šifra je dovoljno sigurna samo na poslednjem mestu gde je korišćena, ne dalje. Ukoliko napadač nađe jednu od šifara korisnika, napadač će mudro iskoristiti istu (kao i njene varijacije) i na ostalim mestima gde korisnik koristi pomenutu šifru.

Socijalno inženjerstvo takođe predstavlja veliki problem kada je reč o šiframa. Na primer, ukoliko korisnika kontaktira neko ko tvrdi da je sistemski administrator kome treba korisnikova šifra kako bi rešio problem sa korisnikovim nalogom. Prema najnovijem istraživanju, 34% korisnika bi dalo svoju šifru ako im se ista zatraži, a taj broj raste na čitavih 70% ukoliko se priča „zašećeri“ nekim dodatnim obećanjem (Passwords revealed by sweet deal, April 20, 2004).

Softver za registraciju putem otkucavanja tipke i slični Spyware softveri takođe predstavljaju ozbiljne pretnje za sigurnost zasnovanoj na šifri. Ukoliko se zaboravi da se promeni ta standardna (default) šifra, to može da predstavlja ogroman izvor za dalje napade. Interesantno bi bili postaviti pitanje ko zapravo pati od loših šifara? Odgovor je da to zavisi – ukoliko korisnik izabere datum svog rođenja za svoj ATM PIN broj, velike su šanse da samo korisnik tu gubi. Sa druge strane, ukoliko izabere slabu šifru na poslu, cela kompanija može biti na gubitku. To objašnjava zašto banke u principu dozvoljavaju klijentima da izaberu bilo koji PIN broj koji oni žele za svoje ATM kartice ali se stoga kompanije generalno trude da „nateraju“ zaposlene da izaberu razumno jake šifre.

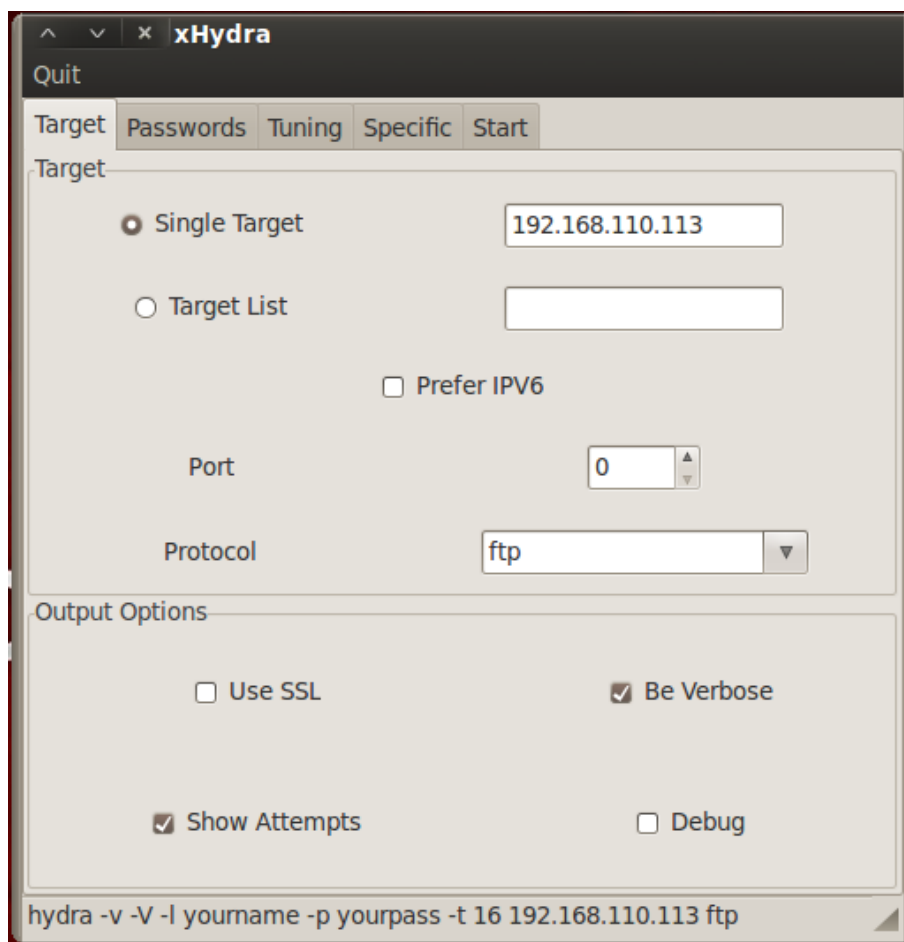
Postoji mnogo popularnih alati za kreiranje šifara, uključujući i L0phtCrack (za Windows) i Džona Trboseka (za Unix). Ovi alati se proizvode sa prekonfigurisanim rečnicima a lako je napraviti prilagođene rečnike. Isti predstavljaju dobre primere vrsta alata koji su dostupni hakerima. Imajući u vidu da zapravo nikakva veština nije potrebna kako bi se upravljalo ovim moćnim alatima, vrata ka kreiranju šifara su zapravo otvorena za sve, nezavisno od sposobnosti istih. Šifre u današnje vreme predstavljaju jedan od najozbiljnijih problema sa bezbednošću u stvarnom svetu, a mala je verovatnoća da će se to uskoro i

promeniti. Potencijalni loši momci zaista imaju prednost kada se radi o šiframa. Zapravo, socijalno inženjerstvo predstavlja najveću opasnost u svim aspektima informacione bezbednosti gde ljudi igraju bilo kakvu ulogu. Biometrija, smart-kartice i slično često može biti najbolji posrednik za premošćavanje gomile postojećih problema sa šiframa. U radu će biti prezentovano i prihvatljivo rešenje sa malim troškovima za integraciju, a bazirano na višekanalnoj autentifikaciji.

Naravno, skoro svaki hakerski alat može da se koristi i u legitimne svrhe. Na primer, alati za krekovanje šifara su od velike važnosti za sistemske administratore, jer na taj način mogu da kontrolišu jačinu šifara u svojim sistemima.

Pored toga što su loše konstruisane šifre su i previše uobičajene i opšte, tako da alati za napad na šifre mogu da budu dobra početna osnova pri pokušaju da se pristupi nekom uređaju ili da se uđe na nalog sa visim nivoom privilegija. Dva najčešća alata koja se koriste kada se obavljaju napadi ovakve vrste zovu se Hydra (Hydra) i Džon Trbosek (John the Ripper).

Hydra predstavlja alat za dobijanje šifara koje se pogađaju putem mnoštva usluga i protokola. Hydra može da “počiva” na mnoštvu operativnih sistema kao i da radi na komandnoj liniji ili GUI, kao što je i predstavljeno na ilustraciji (Slika 25).



Slika 25: Prikaz alata Hydra

Hydra može da se koristi sa pojedinačnim korisničkim imenima i šiframa, ili može da radi sa listama bilo koje od pomenutih ili čak oba. Ukoliko postoji slaba politika šifara na ciljanom sistemu, i postoji razumnu lista šifara sa koje bi sistem mogao da radi,, postojala bi velika verovatnoća da se pogodi šifra za određeni nalog. Ukoliko bi naravno postojalo dovoljno vremena za isto. Uz malo istraživanja, moglo bi se pronaći liste šifara koje sadrže tipske šifre za čitav dijapazon hardvera, ili zajedničke šifre na dosta različitih jezika.

Džon Trbosek ima malo drugačiji pristup napadima šifara. Umesto da pogađa šifre sa lista kao sto to Hydra radi, ovaj alat preuzima enkriptovanu formu šifre, što se obično naziva heš vrednost šifre i pokušava da izvuče šifru iz istog. Heš vrednosti šifre su matematičke funkcije koje, ako se primene na pravi način, praktično se smatraju nemogućim za inverziju. Isto se može zaobići ako se zna koji je has algoritam upotrebljen, tako sto se koristi poznati algoritam kako bi se heš-iralo nekoliko pokušaja o tome koja bi šifra mogla biti prava, dok se ne nađe heš koji se poklapa. U tom trenutku može se znati koja je to šifra koja je predstavljena heš-om. Džon Trbosek, poznatiji samo pod nazivom Džon, može da sprovede

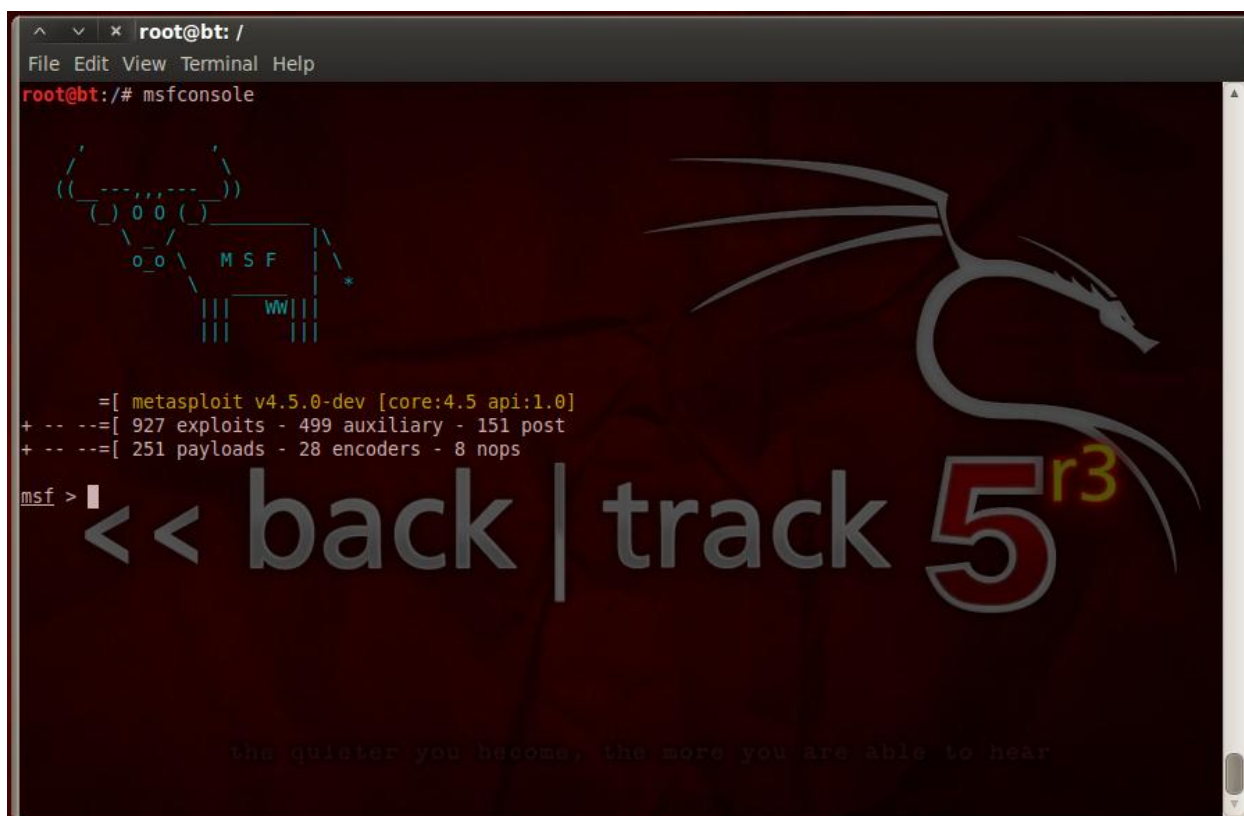
ovu vežbu sa heš vrednostima šifara mnogih operativnih sistema, a može takođe i da radi na raznim operativnim sistemima.

3.2.3.2. [Projekat Metasploit](#)

Projekat Metasploit predstavlja dobro poznatu zbirku bezbednosnih alatki otvorenog koda, koja se na tržištu pojavila 2003-e (proizvođač HD Moore). Metasploit je 2009-e pridružen alatu Rapid7 i sada “uživa” znatno povećana sredstva za sopstveni razvoj. Ovo je dovelo do toga da se Metasploit razgrana na više komercijalnih verzija sa mnogo karakteristika, pored konstantnog razvoja besplatnih alata na originalnoj verziji. U vreme kada je ovaj rad pisan, je bio dostupan u 3 verzije: besplatan Metasploit okvir i komercijalnije Metasploit verzije, Armitrage i Pro.

3.2.3.2.1. *Metasploit okvir*

Metasploit okvir je besplatna verzija Metasploit-a i pre spajanja sa Rapid7, to je bila i jedina verzija koja je bila dostupna. Prvenstvena namena Okvira je bila da se koristi kao alatka komandne linije. Iako postoji osnovna GUI verzija koja je dostupna, isti ne nudi kompletan pristup celom setu alatki koje su dostupne na komandnoj liniji, kao što i predstavljeno na ilustraciji (Slika 26).



Slika 26: Prikaz po pokretanju msfconsole Metasploit okvira

Proces korišćenja Metasploit okvira kako bi se napao sistem je sledeći, objašnjen u par koraka:

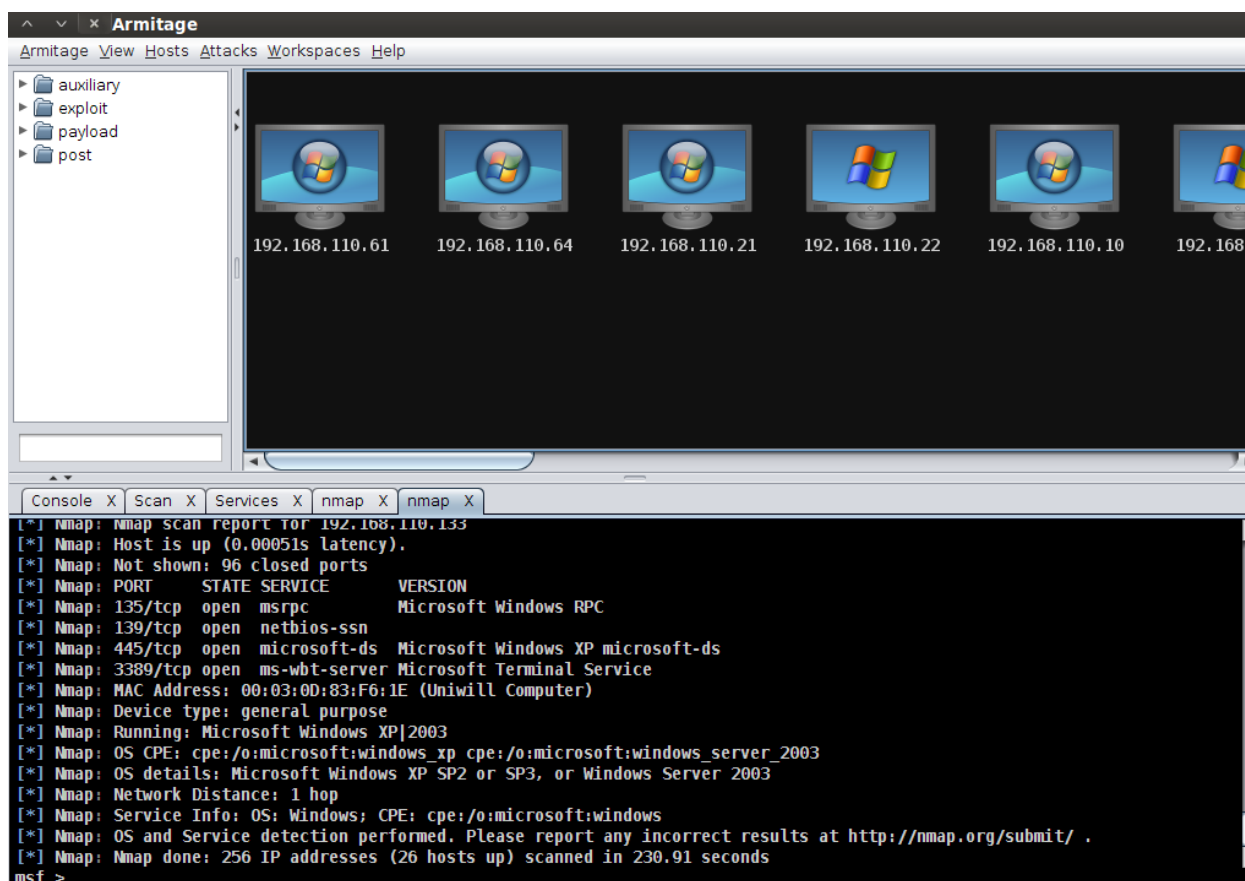
- Sakupite informacije o ciljanom sistemu koristeći alate procene za skeniranje i osetljiva mesta, kao što su prethodno predstavljeni Nmap i Nessus
- Na osnovu sakupljenih informacija, napadač odabere način napada koji se poklapa sa sistemom
- Napadač potom odabira neki koristan teret (payload) koji će pratiti napad, a to često biva udaljeni napad kroz propust operativnog sistema ili dela sistema
- Napadač primenjuje napad kao i koristan teret (payload)

U vreme pisanja ovog rada, Okvir je nudio preko 900 načina napada sa oko 250 korisnih tereta (payload) koji bi mogli biti upotrebljeni u saradnji sa istim. Okvir takođe podržava i naprednije napade kao što je proksi pivotiranje; komunikaciju sa ostalim alatima kao što je Nessus, putem rastegljivog jezika daljinskog proceduralnog poziva (Extensible Markup Language Remote Procedure Call (XML-RPC)), i proširenje putem Rubi jezika, na

kojem je razvijena trenutna verzija Metasploit-a. Okvir poseduje ogroman dijapazon mogućnosti, umnogome zbog bezbednosne zajednice, i predstavlja svestranu alatku.

3.2.3.2.2. *Metasploit Armitage, Metasploit Pro, Immunity CANVAS*

Mrežni sistemi su 2010-e mogli bi reći nespremno dočekali dolazak komercijalnih ponuda Metasploit-a koje naravno nisu bile besplatne. Metasploit Express, prvolansirana komercijalna verzija Metasploit-a(danas paket Armitage), sadrži sve funkcionalnosti kao i metasploit okvir, ali poseduje i dodatne karakteristike. Jedna od najočiglednijih karakteristika u Metasploit Express, Pro i Armitage verzijama je implementacija kompletnog GUI-a, kao što je i predstavljeno na ilustraciji (Slika 27). Ova metasploit nadogradnja omogućila je i manje iskusnim napadačima i testerima sistema da se oprobaju na ovom polju, a administratorima koji su je nespremno dočekali dodatnu glavobolju.



Slika 27: Grafičko okruženje Metasploita kroz Armitage

Pored funkcionalnosti koje pruža i Metasploit okvir, Express ima i dodatne karakteristike koje su osmišljene kako bi automatizovale i olakšale upotrebu Metasploit-a pri

većim napadima ili okruženjima testiranja penetracije. Express obuhvata automatsko otkrivanje mreža, napade na naloge kao i upotrebu različitih vrsta napada. Pored toga, za korišćenje u timskim okruženjima, Express dodaje karakteristike radnog toka, sakupljanje dokaza i reviziju kao i poboljšane alate izveštavanja. S obzirom da je isti komercijalna alatka, dodatna podešavanja podrške su takođe omogućena za korisnike istog.

Metasploit Pro je najnoviji dodatak u porodici Metasploit proizvoda. Pro poseduje sve karakteristike i funkcionalnosti Metasploit okvira i Express-a, sa dodatkom mogućnosti da odradi VPN preusmeravanje, skeniranje i eksploatisanje veb servera, kao i dodatne karakteristike koje omogućavaju bolju timsku saradnju i izveštavanje.

Raskorak u funkcionalnosti između Express-a i Pro-a je zapravo usmeren ka onima koji bi upotrebili isti alat u širim okruženjima gde više napadača ili više timova napadača blisko napadaju određene ciljeve. Jedan od nedostataka Metasploit alata oduvek je bilo pitanje šta uraditi sa informacijama i pristupom jednom kada se isti postigne, a Pro verzija direktno rešava to pitanje. Uloga alata može biti posmatrana i sa strane administratora sistema, koji bi mogli iste koristiti za penetracijsko testiranje stabilnosti svojih mrežnih sistema.

CANVAS Professional od proizvođača Immunity predstavlja alat koji omogućava pristup i eksploataciju sistema na poluautomatizovan ili automatizovan način. CANVAS sadrži dobar izbor načina napada kao i korisnih tereta (oko 400 načina napada u vreme pisanja ovog teksta), kao i veliki broj paketa za načine napada od trećih strana koji omogućavaju intenzivnije napade. Iako bi se moglo istaći da CANVAS nema toliko širok dijapazon načina napada kao neki drugi alati, isti ima tendenciju da se jako brzo razvija kako bi obuhvatio novije i interesantnije načine napada odmah nakon što su isti javno objavljeni, ili čak u nekim slučajevima i pre toga.

CANVAS je razvijen u Python-u, i obuhvata i GUI i interfejs komandne linije. Pored funkcionalnosti koje se i očekuju od ovakve vrste alata, kao što je pivotiranje (Immunity to naziva odskakivanje), skeniranje mreža, alatke za napad sa strane klijenta i ostalo, CANVAS uključuje i neke druge, jedinstvenije karakteristike. Jedna od tih karakteristika koje bi se mogle nazvati korisnim su i geo lociranje i karakteristika mapiranja, što omogućava ciljanim sistemima da budu prikazani na svetskoj mapi u okviru interfejsa. Pored toga, CANVAS poseduje nekoliko oblasti u kojima GUI može da se iskoristi kako bi prikazao grafičke alate, kao što je VisualSploit za grafičke napade objekata ili GUI alati koji mogu da se iskoriste

kako bi istražili udaljeni sistemske fajlove. S 'obzirom da je kreiran u Python-u, CANVAS je alatka koja se može dosta uobličiti, i koja se može doraditi od strane korisnika koji se razumeju u isto. Immunity se ponosi činjenicom da su proizveli alat koji je namenjen upotrebi iskusnih eksperata za bezbednost. Jako je sveobuhvatan ali, kao i svako dobro sredstvo, ima neke "oštrije crte" kojih bi neiskusni administrator trebalo da se pripazi kako ne bi doveo sopstveni sistem mreže do otkaza.

3.2.3.3. Odbrana od izviđanja po dubini, sistema za pristup i iskorišćavanja propusta

Odbrana od alata pristupa i eskalacije se uglavnom zasniva na dobro sastavljenoj i implementiranoj politici šifara, slojevima "krpljenja" i ojačavanju sistema. Sve su ovo uobičajene i opšte poznate tehnike zaštite i u teoriji predstavljaju neke od najosnovnijih bezbednosnih mera koje administratori i korisnici mogu primeniti kako bi osigurali svoje okruženje, ali iste nisu toliko sveprisutno implementirane svuda kako bi se u prvi mah moglo pomisliti.

Zaštita od pogađanja šifre i alata kreiranja se uglavnom zasniva na tome da postoje dobro uspostavljene šifre. Uobičajeni standard za jake šifre je sledeći: minimalna dužina od 8 karaktera, od toga makar jedan karakter treba da bude veliko slovo, makar jedan malo slovo, da ima makar jedan broj i makar jedan simbol. Iako ovo nekima može delovati kao preterivanje, primećena je znatna razlika u korišćenju takvih šifara u odnosu na jednostavnije šifre. Šifra sa 8 karaktera koja ima samo kombinaciju malih i velikih slova poseduje 200 milijardi mogućih kombinacija. Ukoliko bi napadač imao solidnu radnu stanicu (100,000,000 pogađanja u sekundi), napadač bi mogao da prođe kroz sve moguće kombinacije za nekih 30tak min. Ukoliko se koristi jača šema šifre, kao što je gore i navedeno (veliko slovo, malo slovo, brojevi, simboli), ta šifra od 8 karaktera bi imala 7.2 kvadriliona kombinacija i bilo bi potrebno nešto preko dve godine kako bi se ista šifra "kompromitovala". Ukoliko se povećava dužina same šifre i dodaju i dodatni setovi karaktera ovaj trend se samo nastavlja, i u jednom trenutku se pogađanje šifre i kreiranje može totalno onemogućiti, pa čak i za jako moćne ili prodavane kreiranje alate.

Još jedan korak koji korisnici i administratori sistema trebaju preduzeti, posebno u slučajevima da žele da se odbranite od alata kao što su Metasploit ili CANVAS, je da se

postaraju da vam sistem ima mogućnost brze nadogradnje “krpljenja”. Mnogi ovakvi alati mogu da prodru u sistem za samo nekoliko sekundi imajući u vidu ta nepokrivena osetljiva mesta ali to je situacija koja se da lako izbeći. Administratori sistema često imaju podeljena mišljenja oko toga da je instaliranje aplikacije i “krpljenje” operativnog sistema odmah nakon izbacivanja sigurnosne zakrpe nepromišljeno i da će time samo izazvati više problema nego ih rešiti. Implementacija zakrpa se ne mora raditi automatski na svim računarima sistema, trebalo bi odvojiti neko vreme kako bi testirali te “zakrpe” pre nego se implementiraju, sa jako malo izuzetaka u tome.

Administratori sistema bi trebali da ojačaju sisteme što više je moguće ali da im i dalje dozvole da izvršavaju svoje funkcije. Što više portova, usluga, naloga postoji a koji ostanu dostupnim u sistemu, veća će biti površina koju sami napadači imaju za pokušaje ugrožavanja istih. Pojedinačni sistemi u mnogim slučajevima imaju samo nekoliko aktivnosti koje zahtevaju otvoreni spoljni pristup, bilo nadolazeći ili odlazeći, a zatvaranje takvih potencijalnih metoda znatno ograničava set alata koji su prepušteni napadaču za iskorišćenje.

3.3. Zaštita servisa računarskih mreža

Skoro sve kompanije, većina vladinih agencija kao i mnogi pojedinci sada imaju svoje veb sajtove. Broj pojedinaca kao i kompanija koji imaju pristup internetu se neverovatno povećava a svi oni imaju grafičke veb pretraživače. Kao rezultat pomenutog, kompanije se raduju uspostavljanju mrežnih olakšica za elektronsko poslovanje. Ali činjenica je da su i internet i mreža jako podložni kompromitovanju raznih vrsta. Kako se kompanije suočavaju sa ovom realnošću, zahtevi za sve sigurnijim mrežnim uslugama rastu⁴².

Svetska mreža zapravo predstavlja klijentsko-serversku aplikaciju koja funkcioniše putem interneta kao i TCP/IP mreža. S obzirom na to, bezbednosni alati i pristupi o kojima je bilo reči do sada jesu relevantni za pitanje mrežne sigurnosti. Mreža donosi nove izazove koji se u principu ne odobravaju mnogo u smislu kompjutera i mrežne sigurnosti.

⁴² Network Security: Private Communications in a Public World (2nd Edition); Mike Speciner, Radia Perlman and Charlie Kaufman (Apr 22, 2002)

- Internet funkcioniše obostrano. Za razliku od tradicionalnih sredina za objavljivanje, čak i elektronski javni sistemi kao što su teletext, glasovna pošta ili faksiranje putem veba čini istu sredinu podložnom napadima mrežnih usluga putem interneta.
- Veb/mreža sve više služi kao sasvim otvorena “radnja” za korporativne informacije kao i informacije o proizvodima, pa i kao platforma za poslovne transakcije. Određene reputacije mogu biti narušene, novac izgubljen ukoliko se mrežne usluge ugroze i kompromituju.
- Veb pretraživači su jako laki za upotrebu, veb serveri se poprilično lako mogu konfigurisati i njima se može lako upravljati, a veb sadržaj se lako može razviti, sam softver ispod svega toga može biti jako kompleksan. Takav kompleksan softver može kriti mnogo potencijalnih bezbednosnih propusta. Kratka istorija veba je ispunjena primerima novih i unapređenih sistema koju su instalirani na pravi način a koji su opet podložni bezbednosnim napadima.
- Mrežni server može da se iskoristi kao odskočna daska za upad u čitav korporacijski ili agencijski kompjuterski kompleks. Jednom kada upadne na mrežni server, haker ima mogućnost da pristupi podacima kao i sistemima koji nisu deo same mreže ali su povezani sa serverom na lokalnom sajtu.
- Obični i neobučeni korisnici (u sigurnosnom smislu) često koriste mrežne usluge. Takvi korisnici nisu čak možda ni svesni svih bezbedonoskih rizika koji postoje pa stoga nemaju alate niti dovoljno znanja kako bi preduzeli određene protivmere.

3.3.1. Mrežne bezbednosne pretnje

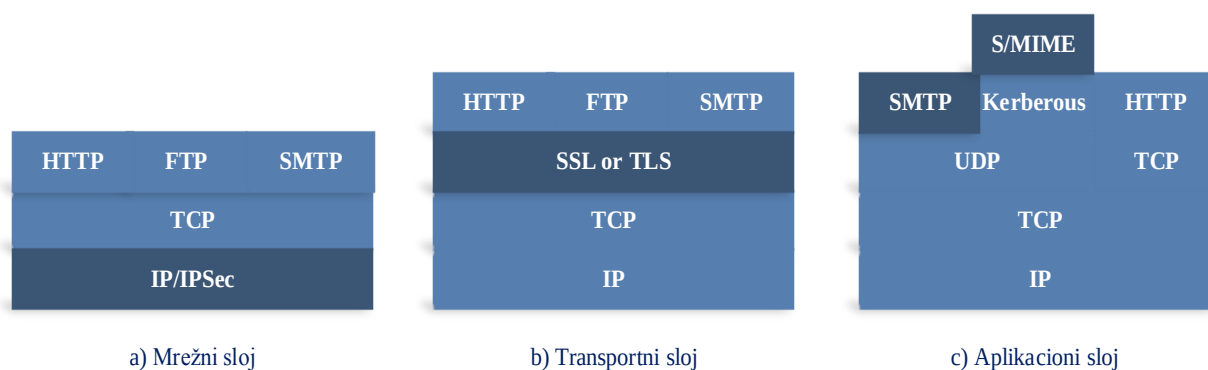
Tabela 1 prikazuje rezime vrsta bezbednosnih pretnji sa kojima se korisnik može suočiti kada je na globalnoj mreži. Jedan od načina da se grupišu ove pretnje jeste podela na pasivne i aktivne napade. Pasivni napadi obuhvataju „prisluškivanje“ mrežnog saobraćaja između pretraživača i servera, i dobijanje pristupa informacijama na veb sajtovima koji bi trebalo da imaju ograničen pristup. Aktivni napadi obuhvataju predstavljanje umesto drugog korisnika, menjanje poruka u tranzitu između klijenta i servera, kao i menjanje informacija na veb sajtu. Još jedan način da se klasifikuju mrežne bezbednosne pretnje je u smislu lokacije same pretnje: veb server, veb pretraživač, i mrežni saobraćaj između pretraživača i servera. Problemi u smislu sigurnosti mrežnog saobraćaja potpadaju u kategoriju mrežne sigurnosti, pa se stoga o istim diskutuje u ovom delu.

	Pretnje	Posledice	Protivmere
Integritet	Promena korisničkih podataka Pretraživač sa sindromom trojanskog konja Modifikacija memorije Modifikacija poruke	Gubitak informacije Kompromitovan računar Ranjivost na sve druge pretnje	Kriptografska provera ček vrednosti
Poverljivost	Prisluškivanje na mreži Krađa informacija sa servera Krađa informacija od klijenta Informacije o konfiguraciji mreže Informacije o razmeni klijenta i servera	Gubitak informacija Gubitak privatnosti	Šifrovanje, Proxy serveri
Odbijanje usluge	Prekidanje korisničkih zahteva Preplavlivanje lažnim zahtevima Popunjavanje memorije Napadi na DNS sisteme	Iritirajuće Sprečavanje obavljanja zahteva, posla Ometajuće	Teško se sprečavaju
Autentifikacija	Predstavljanje kako što bi se predstavio legalni korisnik Falsifikovanje podataka	Pogrešno tumačenje korisnika Uverenje da su lažne informacije tačne	Kriptografske tehnike

Tabela 1: Poređenje pretnji na mreži

3.3.2. Pristupi bezbednosti mrežnog saobraćaja

Postoji veliki broj pristupa za obezbeđenje mrežne sigurnosti. Ti isti razni pristupi su slični u smislu usluga koje pružaju i donekle u smislu mehanizama koje koriste, ali se razlikuju u smislu samog obima primenljivosti i relativnoj lokaciji u okviru TCP/IP protokola.



Slika 28: Relativna lokacija sigurnosti objekata u TCP/IP protokolima

Slika 28 predstavlja ovu razliku. Jedan od načina da se obezbedi mrežna sigurnost je da se koristi IP bezbednosni protokol (IPsec) (Slika 28a). Prednost Ipsec korišćenja je da je ista transparentna za krajnje korisnike i aplikacije, i pruža opšta rešenja. Štaviše, IPsec obuhvata i mogućnost filtriranja pa se tako samo odabrani saobraćaj ne izlaže IPsec procesiranju.

Još jedno relativno opšte rešenje je da se primeni sigurnost odmah iznad linije kontrolnog protokola prenošenja (TCP – Transmission Control Protokol) (Slika 28b). Najistaknutiji primer ovakvog pristupa je sloj sigurnosnog adaptera (Secure Sockets Layer (SSL)) kao i unapređeni internet standard poznatiji kao bezbednosni sloj prenosa (Transport Layer Security (TLS)). Na ovom nivou, postoje dva izbora implementacije. U velikoj većini slučajeva, SSL (ili TLS) mogu da se obezbede kao deo potpornog protokolnog programa pa bi stoga isti bili transparentni aplikacijama. Kao alternativa ovome, SSL može da se ugradi u posebne pakete. Na primer, Netscape, Microsoft Explorer, Mozilla Firefox, Google Chrome pretraživači već imaju u sebi ugrađen SSL, a većina veb servera ima već implementirani protokol.

Bezbednosne usluge koje su prilagođene određenim aplikacijama su već ugrađene u te određene aplikacije. Slika 28c predstavlja primer takve arhitekture. Prednost ovakvog pristupa je da usluga može da bude napravljena u skladu sa posebnim potrebama svake od datih aplikacija.

3.3.3. Sloj sigurnosnog adaptera i bezbednosni sloj prenosa

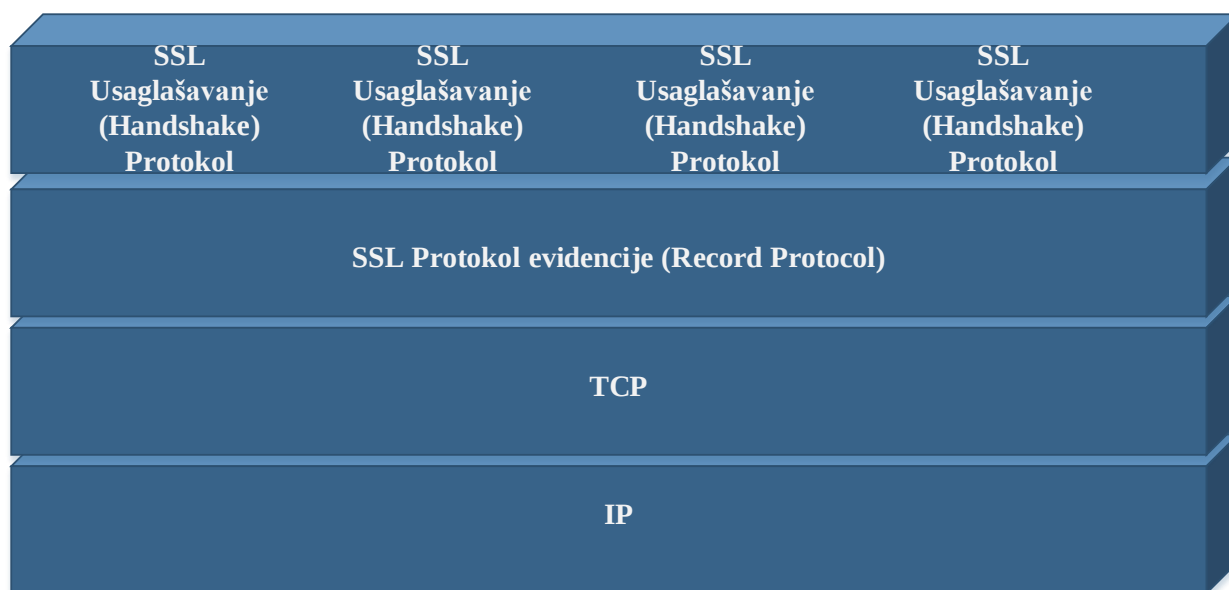
Netscape je zapravo proizveo SSL. Verzija 3 ovog protokola je kreirana nakon javnog prikaza i nakon podataka dobijenih iz industrije a ista je objavljena kao nacrt internet dokumenta. Kao posledica istog, kada je postignut konsenzus da se protokol za internet standardizaciju preda, TLS radna grupa je bila formirana u okviru IETF-a kako bi se razvio jedan opšti standard. Ova prvoobjavljena verzija TLS-a može da se prihvati kao inicijalna SSLv3.1 a jako je bliska i uzajamno kompatibilna sa SSLv3⁴³.

Ovaj deo je posvećen diskusiji o SSLv3. U narednom delu, glavne razlike između SSLv3 i TLS će biti predstavljene.

3.3.3.1. Arhitektura SSL-a

SSL je osmišljen da iskoristi TCP kako bi se uspostavila pouzdana bezbednosna usluga od početka do kraja. SSL ne predstavlja jedinstveni protokol već 2 sloja protokola, kao što je i predstavljeno na ilustraciji (Slika 29).

⁴³ Cross Layer Design Networks: A Security Aspect; Anshu Chaturvedi (Jul 25, 2012)



Slika 29 : Slojevi SSLa

SSL protokolna evidencija pruža osnovne bezbednosne usluge različitim protokolima koji se sastoje iz više slojeva. A posebno hipertext protokol prenosa (HTTP=Hypertext Transfer Protokol (HTTP)), koji pruža uslugu prenosa za interakciju veb klijenta/servera, može funkcionisati povrh SSL. Tri protokola koja se sastoje iz više slojeva definišu se kao deo SSL: protokol sinhronizacije (usaglašavanje) (Handshake Protokol), specijalni protokol za promenu šifre (Change Cipher Spec Protokol), kao i protokol upozorenja (Alert Protokol). Ovi SSL specifični protokoli se koriste za upravljanje SSL razmenom a biće opisani kasnije.

Dva jako važna SSL koncepta su SSL sesija i SSL veza, koji su definisani u navedenim specifikacijama.

- Veza: veza predstavlja prenos (u definiciji slojevitog modela međuveze transporta otvorenog sistema (OSI)) koji pruža odgovarajući deo usluge. Za SSL, takve veze predstavljaju odnose klijent ka klijentu lokalne mreže. Veze su privremene a svaka veza se povezuje sa jednom od sesija.
- Sesija: SSL sesija predstavlja spoj između klijenta i servera. Sesije su stvorene Protokolom sinhronizacije. Sesije definišu set kriptografskih sigurnosnih parametara koje mogu da se dele među višestrukim vezama. Sesije se koriste kako bi se izbegla skupa pregovaranja oko novih sigurnosnih parametara za svaku vezu.

Između svakog para obe strane (aplikacije kao što su HTTP između klijenta i servera), mogu da postoje višestruke sigurnosne veze. U teoriji, takođe mogu da postoje višestruke simultane sesije između strana ali ova karakteristika se ne koristi u praksi.

Postoji nekoliko stanja koja se povezuju sa svakom od sesija. Jednom kada se sesija uspostavi, postoji trenutno operativno stanje i za čitaj i za piši (tačnije, primi i šalji). Pored toga, u toku Protokola sinhronizacije, stvaraju se čitaj i piši stanja na čekanju. Nakon uspešnog završetka Protokola sinhronizacije, ta stanja na čekanju postaju trenutna stanja.

Stanje sesije se definiše sledećim parametrima:

- Određivač sesija/Session identifier: proizvoljna bajt sekvenca odabrana od strane servera kako bi se identifikovalo aktivno ili obnovljivo stanje sesije.
- Sertifikat veze/Peer sertifikat: X509.v3 sertifikat klijenta. Ovaj element stanja može biti prazan/nepostojeći.
- Kompresivna metoda/Compression method: algoritam koji se koristi kako bi se kompresovali podaci pre enkripcije.
- Specifikacija šifre/Cipher spec: određuje količinu podataka algoritma enkripcije (kao što su nepostojeći, AES, itd.) i heš algoritma (kao što su MD5 ili SHA-1) koji se koriste za MAC računicu. Takođe definiše i kriptografska obeležja kao što je veličina heša.
- Glavna tajna/Master secret: 48-bajtovska tajna koja se deli između klijenta i servera.
- Obnovljivo je/Is resumable: zastavica koja ističe da li sesija može da se iskoristi da podstakne nove veze.

Stanje veze se definiše sledećim parametrima:

- Direktno server-klijent/Server and client random: bajt sekvence koje se biraju od strane servera i klijenta za svaku vezu.
- Serverski napisana MAC tajna/Server write MAC secret: tajni ključ koji se koristi u MAC operacijama za podatke koje šalje server.
- Klijentski napisana MAC tajna /Client write MAC secret: tajni ključ koji se koristi u MAC operacijama za podatke koje šalje klijent.
- Serverski napisan ključ/Server write key: tajni enkripcijski ključ za podatke enkriptovane od strane servera a koje dešifruje klijent.

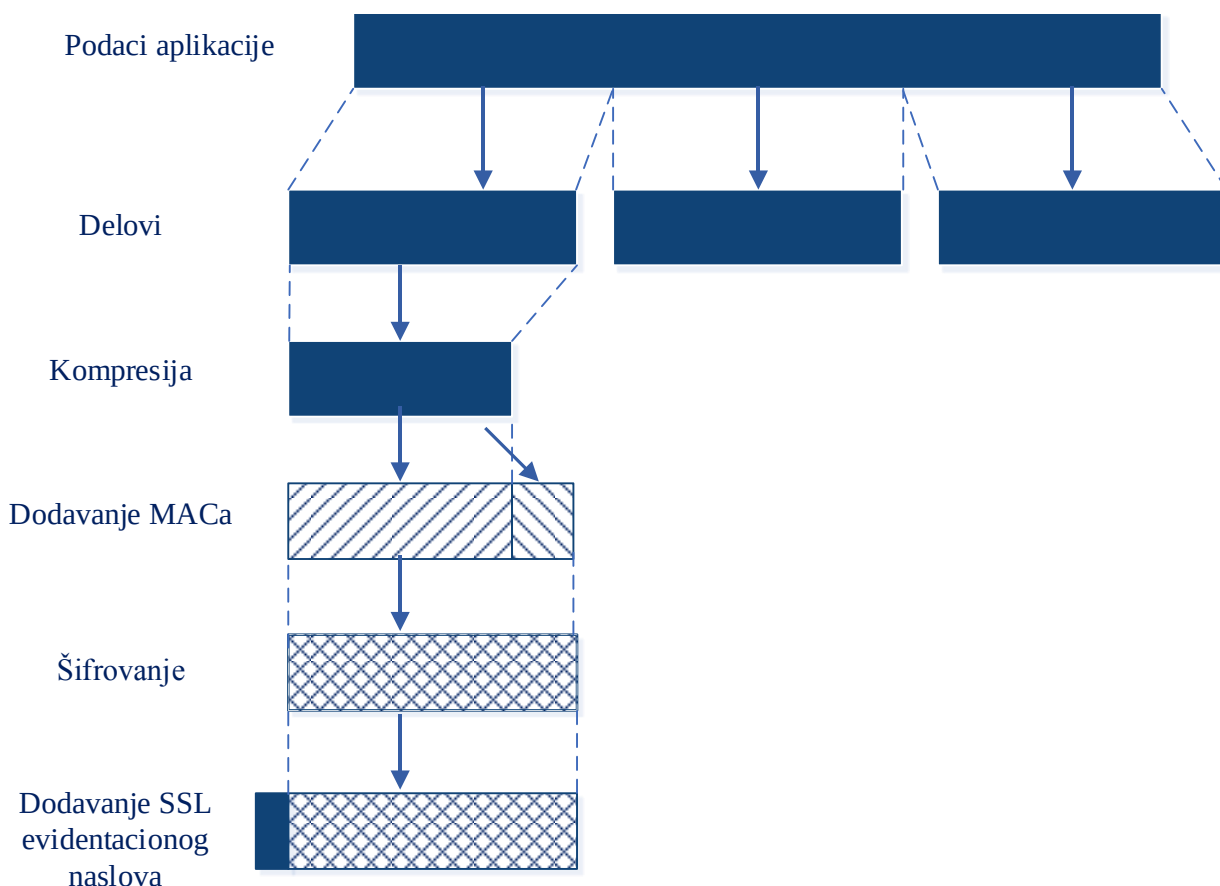
- Klijentski napisan ključ/Client write key: simetričan enkripcijski ključ za podatke enkriptovane od strane klijenta a koje dešifruje server.
- Inicirajući vektori/Initialization vectors: kada se blok-šifra u CBC stanju upotrebi, inicirajući vektor (IV) se čuva za svaki ključ. Ovo polje je prvenstveno inicirano SSL Protokolom sinhronizacije. Nakon toga, krajnji blok šifrovani tekst iz svake evidencije se čuva za upotrebu kao IV sa sledećom evidencijom.
- Uzastopni brojevi/Sequence numbers: svaka strana čuva posebne uzastopne brojeve za slanje i primanje poruka svake od veza. Kada jedna od strana pošalje ili primi poruku za promenu specifikacije šifre, odgovarajući uzastopni broj dostiže nulu. Uzastopni brojevi ne smeju da premaše $2^{64} - 1$.

3.3.3.2. [SSL protokol evidencije](#)

SSL protokol evidencije (SSL Record Protocol) pruža dve usluge za SSL veze⁴⁴:

- Poverljivost: Protokol sinhronizacije definiše zajednički tajni ključ koji se koristi za klasičnu enkripciju SSL paketa.
- Integritet poruke: Protokol sinhronizacije takođe definiše i zajednički tajni ključ koji se koristi kako bi se formirao kod autorizacije poruke (MAC).

⁴⁴ SSL and Tls: Theory and Practice (Artech House Information Security and Privacy); Rolf Oppliger (Aug 30, 2009)



Slika 30: Operacija dodavanja SSL zapisa

Ilustracija (Slika 30) objašnjava celokupnu operaciju SSL protokola evidencije. Protokol evidencije preuzima aplikacionu poruku koja treba da se prenese, fragmentiše podatke u blokove kojima se može upravljati, opciono kompresuje podatke, primenjuje MAC, enkriptuje, dodaje zaglavlje i šalje dobijenu jedinicu u formi TCP segmenta. Primljeni podaci su dešifrovani, verifikovani, dekompresovani i ponovo sakupljeni pre nego što se isporuče ka višestrukim korisnicima.

Prvi korak je fragmentacija. Svaki gornji-sloj poruke se fragmentiše u blokove od 2^{14} bajta (16384 bajta) ili manje. Kao sledeće, opciono se primenjuje kompresovanje. Kompresovanje mora da bude bez gubitaka i ne sme da premaši dužinu sadržaja više od 1024 bajta. U SSLv3 (kao i trenutnoj verziji TLS-a), nijedan algoritam kompresije nije određen tako da je opšti kompresioni algoritam prazan/nepostojeći.

Sledeći korak je da se pristupi izračunavanju koda autorizacije poruke putem kompresovanih podataka. U ovu svrhu se koristi zajednički tajni ključ. Računica se definiše na sledeći način:

$$\begin{aligned} &\text{hash}(\text{MAC_write_secret} \parallel \text{pad_2} \parallel \\ &\quad \text{hash}(\text{MAC_write_secret} \parallel \text{pad_1} \parallel \text{seq_num} \parallel \\ &\quad \text{SSLCompressed.type} \parallel \text{SSLCompressed.length} \parallel \\ &\quad \text{SSLCompressed.fragment})) \end{aligned} \quad (5)$$

Gde je:

$\parallel$	Spajanje
MAC_write_secret	zajednički tajni ključ
hash	kriptografski heš algoritam; MD5 ili SHA - 1
pad_1	bajt 0k36 (0011 0110) ponavlja 48 puta (384 bita) za MD5 i 40 puta (320 bita) za SHA - 1
pad_2	bajt 0k5C (0101 1100) ponavlja 48 puta za MD5 i 40 puta za SHA - 1
seq_num	redni broj za ovu poruku
SSLCompressed.type	protokol višeg nivoa zadužen za obradu dela
SSLCompressed.length	dužina komprimovanog dela
SSLCompressed.fragment	kompresovani deo (ako se kompresija ne koristi , deo je u otvorenom tekstu)

Treba istaći da je ovo jako slično HMAC algoritmu. Razlika je što su 2 potpore koncentrovane u SSLv3 a XOR-ovane u HMAC-u. SSLv3 MAC algoritam se zasniva na originalnom internet nacrtu za HMAC, koji koristi ulančavanje. Finalna verzija HMAC-a (definisana u RFC 2104) koristi XOR.

Kompresovana poruka plus MAC su enkriptovani uz pomoć simetrične enkripcije. Enkripcija ne bi smela da premaši dužinu sadržaja od 1024 bajta, tako da ukupna dužina ne sme da premaši $2^{14} + 2048$. Sledeći enkripcijski algoritmi se dozvoljavaju Tabela 2:

Blokovski šifrat		Šifrat niza	
Algoritam	Veličina ključa	Algoritam	Veličina ključa
AES	125, 256	RC4-40	40
IDEA	128	RC4-128	128
RC2-40	40		
DES-40	40		
DES	56		
3DES	168		
Fortezza	80		

Tabela 2: Enkripcijski algoritmi dozvoljeni pri korišćenju SSLa

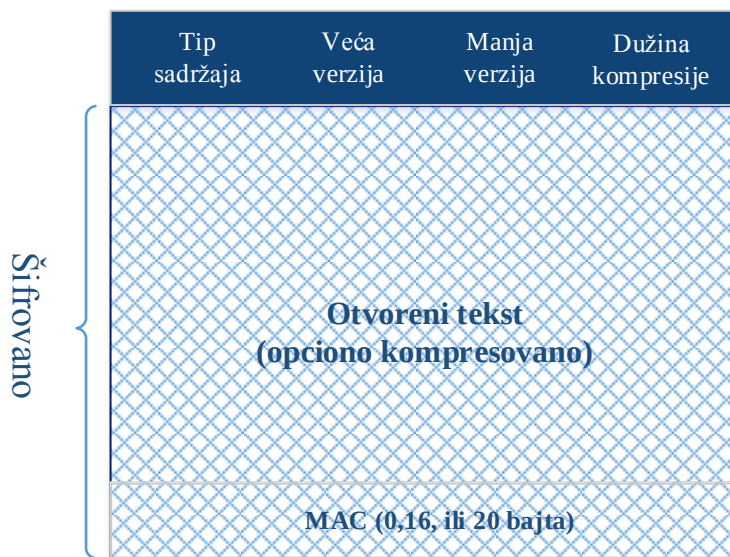
Fortezza može da se upotrebi u šemi enkripcije pametne kartice. Za rednu enkripciju, kompresovana poruka plus MAC se enkriptuju. Potrebno je istaći da se MAC računa pre nego što dođe do enkripcije a onda se MAC enkriptuje zajedno sa nešifrovanim tekstom ili kompresovanim nešifrovanim tekstom.

Za blok enkripciju, umetanje može da se doda nakon MAC-a, pre same enkripcije. Umetanje je u formi broja bajta umetanja praćeno jednobajtovskom indikacijom dužine umetanja. Celokupna količina umetanja predstavlja najmanju količinu, toliku da je ukupna količina podataka koja treba da bude enkriptovana (nešifrovani tekst plus MAC plus umetanje) višestruka dužina bloka šifre. Primer je nešifrovani tekst (ili kompresovani tekst, ukoliko se kompresovanje koristi) od 58 bajta, sa MAC-om od 20 bajta (koji koriste SHA-1), koji je enkriptovan uz pomoć bloka dužine 8 bajta (npr, DES). Sa umetanjem dužine bajta, to daje total od 79 bajta. Kako bi se total načinio celobrojnim množiocem od 8, jedan bajt umetanja se dodaje.

Krajnji korak procesiranja SSL Protokola evidencije jeste priprema zaglavlja koje bi se sastojalo od sledećih polja:

- vrsta sadržaja (8 bita): višestruki protokol koji se koristi kako bi se procesirao priloženi fragment.

- jača verzija (8 bita): označava jaču verziju SSL u upotrebi. Za SSLv3, vrednost je 3.
- slabija verzija (8 bita): označava slabiju verziju u upotrebi. Za SSLv3, vrednost je 0.
- kompresovana dužina (16 bita): dužina bajta fragmenta nešifrovanog teksta (ili kompresovanog fragmenta, ukoliko se kompresovanje koristi). Maksimalna vrednost je $2^{14} + 2048$.

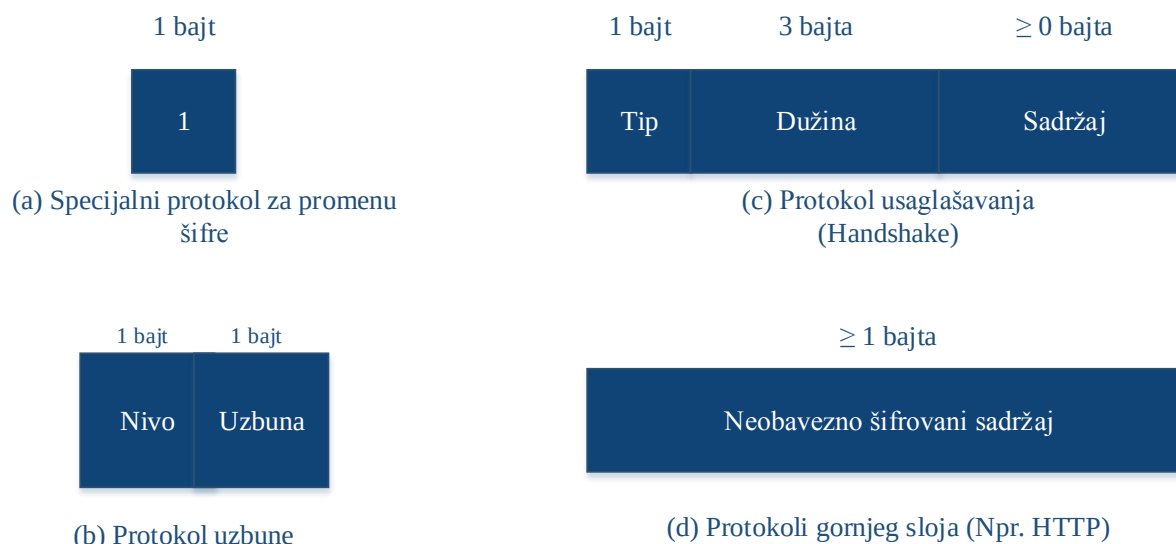


Slika 31: Format SSL zapisa

Vrste sadržaja koje su definisane su *change_cipher_spec*, *alert*, *handshake*, i *application_data*. Prve tri predstavljaju SSL-specifične protokole, i iste su sledeće o kojima će biti reči. Treba se istaći da se nikakva razlika ne pravi između raznih aplikacija (npr, HTTP) koje mogu da koriste SSL; sadržaj tih podataka stvoren takvom aplikacijom je nedokučiv za SSL. Slika 31 predstavlja SSL format evidencije.

3.3.3.3. Specijalni protokol za promenu šifrata

Specijalni protokol za promenu šifrata (Change Cipher Spec Protocol) predstavlja jedan od tri SSL-specifična protokola koji koristi SSL Protokol evidencije, i najjednostavniji je oblik protokola. Isti protokol se sastoji iz jedinstvene poruke (Slika 32a), koja se sastoji od pojedinačnog bajta sa vrednošću 1. Jedina svrha ove poruke je da izazove da se stanje na čekanju iskopira u trenutno stanje, što bi ažuriralo program šifrata da se iskoristi na ovoj vezi.



Slika 32: Paket SSL evidentacionog zapisa

3.3.3.4. Protokol upozorenja

Protokol upozorenja (Alert Protocol) se koristi da se prenesu određene SSL uzbune na srodan entitet. A što se tiče ostalih aplikacija koje koriste SSL, poruke upozorenja se kompresuju i enkriptuju, kao što je i određeno trenutnim stanjem.

Svaka poruka u ovom protokolu se sastoji iz 2 bajta (Slika 32b). Prvi bajt ima upozoravajuću vrednost (1) ili fatalnu (2) kako bi se prenela ozbiljnost poruke. Ukoliko se nivo pokaže fatalnim, SSL istog trenutka prekida vezu. Ostale veze na samoj sesiji se mogu nastaviti, ali se nove veze na istoj sesiji ne mogu uspostaviti. Drugi bajt sadrži kod koji ističe određeno upozorenje. Kao prvo, biće predstavljena ona upozorenja koja su uvek fatalna (definicije iz SSL specifikacija):

- *unexpected_poruka*: neprikladna poruka koja je primljena.
- *bad_record_mac*: neispravan MAC koji je primljen.
- *decompression_failure*: funkcija za dekompresiju koja je primila neispravne ulazne podatke (npr, nije u mogućnosti da dekompresuje ili da dekompresuje više od maksimalno dozvoljene dužine).
- *handshake_failure*: pošiljalac nije bio u mogućnosti da ispregovara prihvatljiv set sigurnosnih parametara, s obzirom na dostupne mogućnosti.

- *illegal_parameter*: polje u poruci sinhronizacije je bilo van dometa ili nekonzistentno sa ostalim poljima.

Preostala upozorenja su sledeća:

- *close_notify*: obaveštava primaoca da pošiljalac neće slati više poruka sa te iste veze. Od svake strane se zahteva da pošalje *close_notify* upozorenje pre zatvaranja pismenog dela veze.
- *no_certificate*: može biti poslat kao odgovor na zahtev za sertifikatom ukoliko nijedan odgovarajući sertifikat nije dostupan.
- *bad_sertifikat*: primljeni sertifikat je nepravilan (npr, sadrži potpis koji nije verifikovan).
- *unsupported_sertifikat*: vrsta primljenog sertifikata koja nije podržana.
- *sertifikat_revoked*: sertifikat koji je opozvan od strane svog potpisnika.
- *sertifikat_expired*: sertifikat koji je istekao.
- *sertifikat_unknown*: neki drugi, neodređeni problem se pojavio u procesuiranju sertifikata, čineći ga neprihvatljivim

3.3.4. Protokol sinhronizacije

Najkompleksniji deo SSL-a je Protokol sinhronizacije. Ovaj protokol dozvoljava serveru i klijentu da autorizuju jedan drugog i da dogovore enkripciju i MAC algoritam i kriptografske ključeve koji će se koristiti kako bi se zaštitili podaci poslati ka SSL evidenciji. Protokol sinhronizacije se koristi pre nego što se bilo koji aplikacioni podaci prenesu⁴⁵.

Protokol sinhronizacije se sastoji od serija poruka koje su razmenjene od strane klijenta i servera. Sve ove poruke imaju format kakva je predstavljen na ilustraciji (Slika 32c). svaka od poruka ima 3 polja:

- vrsta (1 bajt): ističe jednu od 10 poruka. Tabela nabraja definisane vrste poruka.
- dužina (3 bajta): dužina poruke izražena u bajtovima.

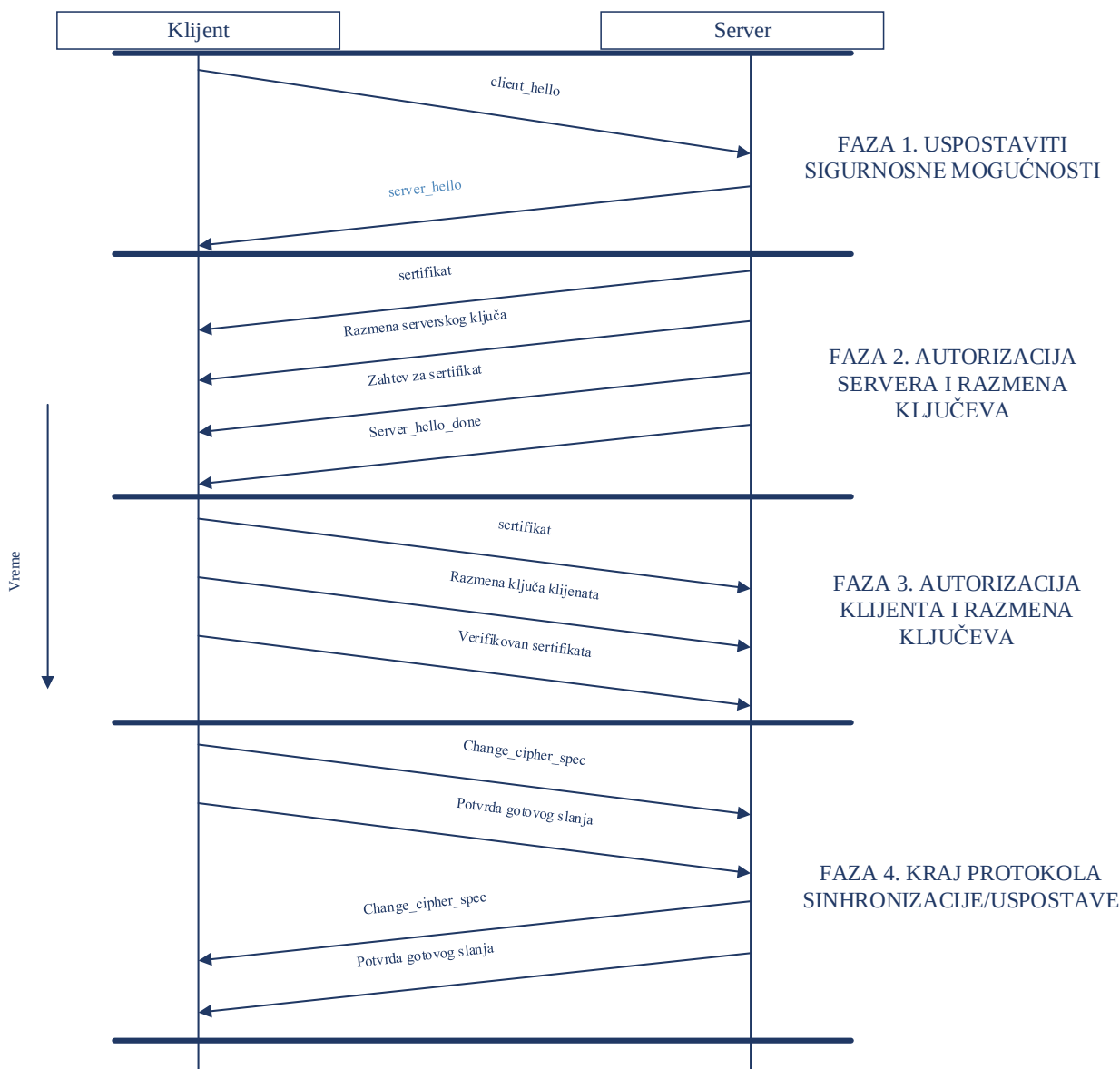
⁴⁵ PKI: Implementing & Managing E-Security; Andrew Nash, Bill Duane, Derek Brink and Celia Joseph (Mar 27, 2001)

- sadržaj (bajta): parametri koji se povezuju sa ovom porukom; isti su nabrojani u sledećoj tabeli (Tabela 3).

Tip poruke	Parametri
hello_request	Null
client_hello	Verzija, ID sesije, metod kompresije, tip šifrata, slučajna vrednost
server_hello	Verzija, ID sesije, metod kompresije, tip šifrata, slučajna vrednost
Certificate	lanac sertifikata X.509v3
server_key_exchange	Parametri, potpis
certificate_request	Tip, ovlašćena tela
server_done	Null
certificate_verify	Potpis
client_key_exchang	Parametri, potpis
finished	Heš vrednost

Tabela 3: Parametri korićeni u protokolu uspostave/sinhronizacije

Slika 33 predstavlja inicijalnu razmenu koja je neophodna kako bi se uspostavila logična veza između klijenta i servera. Razmena može da se posmatra kroz 4 faze.



Slika 33: Inicijalna razmena koja je neophodna kako bi se uspostavila veza između klijenta i servera

3.3.4.1. Faza 1. Uspostaviti sigurnosne mogućnosti

Ova faza se koristi kako bi se inicirala logična veza i uspostavile sigurnosne mogućnosti koje će biti pridodate istoj. Razmenu inicira klijent koji šalje *client_hello* poruku sa sledećim parametrima:

- verzija: najviša SSL verzija koju klijent može da razume.
- nasumično/direktno: nasumična struktura stvorena od strane klijenta koja se sastoji iz 32-bita vremenskih odrednica i 28 bajta sakupljenih sigurnim generatorom

nasumičnim brojeva. Ove vrednosti se koriste samo kao trenutne i koriste se u toku razmene ključeva kako bi se sprečili reprodukovani napadi.

- ID Sesije: određivač sesija izmenljivih dužina. Vrednost koja nije jednaka nuli ističe da klijent želi da poboljša parametre postojeće veze ili da stvori novu vezu na ovoj sesiji. Vrednost nula ističe da klijent želi da uspostavi novu vezu na novoj sesiji.
- šifrat program: ovo je lista koja sadrži kombinaciju kriptografskih algoritama koje klijent podržava, po opadajućem redu preferencije. Svaki element sa liste (svaki šifrat program) definiše i algoritam razmene ključeva i specifikaciju šifre; o istim će biti reči kasnije.
- metoda kompresovanja: ovo je lista metoda kompresovanja koje klijent podržava.

Nakon slanja *client_hello* poruke, klijent čeka na *server_hello* poruku, koja sadrži iste parametre kao i klijentova *hello* poruka. Sledeća pravila se mogu primeniti za *server_hello* poruku. Polje verzija sadrži slabiju verziju koju je predložio klijent i bolju verziju koju server podržava. Nasumično polje se stvara od strane servera i nezavisno je od klijentovog nasumičnog polja. Ukoliko klijentovo polje ID sesije nije bilo nula, ista vrednost se koristi i od strane servera; u suprotnom, server polje ID sesije sadrži vrednost za novu sesiju. Šifrat program polje sadrži jedinstveni šifrat program odabran od strane servera od onih koje je predložio klijent. Polje kompresovanja sadrži metodu kompresovanja odabranu od strane servera od onih koje je predložio klijent.

Prvi element parametra šifrata programa predstavlja metoda razmene ključa (tj, sredstvo pomoću kojeg se kriptografski ključevi za standardnu enkripciju i MAC razmenjuju). Sledeće metode razmene ključa su podržane:

- RSA: tajni ključ koji je enkriptovan RSA javnim ključem primaoca. Sertifikat javnog ključa za ključ primaoca mora da bude dostupan.
- Fiksni Diffie-Hellman: Diffie-Hellman razmena ključeva u kojoj serverski sertifikat sadrži Diffie-Hellman javne parametre koji su potpisani od strane nadležnih za sertifikat (CA). Tačnije, sertifikat javnog ključa sadrži Diffie-Hellman parametre javnog ključa. Klijent pruža svoje Diffie-Hellman parametre javnog ključa ili u sertifikatu, ukoliko se traži autorizacija klijenta, ili u poruci razmene ključa. Ova metoda kao rezultat daje fiksni tajni ključ između dve uspostave srodnika na osnovu Diffie-Hellman računice koja koristi fiksne javne ključeve.

- Privremeni Diffie-Hellman /Ephemeral Diffie-Hellman: ova tehnika se koristi da se kreiraju jednokratni tajni ključevi. U ovom slučaju, Diffie-Hellman javni ključevi se razmenjuju, potpisani korišćenjem privatnog RSA pošiljaoca ili DSS ključa. Primalac može da iskoristi odgovarajući javni ključ kako bi verifikovao potpis. Sertifikati se koriste kako bi se autorizovali javni ključevi. Ovo se čini jednom od najsigurnijih od ove tri Diffie-Hellman opcije, zato što kao rezultat daje privremeni, autorizovani ključ.
- Anonimni Diffie-Hellman /Anonymous Diffie-Hellman: osnovni Diffie-Hellman algoritam se koristi bez autorizacije. Tačnije, svaka strana šalje svoje javne Diffie-Hellman parametre drugoj strani bez autorizacije. Ovakav pristup je podložan napadima posrednika/treće strane, gde hakeri imaju Diffie-Hellman anonimne parametre sa obe strane.
- Fortezza: tehnika koja definiše Fortezza šemu.

Prateći definiciju da je metoda razmene ključa zapravo specifikacija šifrata, ista uključuje sledeća polja:

- Algoritam šifre/CipherAlgoritam: bilo koji od već pomenutih algoritama: RC4, RC2, DES, 3DES, DES40, IDEA, ili Fortezza
- MACAlgoritam: MD5 ili SHA-1
- Vrsta šifre/CipherType: niz ili blok
- Može da se izvozi/IsExportable: istinito ili lažno
- Veličina heša/HashSize: 0, 16 (za MD5), ili 20 (za SHA-1) bajta
- Glavni materijal/Key Material: sekvenca bajta koja sadrži podatke koji se koriste u sakupljanju pisanih ključeva
- Veličina IV/IV Size: veličina vrednosti inicijalizacije enkripcije ulančavanja bloka šifrata (CBC)

3.3.4.2. [Faza 2. Autorizacija servera i razmena ključeva](#)

Server započinje ovu fazu tako što šalje svoj sertifikat ukoliko isti treba da se autorizuje; poruka sadrži jedan ili niz X.509 sertifikata. Sertifikat poruka se zahteva za bilo koju dogovorenu metodu razmene ključeva sem za anonimni Diffie-Hellman. Treba istaći da ako se koristi fiksni Diffie-Hellman, ova sertifikat poruka funkcioniše kao serverska poruka razmene ključeva jer sadrži serverske javne Diffie-Hellman parametre.

Kao sledeće, *server_key_exchange* poruka može biti poslata ukoliko se to zahteva. Ista nije potrebna u 2 slučaja: (1) server je poslao sertifikat sa fiksnim Diffie-Hellman parametrima ili (2) RSA razmena ključeva treba da se upotrebi. *Server_key_exchange* poruka je neophodna u sledećim slučajevima:

- Anonimni Diffie-Hellman /Anonymous Diffie-Hellman: Sadržaj poruke se sastoji od 2 opšte Diffie-Hellman vrednosti (najosnovnijeg broja i osnovnog korena tog istog broja) plus serverskog javnog Diffie-Hellman ključa.
- Privremeni Diffie-Hellman /Ephemeral Diffie-Hellman: Sadržaj poruke obuhvata 3 Diffie-Hellman parametra predviđenih za anonimni Diffie-Hellman plus potpis tih parametara.
- RSA razmena ključeva/RSA key exchange (u kojoj server koristi RSA ali ima RSA ključ koji zahteva samo RSA potpis): u skladu sa tim, klijent ne može jednostavno poslati tajni ključ enkriptovan serverskim javnim ključem. Umesto toga, server mora stvoriti privremeni RSA javno-privatni par ključeva i iskoristiti *server_key_exchange* poruku kako bi poslao javni ključ. Sadržaj poruke obuhvata 2 parametra privremenih RSA javnih ključeva (exponent/izložilac i modulus/apsolutna vrednost plus potpis tih parametara).
- Fortezza

Neki od narednih detalja oko potpisa se garantuju. Kao i obično, potpis se formira tako što se uzme haš poruke i isti se enkriptuje privatnim ključem pošiljaoca. U ovom slučaju, haš se definiše kao:

$$\text{hash}(\text{ClientHello.random} \parallel \text{ServerHello.random} \parallel \text{ServerParams}) \quad (6)$$

Tako da haš pokriva ne samo Diffie-Hellman ili RSA parametre već i dve od sadašnjih inicijalnih hello poruka. Ovim se obezbeđuje zaštita od reprodukovanih napada kao i zloupotrebe predstavljanja. U slučaju DSS potpisa haš koristi SHA-1 algoritam. U slučaju RSA potpisa, oba MD5 i SHA-1 haševi se računaju, i dva povezana haša (36 bajta) se enkriptuju serverskim privatnim ključem.

Kao sledeće, server koji nije anonimni (server koji ne koristi anonimnog Diffie-Hellman-a) može da zatraži sertifikat od klijenta. *Certificate_request* poruka obuhvata dva parametra: *certificate_type* i *certificate_authorities*.

Vrsta sertifikata ističe algoritam javnog ključa i koristi sledeće:

- RSA, samo potpis
- DSS, samo potpis
- RSA za fiksnog Diffie-Hellman-a; u ovom slučaju potpis se koristi samo za autorizaciju, time što se šalje sertifikat koji je RSA potpisan
- DSS za fiksnog Diffie-Hellman-a; opet se koristi samo za autorizaciju
- RSA za privremenog Diffie-Hellman-a
- DSS za privremenog Diffie-Hellman-a
- Fortezza

Drugi parameter u `certificate_request` poruci predstavlja listu karakterističnih imena prihvatljivih sertifikat nadležnosti.

Finalna poruka u fazi 2, kao i ona koja se uvek zahteva, jeste `server_done` poruka, koju šalje server kako bi se istakao kraj `server hello` i sličnim porukama. Nakon slanja ove poruke, server će sačekati odgovor od klijenta. Ova poruka nema parametre.

3.3.4.3. [Faza 3. Autorizacija klijenta i razmena ključeva](#)

Nakon primanja `server_done` poruke, klijent bi trebao da verifikuje da je server pružio validan sertifikat (ukoliko se to zahteva) i da proveriti da su `server_hello` parametri prihvatljivi. Ukoliko je sve zadovoljavajuće, klijent šalje jednu ili više poruka nazad ka serveru.

Ukoliko je server tražio sertifikat, klijent započinje ovu fazu time što šalje sertifikat poruke. Ukoliko nijedan odgovarajući sertifikat nije dostupan, klijent šalje `no_certificate` upozorenje umesto toga.

Sledeća je `client_key_exchange` poruka, koja mora biti poslata u ovoj fazi. Sam sadržaj poruke zavisi od vrste razmene ključeva, kao što i sledi:

- RSA: Klijent generiše 48-bajt pre-master tajnu i enkriptuje je sa javnim ključem serverskog sertifikata ili privremenim RSA ključem iz `server_key_exchange` poruke. Njena upotreba kako bi se preračunala master tajna je objašnjena kasnije.

- Ephemeral or Anonymous Diffie-Hellman: klijentovi javni Diffie-Hellman parametri su poslati.
- Fixed Diffie-Hellman: klijentovi javni Diffie-Hellman parametri su poslati u sertifikatu poruke, tako da je sadržaj same poruke prazan/nepostojeći.
- Fortezza: klijentovi Fortezza parametri su poslati.

Na kraju, u ovoj fazi, klijent može da pošalje `certificate_verify` poruku kako bi obezbedio određenu verifikaciju svog sertifikata. Ova poruka se šalje samo u slučaju da sertifikat klijenta ima mogućnost potpisivanja (tj, svi sertifikati sem onih koji sadrže fiksne Diffie-Hellman parametre). Ova poruka potpisuje haš kod na osnovu prethodnih poruka, koje su definisane na sledeći način:

$$\begin{aligned} \text{CertificateVerify.signature.md5_hash} &= \\ &\text{MD5}(\text{master_secret} \parallel \text{pad_2} \parallel \text{MD5}(\text{handshake_messages} \parallel \\ &\quad \text{master_secret} \parallel \text{pad_1})); \\ \text{CertificateVerify.signature.sha_hash} &= \\ &\text{SHA}(\text{master_secret} \parallel \text{pad_2} \parallel \text{SHA}(\text{handshake_messages} \parallel \\ &\quad \text{master_secret} \parallel \text{pad_1})); \end{aligned} \tag{7}$$

gde `pad_1` i `pad_2` predstavljaju vrednosti definisane ranije za MAC, `handshake_messages` se odnose na sve poruke Protokola sinhronizacije poslate ili primljene sa klijentovog `client_hello` ali ne uključujući ovu poruku, a `master_secret` predstavlja preračunatu tajnu čiji sastav je objašnjen kasnije u ovom odeljku. Ukoliko je korisnikov privatni ključ DSS, onda se isti koristi da enkriptuje SHA-1 haš. Ukoliko je korisnikov privatni ključ RSA, onda se isti koristi da enkriptuje uvezivanje MD5 i SHA-1 haševa. U bilo kom slučaju, svrha je verifikovati klijentovo vlasništvo nad privatnim ključem za klijentski sertifikat. Čak i ukoliko neko pokušava da zloupotrebi klijentski sertifikat, on/ona neće biti u mogućnosti da pošalje ovu poruku.

3.3.4.4. [Faza 4. Kraj](#)

Ova faza završava uspostavljanje sigurne veze. Klijent šalje `change_cipher_spec` poruku i kopira specifikaciju šifre na čekanju u trenutnu specifikaciju šifre. Treba da se

istakne da se ova poruka ne smatra delom protokola sinhronizacije ali da se šalje uz pomoć specijalnog protokola za promenu šifre. Klijent onda odmah šalje završnu poruku pod novim algoritmima, ključevima, kao i tajnama. Završna poruka verifikuje da su razmena ključeva i procesi autorizacije uspešno završeni. Sam sadržaj završne poruke predstavlja uvezivanje dve haš vrednosti:

$$\begin{aligned} & \text{MD5}(\text{master_secret} \parallel \text{pad2} \parallel \text{MD5}(\text{handshake_messages} \parallel \\ & \quad \text{Sender} \parallel \text{master_secret} \parallel \text{pad1})) \\ & \text{SHA}(\text{master_secret} \parallel \text{pad2} \parallel \text{SHA}(\text{handshake_messages} \parallel \\ & \quad \text{Sender} \parallel \text{master_secret} \parallel \text{pad1})) \end{aligned} \tag{8}$$

gde Pošiljalac predstavlja kod koji određuje da je pošiljalac klijent i da su *handshake_messages* svi podaci iz svih poruka sinhronizacije ali ne uključujući ovu poruku.

Kao odgovor na ove dve poruke, server šalje svoju sopstvenu *change_cipher_spec* poruku, prenosi čekanje na trenutnu specifikaciju šifre i šalje svoju završnu poruku. U ovom trenutku, sinhronizacija je kompletna pa klijent i server mogu da započnu razmenu različitih slojeva aplikacionih podataka.

3.3.5. Kriptografski proračuni

Dve teme koje slede su od opšteg interesa: (1) stvaranje zajedničke master tajne putem sredstava razmene ključeva i (2) sakupljanje kriptografskih parametara iz master tajne.

STVARANJE MASTER TAJNE Zajednička master tajna predstavlja jednokratnu 48-bajt vrednost (384 bita) koja se sakuplja za ovu sesiju putem sigurne razmene ključeva. Stvaranje iste je u dve faze. Kao prvo, `pre_master_secret` se razmenjuje. Kao drugo, `master_secret` se računa sa obe strane. Za `pre_master_secret` razmenu, postoje dve mogućnosti.

- **RSA:** 48-bajt `pre_master_secret` se generiše od strane klijenta, enkriptuje putem serverskog javnog RSA ključa i šalje ka serveru. Server dešifruje šifrovani tekst koristeći svoj privatni ključ kako bi obnovio `pre_master_secret`.
- **Diffie-Hellman:** i klijent i server generišu Diffie-Hellman javni ključ. Nakon što dođe do razmene istog, svaka strana izvršava Diffie-Hellman računicu kako bi stvorila zajedničku `pre_master_secret`.

Obe strane sada računaju `master_secret` kao:

$$\begin{aligned} \text{master_secret} = & \text{MD5}(\text{pre_master_secret} \parallel \text{SHA}('A' \parallel \\ & \text{pre_master_secret} \parallel \text{ClientHello.random} \parallel \\ & \text{ServerHello.random})) \parallel \\ & \text{MD5}(\text{pre_master_secret} \parallel \text{SHA}('BB' \parallel \\ & \text{pre_master_secret} \parallel \text{ClientHello.random} \parallel \text{ServerHello.random})) \parallel \\ & \text{MD5}(\text{pre_master_secret} \parallel \text{SHA}('CCC' \parallel \\ & \text{pre_master_secret} \parallel \text{ClientHello.random} \parallel \text{ServerHello.random})) \end{aligned} \quad (9)$$

gde `ClientHello.random` i `ServerHello.random` predstavljaju trenutne vrednosti koje se razmenjuju u inicijalnim hello porukama.

3.3.5.1. Generisanje kriptografskih parametara

Specifikacije šifre zahtevaju klijentski napisanu MAC tajnu, serverski napisanu MAC tajnu, klijentski napisani ključ, serverski napisani ključ, klijentski napisanu IV, serverski napisanu IV, koje se generišu iz master tajne baš tim redosledom. Ovi parametri se generišu iz master tajne hešovanjem master tajne u sekvence sigurnih bajtova dovoljne dužine za sve neophodne parametre. Generisanje glavnih materijala iz master tajne koristi isti format za generisanje master tajne iz pre-master tajne kao i

$$\begin{aligned} \text{key_block} = & \text{MD5}(\text{master_secret} \parallel \text{SHA}(\text{'A'} \parallel \text{master_secret} \parallel \\ & \text{ServerHello.random} \parallel \text{ClientHello.random})) \parallel \\ & \text{MD5}(\text{master_secret} \parallel \text{SHA}(\text{'BB'} \parallel \text{master_secret} \parallel \\ & \text{ServerHello.random} \parallel \text{ClientHello.random})) \parallel \\ & \text{MD5}(\text{master_secret} \parallel \text{SHA}(\text{'CCC'} \parallel \text{master_secret} \parallel \\ & \text{ServerHello.random} \parallel \text{ClientHello.random})) \parallel \dots \end{aligned} \quad (10)$$

dok se ne sakupi dovoljno izlaznih informacija. Rezultat ove algoritmičke strukture jeste pseudo slučajna funkcija. Master_secret može da se posmatra kao pseudo slučajna vrednost koja pothranjuje funkciju. Klijentski i serverski slučajni brojevi mogu da se posmatraju kao „zaštitne“ vrednosti (salt values) koje komplikuju kriptografsku analizu.

3.4. Sigurnosni sloj prenosa

TLS predstavlja IETF inicijativu standardizacije koja za cilj ima stvaranje internet standardne verzije SSL. TLS se definiše kao predloženi internet standard u RFC 5246. Sam RFC 5246 je jako sličan sa SSLv3. Razlike će biti diskutovane i istaknute u ovom odeljku.

Broj verzije

TLS evidencioni format je zapravo isti kao i SSL format evidencije (Slika 31), a polja u zaglavlju imaju ista značenja. Jedina razlika je u vrednostima verzija.

MAC=Message Authentication Code

Postoje dve razlike između SSLv3 i TLS MAC šema: trenutni algoritam i obim MAC računice. TLS upotrebu HMAC algoritma definiše u RFC 2104. HMAC definiše se kao:

$$\text{HMAC}_K(M) = H[(K^+ \oplus \text{opad}) || H[(K^+ \oplus \text{ipad}) || M]] \quad (11)$$

Gde je:

H	Utisnuta heš funkcija za TLS, MD5 ili SHA-1
M	Ulazna poruka za HMAC
K^+	Tajni ključ sa dodatim nulama sa leve strane, tako da je rezultat jednak dužini bloka heš koda (za MD5 i SHA-1, blok dužine 512 bita)
ipad	00110110 (36 u heksadecimalnom) ponovljeno 64 puta (512 bita)
opad	01011100 (5C u heksadecimalnom) ponovljeno 64 puta (512 bita)

SSLv3 koristi isti algoritam, sem što je umetanje bajta uvezano sa tajnim ključem a ne sa XORovanjem sa tajnim ključem ispunjenim do dužine bloka. Nivo sigurnosti je manje-više isti u oba slučaja.

Što se tiče TLS-a, MAC računica obuhvata polja naznačena u sledećem izražavanju:

$$\begin{aligned} \text{MAC}(\text{MAC_write_secret}, \text{seq_num} || \text{TLSCompressed.type} || \\ \text{TLSCompressed.version} || \text{TLSCompressed.length} || \\ \text{TLSCompressed.fragment}) \end{aligned} \quad (12)$$

MAC računica pokriva sva polja obuhvaćena SSLv3 računicom, plus polja iz *TLSCompressed.version*, što predstavlja verziju protokola koji je u upotrebi.

TLS koristi pseudo slučajnu funkciju koja se odnosi na PRF kako bi se tajne proširile na blokove podataka u svrhu generisanja ključa ili validacije. Cilj je upotrebiti relativno male vrednosti zajedničke tajne ali generisati duže blokove podataka na način koji je bezbedan od vrsta napada koji se stvaraju na haš funkcijama i MAC-ovima. PRF je zasnovan na funkciji proširenja podataka Slika 34, kao što je ispod i predstavljeno:

$$\begin{aligned} P_hash(secret, seed) = & HMAC_hash(secret, A(1) || seed) || \\ & HMAC_hash(secret, A(2) || seed) || \\ & HMAC_hash(secret, A(3) || seed) || \dots \end{aligned} \quad (13)$$

Gde je $A()$ definisano kao:

$A(0) = \text{dodatak (seed)}$

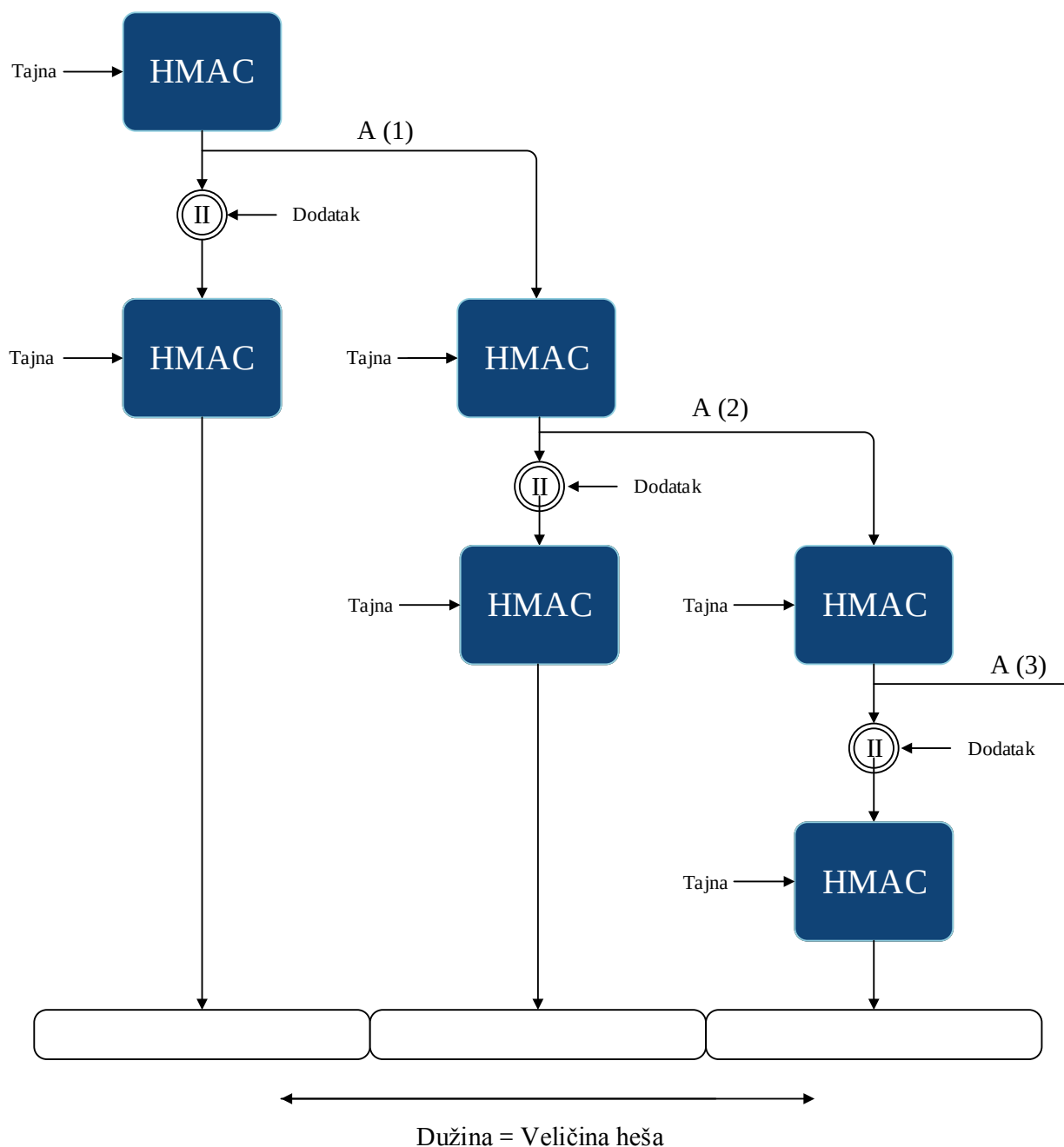
$A(i) = HMAC_hash(tajna, A(i-1))$

Funkcija proširenja podataka omogućava korišćenje HMAC algoritma sa MD5 ili SHA-1 kao osnovu heš funkcije. Kao što se vidi, može biti P_hash ponavlja onoliko puta koliko je potrebno da se proizvede tražena količina podataka. Na primer, ako je korišćen P_SHA-1 za generisanje 64 bajtova podataka, onda bi moralo da bude ponavljano četiri puta, proizvodeći 80 bajtova podataka od kojih će poslednjih 16 biti odbačeni. U ovom slučaju, P_MD5 će takođe morati da se ponavlja četiri puta, proizvodeći tačno 64 bajtova podataka.

Da bi PRF bio što sigurnije, koriste se dva heš algoritama na način koji bi trebalo da garantuje svoju bezbednost ukoliko bar jedan od algoritama ostaje bezbedan.

PRF se definiše kao:

$$PRF(secret, label, seed) = P_hash(S1, label || seed) \quad (14)$$



Slika 34: TLS funkcija P_hash (tajna, dodatak)

PRF kao ulazne informacije koristi tajnu vrednost, identifikacionu oznaku i pothranjujuću vrednost, i kreira rezultat proizvoljne dužine.

Kodovi upozorenja/Alert codes

TLS podržava sev kodove upozorenja definisane u SSLv3 sa izuzetkom no_certificate. Broj dodatnih kodova je definisan u TLS; od ovih, sledeći kodovi su uvek fatalni:

- `record_overflow`: TLS evidencija je primljena sa korisnim teretom (šifrovanim tekstom) čija dužina premašuje $2^{14} + 2048$ bajta, ili šifrovanim tekstom defirovanim na dužinu koja je veća od $2^{14} + 1024$ bajta.

- `unknown_ca`: Ispravan lanac sertifikata ili deo lanca je primljen ali je sertifikat odbijen jer CA sertifikata (nadležnost sertifikata) nije mogla da se locira ili nije mogla da se uporedi sa znanom i proverenom CA.

- `access_denied`: Ispravan sertifikat je primljen ali kada je pristupna kontrola primenjena, pošiljalac je odlučio da ne nastavi sa pregovaranjem.

- `decode_error`: poruka nije mogla da se dekodira jer je polje bilo van svog specifičnog obima ili je dužina same poruke bila neispravna.

- `protokol_version`: klijentska verzija protokola u pokušaju pregovaranja je prepoznata ali nije podržana.

- `insufficient_security`: vraćeno umesto `handshake_failure` kada je pregovaranje propalo posebno zbog razloga što server zahteva mnogo sigurnije šifre u odnosu na one koje klijent podržava.

- `unsupported_extension`: poslato od strane klijenta koji je primio produženi serverski hello koja sadrži ekstenziju koja nije u skladu sa odgovarajućim klijentskim hello.

- `internal_error`: interna greška koja nije u vezi sa srodnikom ili samom ispravnošću protokola onemogućava dalji rad.

- `decrypt_error`: neuspela kriptografska operacija sinhronizacije, uključujući i nemogućnost da se verifikuje potpis, dešifruje razmena ključeva, ili potvrdi završna poruka.

Preostala upozorenja obuhvataju sledeće:

- `user_canceled`: ova sinhronizacija je otkazana iz nekog razloga koji nije u vezi sa neuspešnim protokolom.

- `no_renegotiation`: poslato od strane klijenta kao odgovor na hello zahtev ili od strane servera kao odgovor na klijentski hello nakon inicijalne sinhronizacije. Svaka od ovih poruka bi obično kao rezultat donela obnovljeno pregovaranje, ali ova uzbuna ističe da pošiljalac nije u mogućnosti da nastavi pregovaranje. Ova poruka uvek predstavlja upozorenje.

Šifrat programi

Postoji nekoliko manjih razlika između šifrat programa koji su dostupni pod SSLv3 i pod TLS:

- Razmena ključeva/Key Exchange: TLS podržava sve tehnike razmene ključeva iz SSLv3 sa izuzetkom Fortezza-e.
- Simetrični algoritmi enkripcije/Symmetric Encryption Algorithms: TLS obuhvata sve simetrične algoritme enkripcije koji se mogu naći u SSLv3, sa izuzetkom Fortezza-e.

Vrste klijentskih sertifikata

TLS definiše sledeće vrste sertifikata koje se zahtevaju u certificate_request poruci: rsa_sign, dss_sign, rsa_fixed_dh, i dss_fixed_dh. Svi ovi su definisani u SSLv3. Pored toga, SSLv3 obuhvata rsa_ephemeral_dh, dss_ephemeral_dh, i fortezza_kea. Privremeni Diffie-Hellman obuhvata potpisivanje Diffie-Hellman parametara sa RSA ili DSS. Što se tiče TLS-a, rsa_sign i dss_sign vrste se koriste za tu fnkciju; posebna vrsta potpisivanja nije neophodna kako bi se potpisali Diffie-Hellman parametri. TLS ne obuhvata Fortezza šemu.

Certificate_verify i završne poruke

U TLS certificate_verify poruci, MD5 i SHA-1 haševi se računaju samo putem handshake_messages. Podsećanja radi za SSLv3, haš računica je takođe obuhvatala i master tajnu i potporu. Ova dodatna polja ne doprinose nikakvoj dodatnoj sigurnosti.

Što se tiče završne poruke u SSLv3, završna poruka u TLS predstavlja haš zasnovan na zajedničkoj master_secret, prethodnim porukama sinhronizacije i oznaci koja identifikuje klijenta ili server. Računica se donekle razlikuje. Za TLS, računica je sledeća

$$\begin{aligned} & \text{PRF}(\text{master_secret}, \text{finished_label}, \text{MD5}(\text{handshake_messages}) || \\ & \quad \text{SHA-1}(\text{handshake_messages})) \end{aligned} \tag{15}$$

gde finished_label predstavlja kariku “klijent završio” za klijenta i “server završio” za server.

Pre_master_secret za TLS se računa na isti način kao i za SSLv3. Što se tiče SSLv3, master_secret za TLS se računa kao haš funkcija pre_master_secret i 2 hello slučajna broja. Forma TLS računice se razlikuje od forme za SSLv3 i definiše se na sledeći način:

$$\text{master_secret} = \text{PRF}(\text{pre_master_secret}, \text{"master secret"}, \\ \text{ClientHello.random} || \text{ServerHello.random}) \quad (16)$$

Ovaj algoritam se primenjuje sve dok se ne postigne 48 bajta pseudo slučajnih izlaznih informacija. Računica glavnog blok materijala (MAC tajni ključevi, enkripcioni ključevi sesije, i IV-e) definiše se na sledeći način:

$$\text{key_block} = \text{PRF}(\text{master_secret}, \text{"key expansion"}, \\ \text{SecurityParameters.server_random} || \\ \text{SecurityParameters.client_random}) \quad (17)$$

dok se ne generiše dovoljno izlaznih informacija. Što se tiče SSLv3, key_block predstavlja funkciju master_secret i klijentskih i serverskih slučajnih brojeva, ali trenutni algoritam za TLS se razlikuje.

Umetanje

U SSL, umetanje koje je dodato pre enkripcije korisnikovih podataka predstavlja minimalnu zahtevanu količinu tako da je ukupna veličina podataka koji trebaju da budu enkriptovani višestruka dužina bloka šifre. U TLS, umetanje može da bude bilo koja količina koja kao rezultat ima total koji je višestruka dužina bloka šifre, do maksimalno 255 bajta. Na primer, ukoliko nešifrovani tekst (ili kompresovani tekst ukoliko se kompresovanje primenjuje) plus MAC plus umetanje daje dužinu od 79 bajta onda dužina umetanja (izraženo u bajtovima) može da bude 1, 9, 17, i tako dalje, sve do 249. Promenljiva dužina umetanja može da se koristi kako bi se osujetili napadi zasnovani na analizi dužine razmenjenih poruka.

3.5. HTTPS

HTTPS (HTTP preko SSL) se odnosi na kombinaciju HTTP i SSL kako bi se implementirala sigurna komunikacija između veb pretraživača i mrežnog servera. Mogućnost postojanja HTTPS-a je ugrađena u sve moderne veb pretraživače. Upotreba istog zavisi od

veb servera koji podržava HTTPS komunikaciju. Na primer, neki pretraživački uređaji ne podržavaju HTTPS⁴⁶.

Glavna razlika koja može da se vidi od strane korisnika veb pretraživača jeste da URL (uniform resource locator – nepromenljiv detektor resursa) adrese počinju sa HTTPS:// a ne sa HTTP://. Normalna HTTP veza koristi port 80. Ukoliko je HTTPS određen, koristi se port 443, što „poziva“ SSL.

Kada se koristi HTTPS, sledeći elementi komunikacije se enkriptuju:

- URL zatraženog dokumenta
- sadržaj dokumenta
- sadržaj pretraživačkih formi (koje popunjava korisnik pretraživača)
- „kolačići“ koji se šalju od pretraživača ka serveru i obrnuto
- sadržaj HTTP zaglavlja

HTTPS se dokumentuje u RFC 2818, HTTP preko TLS. Ne postoji neka značajna promena pri korišćenju HTTP-a putem SSL ili TLS-a, i obe primene se nazivaju HTTPS.

3.5.1. Iniciranje veze

Za HTTPS, činilac koji se ponaša kao HTTP klijent takođe predstavlja i TLS klijenta. Klijent inicira vezu ka serveru na odgovarajućem portu a onda šalje TLS ClientHello kako bi se započela TLS sinhronizacija. Kada se TLS sinhronizacija završi, klijent može onda da inicira prvi HTTP zahtev. Svi HTTP podaci se šalju kao TLS aplikacioni podaci. Normalno HTTP ponašanje, uključujući i zadržane veze, bi trebalo pratiti.

Potrebno je da se razjasni da postoje 3 nivoa svesnosti o vezi u HTTPS. Na HTTP nivou, HTTP klijent zahteva vezu na HTTP serveru time što šalje zahtev za vezu na sledeći najniži sloj. Obično je taj sledeći najniži sloj TCP, ali isti može takođe biti i TLS/SSL. Na TLS nivou, sesija se uspostavlja između TLS klijenta i TLS servera. Ova sesija može da podrži jednu ili više veza u bilo koje vreme. Kao što je već viđeno, TLS zahtev da se

⁴⁶ Implementing SSL / TLS Using Cryptography and PKI; Joshua Davies (Jan 11, 2011)

uspostavi veza počinje sa uspostavom TCP veze između TCP entiteta sa klijentove strane i TCP entiteta sa serverske strane.

3.5.2. Zatvaranje veze

HTTP klijent ili server mogu da istaknu zatvaranje veze time što će uključiti sledeći red u HTTP evidenciju: `connection: close`. Isto naglašava da će veza biti zatvorena nakon što se ova evidencija prenese.

Zatvaranje HTTPS veze zahteva da TLS zatvori vezu sa srodnim TLS entitetom daljinskim putem, što bi obuhvatilo zatvaranje potporne TCP veze. Na TLS nivou, pravilan način da se zatvori veza je da svaka strana upotrebi TLS protokol upozorenja kako bi poslala `close_notify` upozorenje. TLS implementacije moraju da iniciraju upozorenja pred zatvaranje pre zatvaranja same veze. TLS implementacija može i, nakon slanja upozorenja pred zatvaranje, zatvoriti samu vezu bez čekanja da druga strana pošalje svoje upozorenje pred zatvaranje, time stvarajući “nepotpuno zatvaranje”. Pažnja se treba obratiti da implementacija koja radi pomenuto može odabrati da ponovo upotrebi sesiju. Isto treba da se uradi samo u slučaju kada aplikacija zna (obično putem otkrivanja granica HTTP poruka) da su primljeni svi podaci poruke koji su joj od interesa.

HTTP klijenti moraju takođe znati da reaguju u situaciji kada se potporna TCP veza završava bez prethodnog `close_notify` upozorenja i bez `connection: close` indikatora. Do takve situacije može da dođe zbog programske greške na serveru ili komunikacione greške koja izazova da se TCP veza obustavi.

Međutim, nenajavljeno TCP zatvaranje može da bude i dokaz neke vrste napada. HTTPS klijent bi trebalo da izda neku vrstu bezbednosnog upozorenja svaki put kada se isto desi.

3.6. Sigurnosni omotač (SSH – secure shell)

Sigurnosni omotač (SSH) predstavlja protokol za sigurnu mrežnu komunikaciju koja je osmišljena da bude relativno jednostavna i ne toliko skupa za implementaciju. Inicijalna verzija, SSH1 je bila fokusirana na pružanje sigurnog daljinskog pristupa koji bi zamenio TELNET i ostale daljinske šeme pristupa koje nisu pružale sigurnost. SSH takođe pruža i

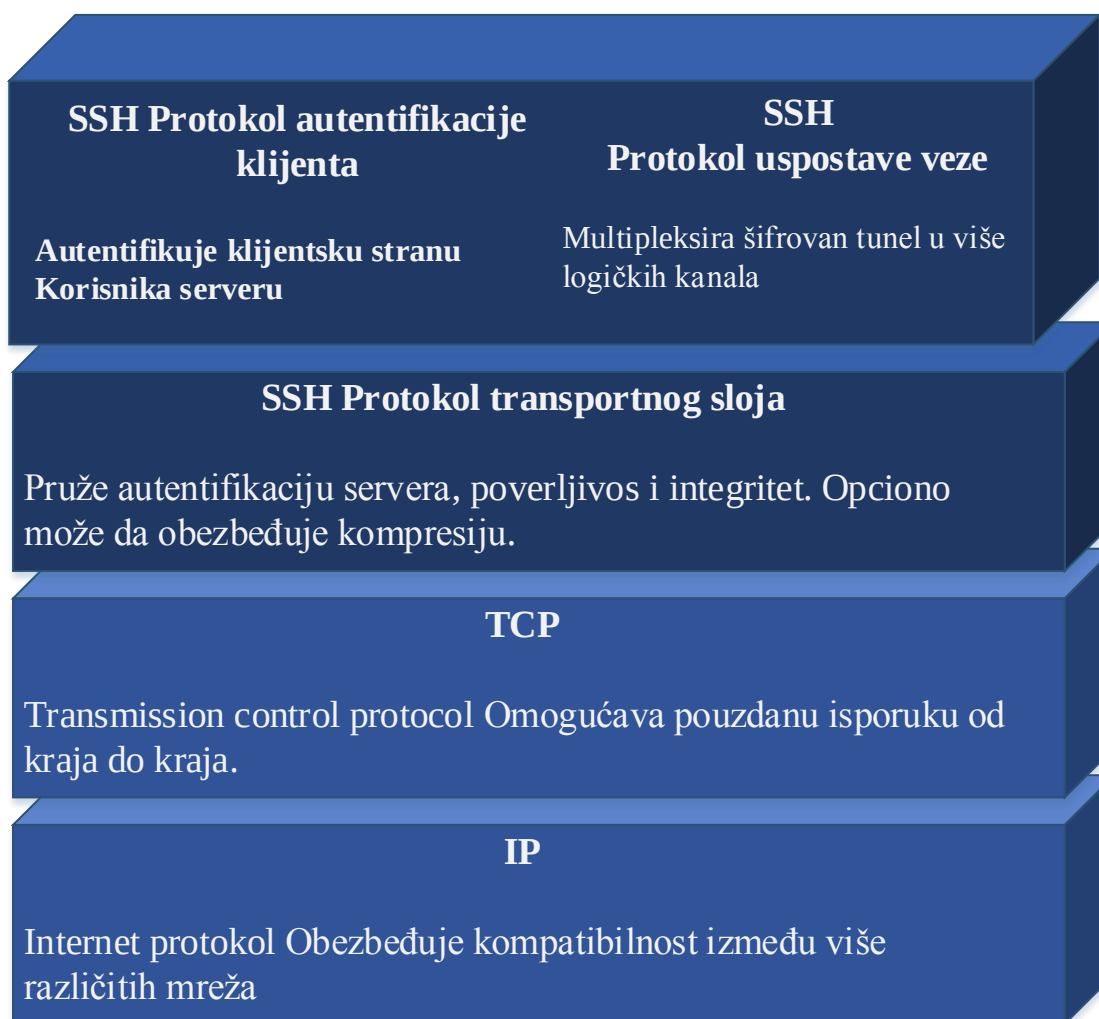
opštiju klijentsko/serversku mogućnost i može da se koristi za mrežne funkcije kakve su npr. prenos fajlova i e-mail slanje. Nova verzija zvana SSH2, rešava određeni broj sigurnosnih nedostataka koji postoje u toj originalnoj šemi. SSH2 se vodi kao predloženi standard u IETF RFCs 4250 putem 4256 ⁴⁷.

SSH klijent i server aplikacije su u širokoj upotrebi za većinu operativnih sistema. Isto je postalo samo stvar izbora za daljinsko pristup, a sve brže postaje i jedna od najsveobuhvatnijih aplikacija za tehnologiju enkripcije van aplikacija koje su integrisane u samim operativnim sistema.

SSH je sastavljen iz 3 protokola koja obično funkcionišu na TCP-u (Slika 35):

- Protokol sloja prenosa/Transport Layer Protokol: pruža autorizaciju servera, poverljivost podataka, kao i integritet podataka sa prenesenim očuvanjem tajnosti (tj, ukoliko je ključ kompromitovan tokom jedne sesije, to saznanje ne utiče na bezbednost prethodne sesije). Prenosni sloj može opcionalno da pruži i kompresovanje.
- Protokol autentifikacije klijenta/User Authentication Protokol: autorizuje korisnika serveru.
- Protokol veze/Connection Protokol: umnožava višestruke logičke kanale komunikacije putem jedinstvene, potporne SSH veze.

⁴⁷ Hiding Web Traffic with SSH: How to Protect Your Internet Privacy against Corporate Firewall or Insecure Wireless; Slava Gomzin (Apr 24, 2012)



Slika 35: SSH Protokol

3.6.1. Transportni Protokol

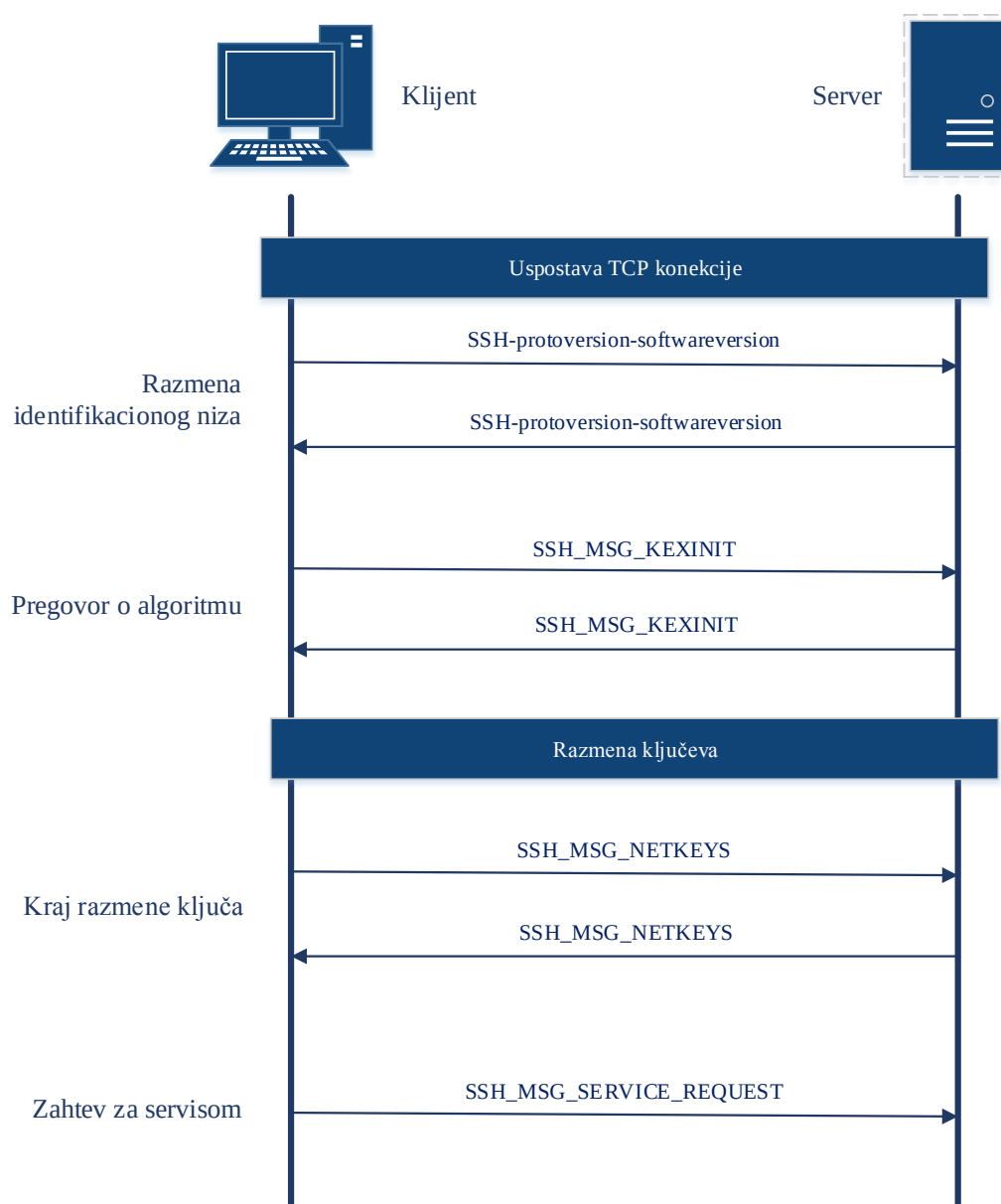
Autorizacija servera se dešava na transportnom sloju, na osnovu servera koji poseduje par javni/privatni ključ. Server može da ima višestruke matične ključeve koristeći višestruke različite, asimetrične algoritme enkripcije. Višestruki „domaćini“ mogu deliti isti matični ključ. U svakom slučaju, serverski matični ključ se koristi u toku razmene ključeva kako bi se dokazala autentičnost identiteta domaćina. Da bi isto bilo moguće, klijent mora da ima prethodno poznavanje serverskog javnog matičnog ključa. RFC 4251 nameće dva alternativna modela od poverenja koji mogu da se iskoriste⁴⁸:

⁴⁸ Hiding Web Traffic with SSH: How to Protect Your Internet Privacy against Corporate Firewall or Insecure Wireless; Slava Gomzin (Apr 24, 2012)

1. Klijent poseduje lokalnu bazu podataka koja povezuje ime svakog domaćina (kako je sam korisnik uneo) sa odgovarajućim javnim matičnim ključem. Ova metoda ne zahteva infrastrukturu centralnog upravljanja kao ni koordinaciju treće strane. Mana istog može da bude da baza podataka asocijacija ime-ključ može da postane isuviše opterećujuća za održavanje.

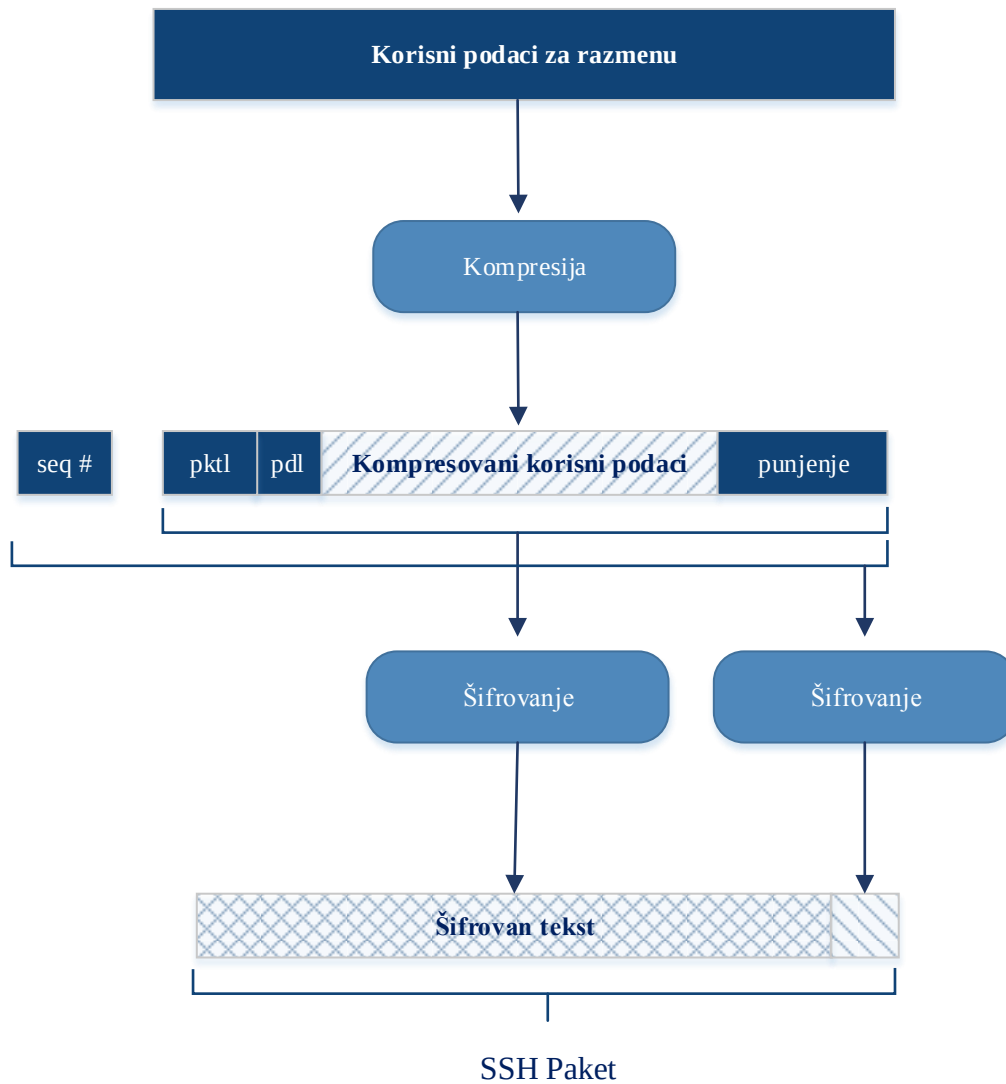
2. Domaćinska asocijacija ime-ključ proverena je od strane bezbedonosne nadležnosti (CA – certified authority). Klijent poznaje samo CA bazni ključ i može potvrditi ispravnost svih domaćinskih ključeva koji su provereni od strane prihvaćenih CA-i. Ova alternativa olakšava problem održavanja, jer u idealnom slučaju, samo jedinstveni CA ključ treba da bude sigurno uskladišten kod klijenta. Sa druge strane, svaki domaćinski ključ mora da bude zvanično proveren od strane bezbedonosnih nadležnosti pre nego što uopšte dođe do autorizacije.

RAZMENA PAKETA (Slika 36) objašnjava sekvencu događaja na SSH protokolu sloja prenosa. Kao prvo, klijent uspostavlja TCP vezu sa serverom. Ovo se radi putem TCP protokola i ne predstavlja deo protokola sloja prenosa.



Slika 36: SSH Protokol Transportnog sloja

Jednom kada je veza uspostavljena, klijent i server razmenjuju podatke, i isti se nazivaju paketi, u polju podataka na TCP segmentu. Svaki paket je sledećeg formata (Slika 37).



Pktl = Dužina paketa

Pdl = Dužina punjenja

Slika 37: Formiranje SSH transportnog paketa

- Dužina paketa/Package length: dužina paketa u bajtovima, ne uključujući samu dužinu paketa i MAC polja.
- Dužina umetanja/Embedding length: dužina slučajnih umetnutih polja.
- Korisni teret/Payload: koristan sadržaj paketa. Pre pregovaranja oko algoritma, ovo polje nije kompresovano. Ukoliko se kompresovanje ispregovara, onda u narednim paketima ovo polje postaje kompresovano.

- Slučajno umetanje/Random Embedding: jednom kada se algoritam enkripcije ispregovara, ovo polje se dodaje. Isto sadrži nasumične bajtove umetanja tako da je ukupna dužina paketa (ne računajući MAC polja) višestruke veličine bloka šifre, ili 8 bajta za niz šifara.

- Message authentication code (MAC): ukoliko je poruka autorizacije ispregovarana, ovo polje sadrži MAC vrednost. MAC vrednost se računa preko celog paketa plus broja sekvence, ne računajući MAC polja. Broj sekvence je implicitna 32-bitna paket sekvenca koja se inicira od nule za prvi paket a onda raste za svaki sledeći paket. Broj sekvence nije uključen u paket koji se šalje preko TCP veze.

Jednom kada se algoritam enkripcije ispregovara, čitav paket (ne računajući MAC polja) se enkriptuje nakon što se MAC vrednost preračuna.

Paket razmena SSH prenosnog sloja sastoji se od sekvenci koraka (Slika 36). Prvi korak, identifikacija razmene niza počinje sa klijentovim slanjem paketa sa identifikacionim nizom u sledećoj formi:

SSH-protoversion-softwareversion SP comments CR LF (18)

gde SP, CR, i LF predstavljaju prazan karakter, povraćaj prenosa i linijsko snabdevanje (tim redosledom).

Primer validnog niza je SSH-2.0-billsSSH_3.6.3q3<CR><LF>. Server odgovara svojim sopstvenim identifikacionim nizom. Ovi nizovi se koriste u Diffie-Hellman razmeni ključeva.

Sledeće na red dolazi pregovaranje oko algoritma. Svaka strana šalje SSH_MSG_KEXINIT koji sadrže listu podržanih algoritama u skladu sa preferencama pošiljaoca. Postoji jedna lista za svaku vrstu kriptografskog algoritma. Algoritmi obuhvataju razmenu ključeva, enkripciju, MAC algoritam, i algoritam kompresovanja. Slika 38 prikazuje dostupne mogućnosti za enkripciju, MAC, i kompresovanje. Za svaku kategoriju, izabrani algoritam je prvi algoritam na klijentovoj listi koji je takođe podržan i od strane servera.

Šifrovanje	
3des-cbc*	Trostruki ključ 3DES u CBC modu
Blowfish-cbc	Blowfish u CBC modu
twofish256-cbc	Twofish u CBC modu sa 256-bitnim ključem
twofish192-cbc	Twofish sa 192-bitnim ključem
twofish128-cbc	Twofish sa 128-bitnim ključem
aes256-cbc	AES u CBC modu sa 256-bitnim ključem
aes192-cbc	AES sa 192-bitnim ključem
aes128-cbc**	AES sa 128-bitnim ključem
Serpent256-cbc	Serpent u CBC modu sa 256-bitnim ključem
Serpent192-cbc	Serpent sa 192-bitnim ključem
Serpent128-cbc	Serpent sa 128-bitnim ključem
arcfour	RC4 sa 128-bitnim ključem
Cast128-cbc	CAST-128 u CBC modu

Šifrovanje	
hmac-sha1*	HMAC-SHA1; Sažeta dužina = dužina ključa = 20
hmac-sha1-96*	Prvih 96 bita od HMAC-SHA1; Sažeta dužina=12; dužina ključa = 20
hmac-md5	HMAC-SHA1; Sažeta dužina = dužina ključa = 16
hmac-md5-96	Prvih 96 bita of HMAC-SHA1; Sažeta dužina=12; dužina ključa = 16

Šifrovanje	
bez*	Bez kompresije
zlib	Definisan u RFC 1950 i RFC 1951

*=Zahtevan
**=Preporučen

Slika 38: SSH Transportni sloj šifarski algoritmi

Sledeći korak je razmena ključeva. Specifikacija dozvoljava alternativne metode razmene ključeva ali za sada, samo dve verzije Diffie-Hellman razmene ključeva su obuhvaćene. Obe verzije se definišu u RFC 2409 i zahtevaju samo 1 paket u svakom pravcu. Naredni koraci su neophodni za razmenu. U istom, C je klijent; S je server; p je ogromna sigurnosna osnova; g predstavlja generator za GF podgrupu (p); q je redosled podgrupe; V_S je serverov identifikacioni niz; V_C je klijentov identifikacioni niz; K_S je serverov javni matični ključ; I_C je klijentova SSH_MSG_KEXINIT poruka a I_S je serverova SSH_MSG_KEXINIT poruka koje se razmenjuju pre nego što se dođe do ovog dela. Vrednosti p, g i q su poznate i klijentu i serveru kao rezultat odabira algoritma pregovaranja. Haš funkcija hash() se takođe određuje u toku algoritma pregovaranja.

- C generiše slučajni broj $x(1 < x < q)$ i računa $e = g^x \bmod p$. C šalje e ka S.
- S generiše slučajni broj $y(0 < y < q)$ i izračunava $f = g^y \bmod p$. S prima e . On računa $K = e^y \bmod p$, $H = \text{hash}(V_C \parallel V_S \parallel I_C \parallel I_S \parallel K_S \parallel e \parallel f \parallel K)$, i potpis s na H sa svojim privatnim ključem. S šalje $(K_S \parallel f \parallel s)$ ka C.
- C verifikuje da K_S je to zaista matični ključ za S. Za C je takođe dozvoljeno da prihvati ovaj ključ bez provere, ipak ovo bi moglo da učini protokol ranjivim na napade. C onda računa $K = f^x \bmod p$, $H = \text{hash}(V_C \parallel V_S \parallel I_C \parallel I_S \parallel K_S \parallel e \parallel f \parallel K)$, i proverava potpis s na H .

Kao rezultat ovih koraka, obe strane sada dele master ključ K . Pored toga, server je autorizovan za klijenta, jer je isti server upotrebio svoj privatni ključ da potpiše svoju polovinu Diffie-Hellman razmene. I na kraju, haš vrednost H služi kao identifikator sesije ove veze. Jednom kada se isti izračuna, identifikator sesije se ne menja, čak iako se razmena ključeva obavi ponovo na ovoj vezi kako bi se dobili novi ključevi.

Sam kraj razmene ključeva se signalizira razmenom `SSH_MSG_NEWKEYS` paketa. U tom trenutku, obe strane mogu da počnu da koriste ključeve koji se generišu iz K , kao što će kasnije i biti objašnjeno.

Krajnji korak je zahtev za uslugom. Klijent šalje `SSH_MSG_SERVICE_REQUEST` paket kako bi zatražio ili autorizaciju korisnika ili protokol veze. Nakon svega ovoga, svi podaci se razmenjuju kao koristan teret paketa SSH transportnog sloja zaštićeni enkripcijom i MAC-om.

GENERISANJE KLJUČEVA Ključevi koji se koriste za enkripciju i MAC (ako je neophodno i IV) se generišu iz zajedničkog tajnog ključa K , haš vrednosti iz razmene ključeva i iz identifikatora sesije, što je jednako sa H ukoliko nije došlo do kasnije razmene ključeva nakon inicijalne. Vrednosti su izračuna na sledeći način:

- Početni IV klijent ka serveru : $\text{HASH}(K \parallel H \parallel A \parallel \text{session_id})$
- Početni IV server ka klijentu : $\text{HASH}(K \parallel H \parallel B \parallel \text{session_id})$
- Ključ za šifrovanje klijent ka serveru : $\text{HASH}(K \parallel H \parallel C \parallel \text{session_id})$
- Ključ za šifrovanje server ka klijentu : $\text{HASH}(K \parallel H \parallel D \parallel \text{session_id})$
- Integritet ključ klijenta do servera : $\text{HASH}(K \parallel H \parallel E \parallel \text{session_id})$
- Integritet ključ server do klijenta : $\text{HASH}(K \parallel H \parallel F \parallel \text{session_id})$

gde HASH() predstavlja haš funkciju koja se određuje u toku algoritma pregovaranja.

3.6.2. Protokol autorizacije korisnika/User Authentication Protokol

Protokol autorizacije korisnika pruža sredstva kojima se klijent može autorizovati za server⁴⁹.

VRSTE I FORMATI PORUKA Tri vrste poruka se uvek koriste u protokolu autorizacije korisnika. Zahtevi za autorizaciju od strane klijenta imaju sledeći format:

- bajt SSH_MSG_USERAUTH_REQUEST
- korisničko ime niza/string user name
- korisničko ime usluge/string service name
- ime metode niza/string method name
- specifična polja metode/method specific fields

gde korisničko ime predstavlja autorizaciju identiteta koji klijent tvrdi da ima, ime usluge predstavlja mogućnost da klijent zatraži pristup (obično preko SSH protokola veze), a ime metode predstavlja metod autorizacije koji se koristi u ovom zahtevu. Prvi bajt ima decimalnu vrednost 50, što se tumači kao SSH_MSG_USERAUTH_REQUEST.

Ukoliko server (1) odbaci zahtev za autorizaciju ili (2) prihvati zahtev ali zatraži jednu ili više dodatnih metoda autorizacije, onda server šalje poruku sledećeg formata:

bajt SSH_MSG_USERAUTH_FAILURE
name-list authentications that can continue
boolean partial success

gde ime-lista predstavlja listu metoda koje mogu efikasno nastaviti dijalog. Ukoliko server prihvati autorizaciju, isti šalje jedinstvenu bajt poruku: SSH_MSG_USERAUTH_SUCCESS.

RAZMENA PORUKA Razmena poruka obuhvata naredne korake:

1. Klijent šalje SSH_MSG_USERAUTH_REQUEST bez određene metode.

⁴⁹ Mechanics of User Identification and Authentication: Fundamentals of Identity Management; Dobromir Todorov (Jun 18, 2007)

2. Server proverava da li je korisničko ime ispravno. Ukoliko nije, server vraća SSH_MSG_USERAUTH_FAILURE sa delimično lažnom vrednosti uspeha. Ukoliko je korisničko ime ispravno, server nastavlja sa korakom 3.
3. Server vraća SSH_MSG_USERAUTH_FAILURE sa listom jedne ili više metoda autorizacije koje trebaju da budu upotrebljene.
4. Klijent bira jednu od prihvatljivih metoda autorizacije i šalje SSH_MSG_USERAUTH_REQUEST sa imenom metode i zahtevanim poljima za tu metodu. U toj fazi, verovatno će postojati određeni niz razmena kako bi se metoda primenila.
5. Ukoliko autorizacija uspe a više metoda autorizacije se zahtevaju, server nastavlja sa korakom 3, koristeći delimično istinitu vrednost uspeha. Ukoliko autorizacija ne uspe, server nastavlja sa korakom 3, koristeći delimično lažnu vrednost uspeha.
6. Kada sve zahtevane metode autorizacije uspeju, server šalje SSH_MSG_USERAUTH_SUCCESS poruku, i protokol autorizacije se završava.

METODE AUTORIZACIJE Server može da zatraži jednu ili više sledećih metoda autorizacije.

- **publickey:** detalji ove metode zavise od odabranog algoritma javnog ključa. U principu, klijent šalje poruku serveru koji sadrži javni ključ klijenta, sa porukom potpisanom klijentovim privatnim ključem. Kada server primi tu poruku, isti proverava da li je obezbeđeni ključ prihvatljiv za autorizaciju i, ako je to slučaj, proverava da li je potpis ispravan.
- **password:** Klijent šalje poruku koja sadrži šifru nešifrovanog teksta, a koja je zaštićena enkrijpcijom u protokolu sloja prenosa.
- **hostbased:** autorizacija se izvršava na klijentovom „domaćinu“ a ne na samom klijentu. Stoga, domaćin koji podržava više klijenata mora da pruži autorizaciju sa sve svoje klijente. Ova metoda funkcioniše tako što klijent šalje potpis stvoren privatnim ključem klijentovog „domaćina“. Stoga, umesto da direktno odredi identitet korisnika, SSH server verifikuje identitet klijentovog „domaćina“ — a onda „veruje“ domaćinu kada isti potvrdi da se klijent već autorizovao sa svoje strane.

3.6.3. Protokol veze

SSH Protokol veze funkcioniše povrh SSH protokola sloja prenosa i pretpostavlja da se koristi sigurna veza autorizacije. RFC 4254, sigurnosni omotač (SSH) Protokola veze, potvrđuje da Protokol veze radi povrh protokola sloja prenosa kao i protokola autorizacije korisnika. RFC 4251, SSH Protokol arhitekture potvrđuje da protokol veze funkcioniše povrh protokola autorizacije korisnika. Zapravo, protokol veze radi putem protokola sloja prenosa, ali se pretpostavlja da je protokol autorizacije korisnika pre tog bio uspostavljen. Sigurna veza autorizacije, koja se drugačije naziva i tunel, koristi se da bi protokol veze umnožio broj logičkih kanala⁵⁰.

3.6.3.1. Kanalni mehanizam

Sve vrste komunikacije koje koriste SSH, kao što je npr terminalna sesija, podržane su putem pojedinačnih kanala. Svaka strana može otvoriti svoj kanal. Za svaki kanal, svaka strana koristi jedinstveni kanalni broj, koji ne mora biti isti na oba kraja. Protok kroz kanale se kontroliše uz pomoć prozornih mehanizama. Nijedan podatak ne može biti poslat ka kanalu dok nije primljena poruka koja ističe da je prozorni prostor dostupan.

Sam ciklus kanala prolazi kroz 3 faze: otvaranje kanala, prenos podataka i zatvaranje kanala. Kada svaka od strana želi da otvori novi kanal, istom alocira lokalni broj a onda šalje poruku sledeće forme:

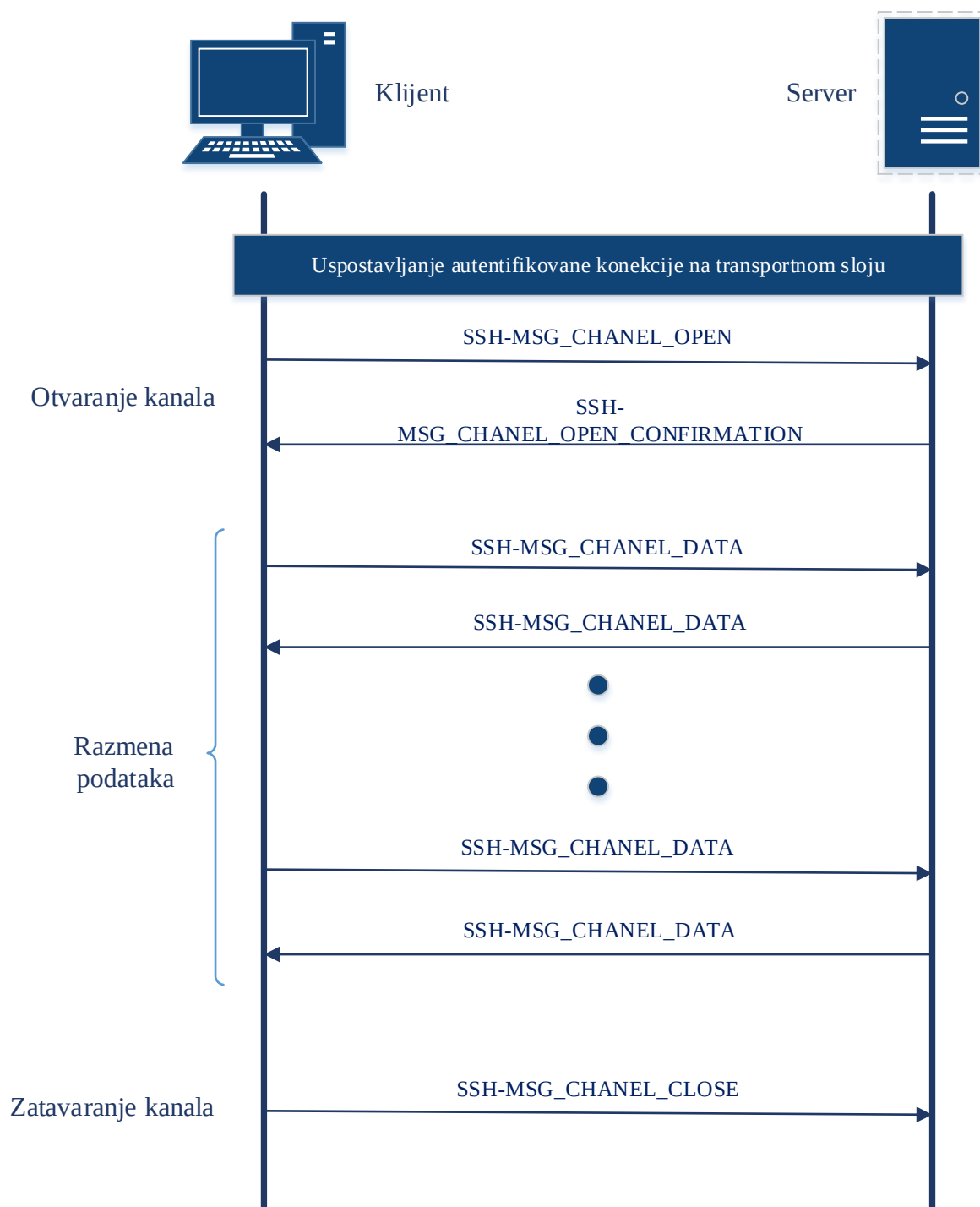
```
bajt    SSH_MSG_CHANNEL_OPEN
string  channel type
uint32  sender channel
uint32  initial window size
uint32  maximum packet size
....    channel type specific data follows
```

⁵⁰ Introduction to Modern Cryptography: Principles and Protocols (Chapman & Hall/CRC Cryptography and Network Security; Jonathan Katz and Yehuda Lindell (Aug 31, 2007)

gde uint32 predstavlja nepotpisanu 32-bitnu celu vrednost. Vrsta kanala identifikuje aplikaciju za ovaj kanal, kao što je dalje i objašnjeno. Kanal pošiljaoca je lokalni kanalni broj. Inicijalna veličina prozora određuje koliko bajta kanalnih podataka ove poruke mogu da se pošalju ka pošiljaocu bez prilagođavanja prozora. Maksimalna veličina paketa predstavlja maksimalnu veličinu pojedinačnih podataka paketa koji mogu biti poslani ka pošiljaocu. Na primer, neko bi možda želeo da iskoristi manji paket za interaktivnu vezu kako bi dobio bolji interaktivni odgovor sporijih linkova. Ukoliko je udaljena strana u mogućnosti da otvori kanal, ista vraća SSH_MSG_CHANNEL_OPEN_CONFIRMATION poruku, koja obuhvata kanalni broj pošiljaoca, kanalni broj primaoca, i prozor i vrednosti veličine paketa nadolazećeg saobraćaja. U suprotnom, udaljena strana vraća SSH_MSG_CHANNEL_OPEN_FAILURE poruku sa kodom objašnjenja koji ističe razlog neuspeha.

Jednom kada je kanal otvoren, prenos podata se vrši preko SSH_MSG_CHANNEL_DATA poruka, koje uključuju i kanalni broj primaoca i blok podataka. Ove poruke, u oba pravca, mogu da opstanu dok god je kanal otvoren.

Kada bilo koja od strana odluči da zatvori kanal, ista šalje SSH_MSG_CHANNEL_CLOSE poruku, koja obuhvata kanalni broj primaoca.



Slika 39: Primer razmene poruka na protokolu veze

VRSTE KANALA, četiri vrste kanala se prepoznaju u specifikaciji SSH protokola veze.

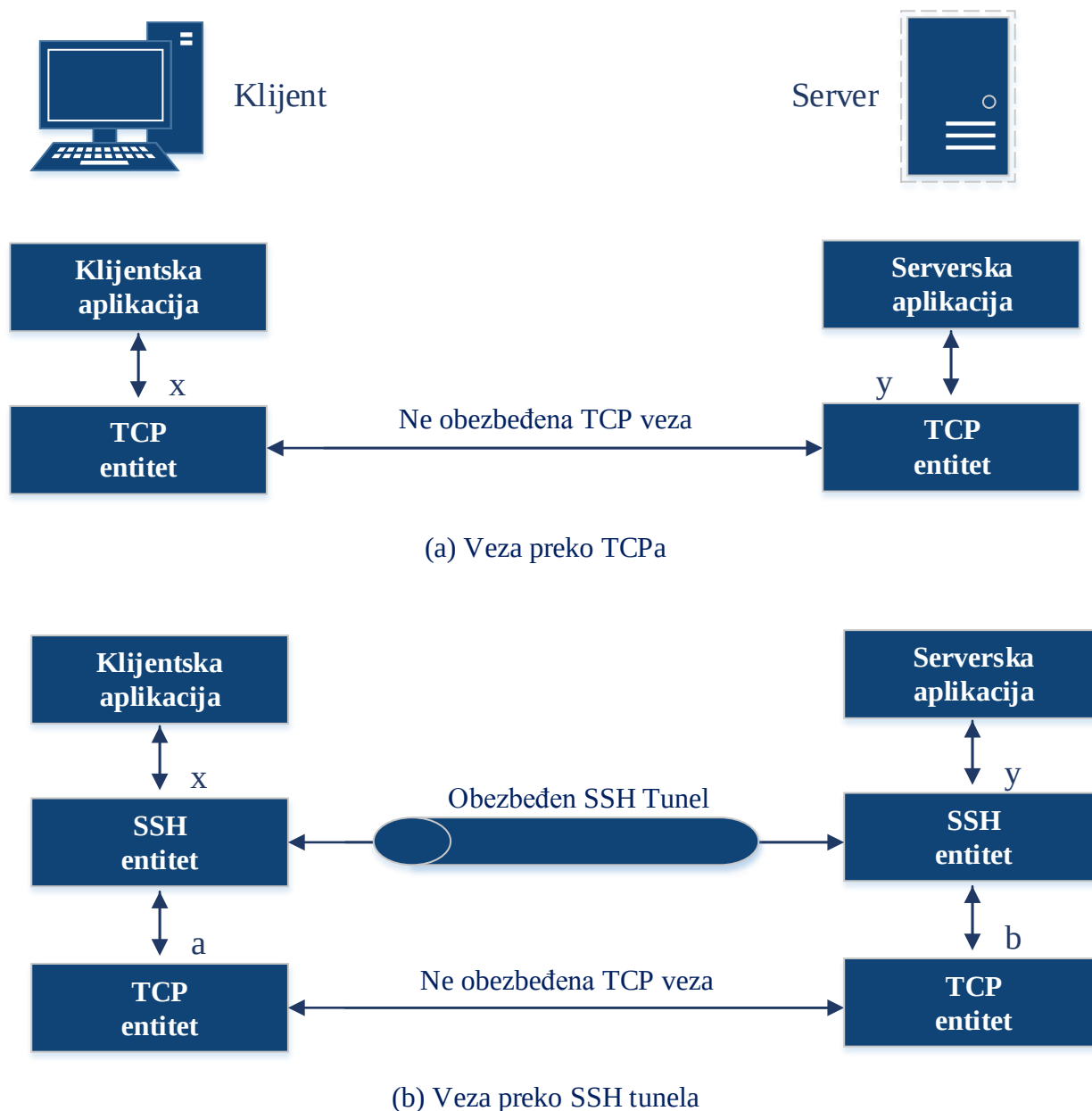
1. session: udaljeno zaustavljanje programa. Ovaj program može da bude omotač, aplikacija kao što su prenos fajlova ili e-mail slanje, sistemska komanda, ili neki ugrađeni podsistem. Jednom kada se kanal sesije otvori, sledeći zahtevi se koriste kako bi se pokrenuo udaljeni program.
2. x11: isto se odnosi na X Window sistem, kompjuterski softver sistem i mrežni protokol koji pruža interfejs grafičkog korisnika (graphical user interface-GUI) za umrežene kompjutere. X dozvoljava aplikaciji da funkcioniše na mrežnom serveru ali da bude prikazana na desktopu.
3. forwarded-TCPip: isto predstavlja udaljeno prosleđivanje porta, kao što je kasnije i objašnjeno.
4. direct-TCPip: isto predstavlja lokalno prosleđivanje porta, kao što je kasnije i objašnjeno.

PROSLEĐIVANJE PORTA Jedna od najkorisnijih SSH karakteristika je prosleđivanje porta. U principu, prosleđivanje porta pruža mogućnost da se bilo koja nesigurna TCP veza prebaci u sigurnu SSH vezu. Ovo se isto odnosi i na SSH „efekat tunela“. Potrebno je znati koji se port koristi u ovom kontekstu. Port je pokazatelj korisnika TCP-a. Stoga, svaka aplikacija koja funkcioniše povrh TCP ima port broj. Nadolazeći TCP saobraćaj se isporučuje odgovarajućoj aplikaciji na osnovu port broja. Aplikacija može da upotrebi više port brojeva. Na primer, za jednostavni protokol prenosa mejlova (Simple Mail Transfer Protokol-SMTP), serverska strana obično koristi port 25, tako da dolazeći SMTP zahtev koristi TCP i upućuje podatke ka portu 25. TCP prepoznaje da je to SMTP serverska adresa i usmerava podatke ka SMTP serverskoj aplikaciji.

Slika 40 predstavlja osnovni koncept koji stoji iza otpremanja porta. Postoji klijentska aplikacija koja je određena port brojem kao i serverska koja je određena port brojem. U nekom trenutku, klijentska aplikacija „poziva“ lokalni TCP entitet i zahteva vezu ka udaljenom serveru na portu. Lokalni TCP entitet pregovara TCP vezu sa udaljenim TCP entitetom, takvu da ta veza povezuje lokalni port x sa udaljenim portom y.

Kako bi se osigurala ova veza, SSH je sastavljen tako da SSH protokol sloja prenosa uspostavlja TCP vezu između SSH entiteta klijenta i servera sa TCP port brojevima a i b, tim redosledom. Sigurni SSH tunel se supostavlja preko ove TCP veze. Saobraćaj od strane

klijenta na portu x se preusmerava na lokalni SSH entitet i prosleđuje tunelom gde udaljeni SSH entitet isporučuje podatke serverskoj aplikaciji na portu y. Saobraćaj u suprotnom pravcu se slično preusmerava.



Slika 40: SSH transportni sloj, razmena podataka

SSH podržava dve vrste prosleđivanja porta: lokalno i daljinsko prosleđivanje. Lokalno prosleđivanje omogućava klijentu da uspostavi “kidnaperski” proces. Isto će presresti odabrani saobraćaj na nivou aplikacije i preusmeriti ga sa nesigurne TCP veze ka sigurnom SSH tunelu. SSH je sastavljen tako da „osluškuje“ odabrane portove. SSH „hvata“

sav saobraćaj koristeći odabrane portove i šalje iste kroz SSH tunel. Sa druge strane, SSH server šalje nadolazeći saobraćaj na željeni port koji određuje klijentska aplikacija.

Sledeći primer bi mogao da posluži da se razjasni lokalno prosleđivanje. Neka je pretpostavka da postoji e-mail klijent na desktopu personalnog računara i da se isti koristi za dobijanje mejlova sa mejl servera preko protokola pošte (Post Office Protokol-POP). Dodeljeni port broj za POP3 je port 110. Ovaj saobraćaj se može obezbediti na sledeći način:

1. SSH klijent uspostavlja vezu na udaljenom serveru.
2. Neka je odabran neki nekorišćeni lokalni port broj (npr. 9999), a SSH je sastavljen tako da prihvati saobraćaj sa ovog porta koji je usmeren ka portu 110 na serveru.
3. SSH klijent obaveštava SSH server da kreira vezu na toj destinaciji, u ovom slučaju na mejl serveru porta 110.
4. Klijent preuzima svaki podatak poslat ka lokalnom portu 9999 i šalje iste ka serveru unutar enkriptovane SSH sesije. SSH server dešifruje nadolazeće podatke i šalje nešifrovani tekst ka portu 110.
5. U drugom pravcu, SSH preuzima svaki primljen na portu 110 i šalje iste unutar SSH sesije nazad ka klijentu, koji iste dešifruje i šalje ih ka procesu povezanim sa portom 9999.

Kod daljinskog prosleđivanja, korisnikov SSH klijent deluje u ime servera. Klijent prima saobraćaj sa datom destinacijom port broja, „stavlja“ saobraćaj na ispravni port i šalje ga na destinaciju koju korisnik izabere. Tipičan primer daljinskog prosleđivanja je sledeći: zaposleni želi da pristupi serveru na poslu sa svog kućnog kompjutera. Radni server je iza „zaštitnog zida“ (firewall), pa stoga neće prihvatiti SSH zahtev sa kućnog kompjutera zaposlenog. Međutim, sa posla možete uspostaviti SSH tunel koristeći daljinsko prosleđivanje.

Isto uključuje naredne korake:

1. Sa radnog kompjutera se uspostavi SSH veza ka sa kućnom kompjuteru zaposlenog. „Zaštitni zid“ će dozvoliti isto jer je zaštićen odlazećom vezom.
2. SSH server se konfiguriše kako bi „osluškivao“ lokalni port (npr. 22), i isporučivao podatke preko SSH veze usmerene ka udaljenom portu (npr. 2222).
3. Sada je moguće doći do kućnog kompjutera zaposlenog, a sastav SSH će prihvatiti saobraćaj na portu 2222.
4. SSH tunel koji može da se koristi za daljinsku prijavu na radni server je formiran.

Naravno implementacija je moguća i na strani poslovnog rešenja ali zahteva konfiguraciju kojoj zaposleni iz bezbednosnih razloga najčešće nemaju pravo pristupa.

4. NOVI SISTEMI ANALIZE I ZAŠTITE RAČUNARARSKE MREŽE

U okviru ovog poglavlja rada biće predstavljene nove metode testiranja računarskih mreža, ukazano na moguće probleme. Eksperimentalna analiza obuhvatiće aktuelne probleme u najbrže rastućem tipu računarskih mreža, bežičnim računarskim mrežama. Predstavljena su aktuelna ali i sopstveno rešenje za testiranje integriteta bežične računarske mreže. Analiziran će biti napad po principu čovek u sredini, sa implementacijom novih metoda napada koje uključuju i napad na saobraćaj štićen prethodno objašnjenom metodom SSL i HTTPS saobraćaja. Nakon analize problema aktuelnih rešenja biće prikazano sopstveno rešenje za sprečavanje ili neutralisanje ove i sličnih vrsta napada metodom višekanalne autentifikacije.

4.1. Analiza WPA2 PSK računarske mreže

U ovom segmentu rada predstavljena je analiza pristupa bežičnoj računarskoj mreži koja se štiti trenutno aktuelnim i podrazumevano najbezbednijim rešenjem koje bi trebalo da garantuje bezbedan rad svih korisnika takvog sistema.

U cilju prikazivanja realnih rezultata definisan je referentni model računara koji će se koristiti u eksperimentalnoj analizi. Koristi se prenosivi računar konfiguracije prikazane u narednoj tabeli (Tabela 4). Od velikog značaja je vrsta uređaja za povezivanje na bežične mreže (eng. Wireless Card) jer, kao što će se videti u daljem radu, uređaj mora da podržava poseban režim rada koji omogućava računaru osluškivanje mreže.

Procesor:	2.4GHz Intel Core 2 Duo
RAM Memorija:	4GB (2x 2GB SO-DIMMs) 1066MHz DDR3 SDRAM
GPU:	NVIDIA GeForce 320M (256MB of DDR3 SDRAM)
Hard Disk:	128GB SSD SATA 6.0Gps
Wi-Fi Kartica:	Broadcom 802.11n Network Adapter

Tabela 4: Referentni računar

Operativni sistem korišćen za izvođenje eksperimenta je distribucija Linuks (eng. Linux) operativnog sistema pod nazivom Bektrek 5R3 (eng. Backtrack 5R3). Bektrek

predstavlja jednu od najuspešnijih Linuks distribucija sastavljenih specijalno za penetracijsko testiranje. Backtrack već poseduje sve alate potrebne za izvođenje testova prikazanih u narednim delovima dokumenta⁵¹.

Nakon teorijskih osnova prikazuju se koraci potrebni za „hvatanje“ paketa u komunikaciji između korisnika i pristupne tačke. Sadržaj ovih paketa kod mreže zaštićene WPA i WPA2 algoritmom je veoma sličan, a postupak hvatanja je potpuno isti, pa će postupak biti opisan radeći sa mrežom zaštićenom WPA2 algoritmom. Svi eksperimenti su izvršeni na način takav da nije ugrožena privatnost privatnih i produkcijskih bežičnih računarskih mreža⁵².

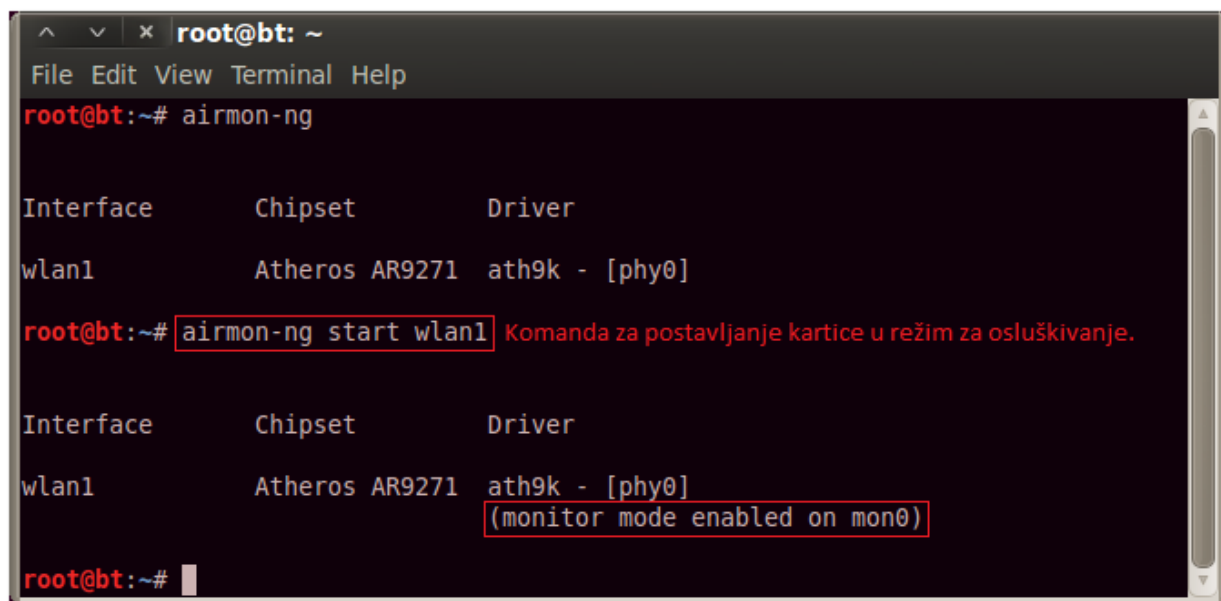
Po startovanju Backtrack operativnog sistema potrebno je izvesti sledeće korake:

4.1.1. Prvi korak napada na zaštićenu bežičnu računarsku mrežu

Prvi korak predstavlja prebacivanje kartice u monitor režim koji omogućava prisluškivanje saobraćaja na mrežama u dometu. Potrebno je da računar poseduje karticu koju je moguće prebaciti u monitor mod ili nabaviti USB karticu koja ovo omogućava. Najjednostavniji način za prebacivanje kartice u režim za osluškivanje je korišćenje Linuks distribucije, kao što se i u ovom primeru pokazuje, sa posebnim setom drajvera pisanih isključivo za ovu namenu. Neki proizvođači su čak videli i tržišnu priliku za kartice sa mogućnostima prebacivanja u monitor mod pa nude posebne kartice i posebne drajvere koje omogućavaju rad pod Windows okruženjem.

⁵¹ The Basics of Hacking and Penetration Testing: Ethical Hacking and Penetration Testing Made Easy (Syngress Basics; Patrick Engebretson (Aug 4, 2011))

⁵² Violent Python: A Cookbook for Hackers, Forensic Analysts, Penetration Testers and Security Engineers; TJ O'Connor (Nov 22, 2012)



```
root@bt: ~  
File Edit View Terminal Help  
root@bt:~# airmon-ng  
  
Interface      Chipset      Driver  
wlan1          Atheros AR9271  ath9k - [phy0]  
  
root@bt:~# airmon-ng start wlan1 Komanda za postavljanje kartice u režim za osluškivanje.  
  
Interface      Chipset      Driver  
wlan1          Atheros AR9271  ath9k - [phy0]  
                (monitor mode enabled on mon0)  
  
root@bt:~#
```

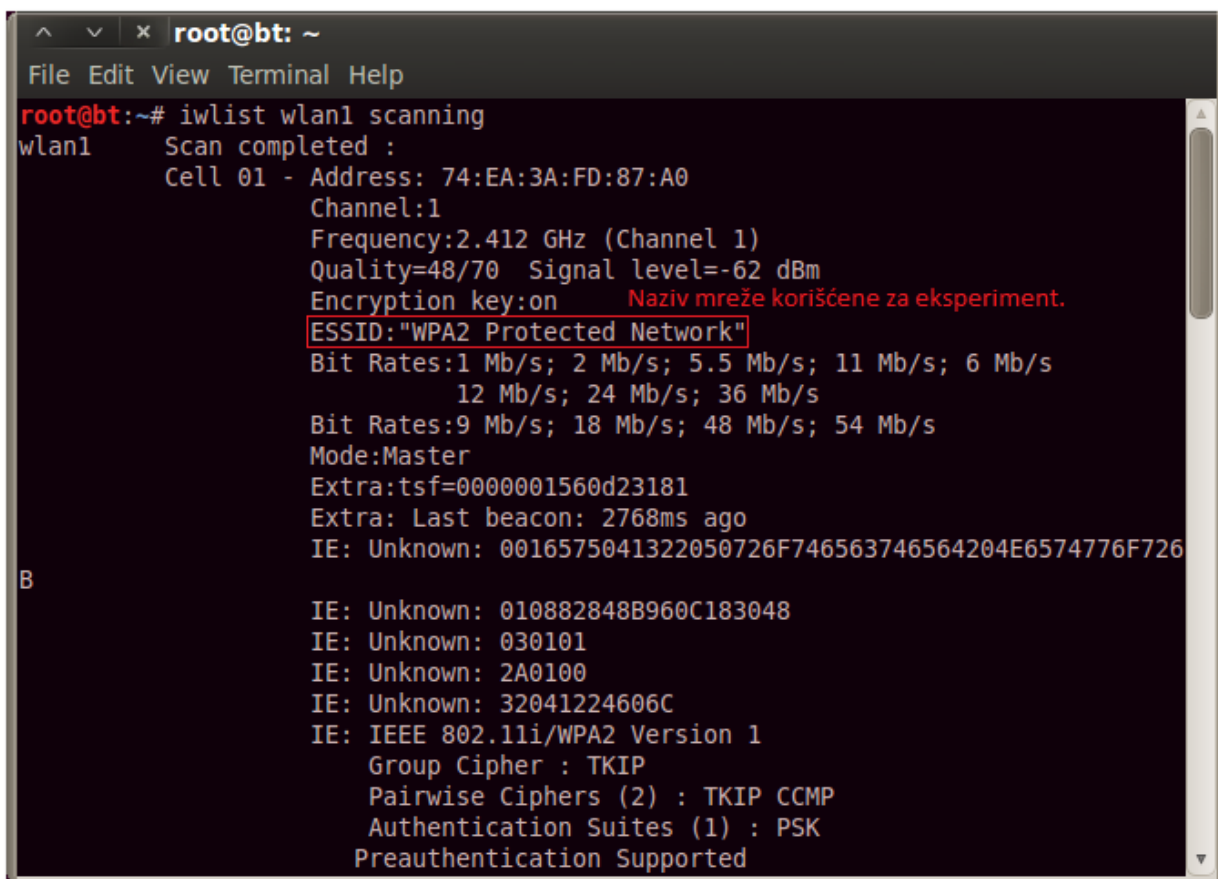
Slika 41: Prebacivanje kartice u mod za osluškivanje

Lažiranje MAC ili proizvođače adrese takođe nije moguće na svim bežičnim mrežnim karticama, a nekada je potreban deo procesa kompromitovanja bežične mreže. Ovo se radi u slučaju uključene opcije filtriranja MAC adresa na ruteru ili pristupnoj tački koja je meta.

Nakon izvršavanja komandi prikazanih na slici, kreira se virtuelni interfejs (mon0) koji predstavlja karticu (wlan0) u monitoring modu. Dobijeni interfejs biće korišćen u daljim koracima izvršavanja napada.

4.1.2. Drugi korak napada na zaštićenu bežičnu računarsku mrežu

Drugi korak predstavlja prikupljanje informacija o okolnim bežičnim mrežama. Na slici se jasno vidi da adapter prikuplja informacije o svim tipovima mreža, čak i o skrivenim mrežama. Prikazuje tip protokola, kanal na kom mreža emituje signal, jačinu signala, čak i klijente stanice.



```

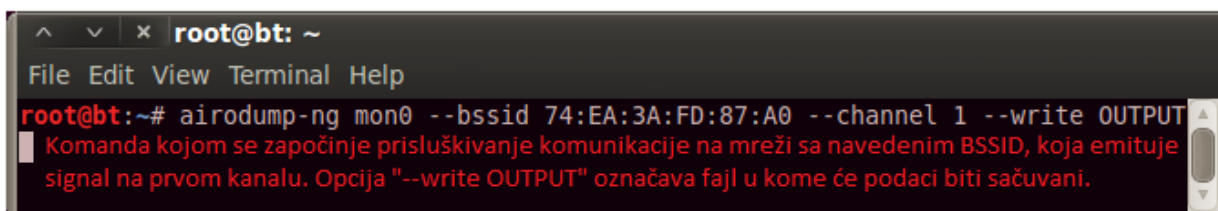
root@bt: ~
File Edit View Terminal Help
root@bt:~# iwlist wlan1 scanning
wlan1    Scan completed :
          Cell 01 - Address: 74:EA:3A:FD:87:A0
                   Channel:1
                   Frequency:2.412 GHz (Channel 1)
                   Quality=48/70  Signal level=-62 dBm
                   Encryption key:on      Naziv mreže korišćene za eksperiment.
                   ESSID:"WPA2 Protected Network"
                   Bit Rates:1 Mb/s; 2 Mb/s; 5.5 Mb/s; 11 Mb/s; 6 Mb/s
                               12 Mb/s; 24 Mb/s; 36 Mb/s
                   Bit Rates:9 Mb/s; 18 Mb/s; 48 Mb/s; 54 Mb/s
                   Mode:Master
                   Extra:tsf=00000001560d23181
                   Extra: Last beacon: 2768ms ago
                   IE: Unknown: 0016575041322050726F746563746564204E6574776F726
B
                   IE: Unknown: 010882848B960C183048
                   IE: Unknown: 030101
                   IE: Unknown: 2A0100
                   IE: Unknown: 32041224606C
                   IE: IEEE 802.11i/WPA2 Version 1
                       Group Cipher : TKIP
                       Pairwise Ciphers (2) : TKIP CCMP
                       Authentication Suites (1) : PSK
                       Preauthentication Supported

```

Slika 42: Prikaz rezultata skeniranja okolnih mreža

4.1.3. Treći korak napada na zaštićenu bežičnu računarsku mrežu

U trećem koraku odlučuje se koja će mreža biti napadnuta i od tog trenutka pa nadalje se prisluškuje samo komunikacija na toj mreži i podaci na istoj. Lokalna kartica se vezuje za BSSID mreže i kanal na kojem mreža pod napadom emituje signal. Od najvećeg interesa za napadača predstavljaju poruke sinhronizacije/uspostave (hendšejk) opisane u teorijskom delu rada.



```

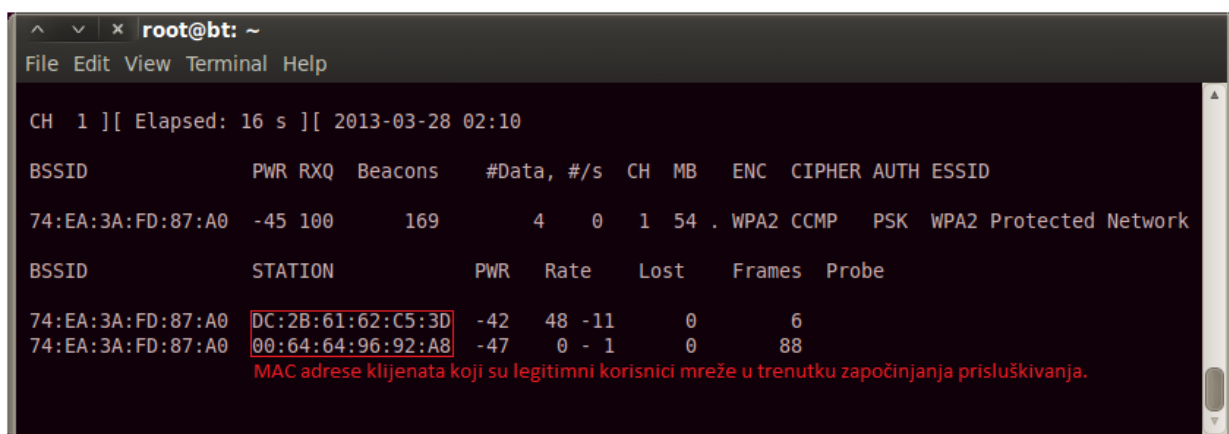
root@bt: ~
File Edit View Terminal Help
root@bt:~# airodump-ng mon0 --bssid 74:EA:3A:FD:87:A0 --channel 1 --write OUTPUT
Komanda kojom se započinje prisluškivanje komunikacije na mreži sa navedenim BSSID, koja emituje
signal na prvom kanalu. Opcija "--write OUTPUT" označava fajl u kome će podaci biti sačuvani.

```

Slika 43: Prikaz otpočinjanja prisluškivanja

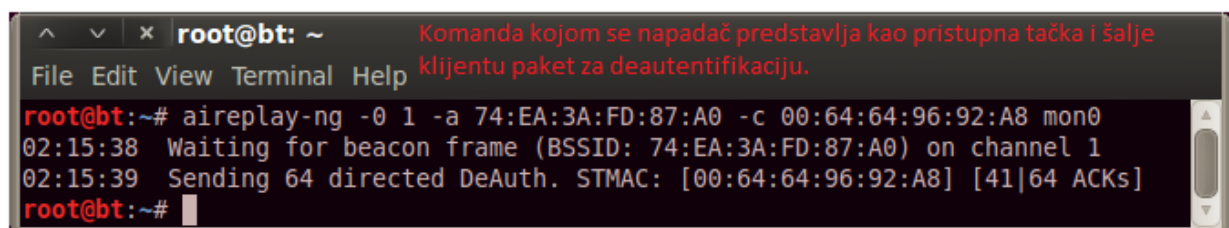
4.1.4. Četvrti korak napada na zaštićenu bežičnu računarsku mrežu

Dolazi se do hvatanja podataka u komunikaciji između legitimnog korisnika mreže i pristupne tačke. Ako se otkrije da je korisnik član mreže u trenutku napada, to govori napadaču da je već prošao kroz protokol autentifikacije i sve što može da dobije prisluškivanjem komunikacije između takvog korisnika i pristupne tačke su šifrovani podaci u paketima.



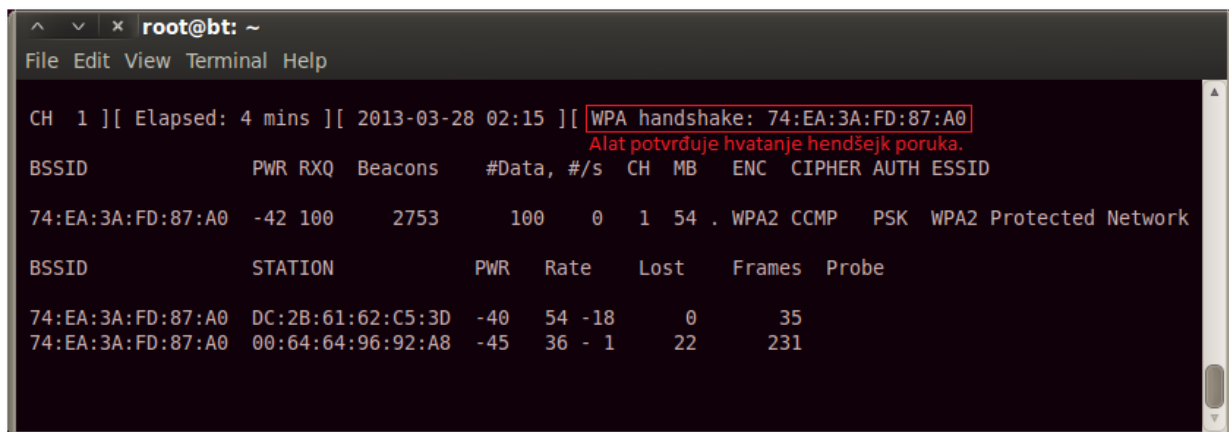
Slika 44: Prikaz rezultata prisluškivanja

Potrebno je uhvatiti pakete koji se razmenjuju na samom početku pristupanja mreži tj. pakete koji sadrže hendšejk. To se može uraditi u slučaju kada se neki novi legitimni korisnik prijavljuje na mrežu za vreme prisluškivanja saobraćaja, ili u slučaju kada se korišćenjem dostupnih alata, deautentifikuje već autentifikovani korisnik mreže i time izazove pokretanje mehanizma za ponovnu autentifikaciju. Do ovakve reakcije dolazi zbog toga što operativni sistemi legitimnih korisnika „misle“ da je jednostavno došlo do greške u komunikaciji, pa kako ne bi dovodili korisnika u neprijatnu situaciju, automatski se ponovo autentifikuju.



Slika 45: Prikaz deautentifikacije korisnika

Nakon prikazane deautentifikacije korisnika, alat korišćen za prisluškivanje saobraćaja prijavljuje hvatanje hendšejk paketa korišćenih za autentifikaciju. Sada, pošto su prikupljeni svi potrebni podaci koji se nalaze u hendšejk paketima, prelazi se na prikazivanje mogućih metoda za pronalazak lozinke korišćene za autentifikaciju korisnika⁵³.



```
root@bt: ~
File Edit View Terminal Help

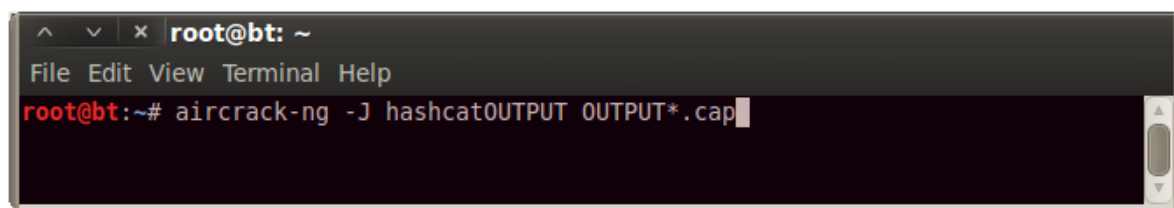
CH 1 ][ Elapsed: 4 mins ][ 2013-03-28 02:15 ][ WPA handshake: 74:EA:3A:FD:87:A0
Alat potvrđuje hvatanje hendšejk poruka.
BSSID          PWR RXQ Beacons  #Data, #/s CH MB ENC CIPHER AUTH ESSID
74:EA:3A:FD:87:A0 -42 100 2753 100 0 1 54 . WPA2 CCMP PSK WPA2 Protected Network

BSSID          STATION          PWR Rate Lost Frames Probe
74:EA:3A:FD:87:A0 DC:2B:61:62:C5:3D -40 54 -18 0 35
74:EA:3A:FD:87:A0 00:64:64:96:92:A8 -45 36 -1 22 231
```

Slika 46: Prikaz uspešnog hvatanja hendšejk poruka

4.2. Analiza prikupljenih podataka

Nakon uspešnog prikupljanja paketa koji opisuju eapol hendšejk poruke, prelazi se na izvoz sadržaja poruka. Biće prikazan potpun oblik podataka, i podaci kao takvi će biti korišćeni od strane alata za pronalaženje lozinke u svrhe izračunavanja mogućih lozinki.



```
root@bt: ~
File Edit View Terminal Help

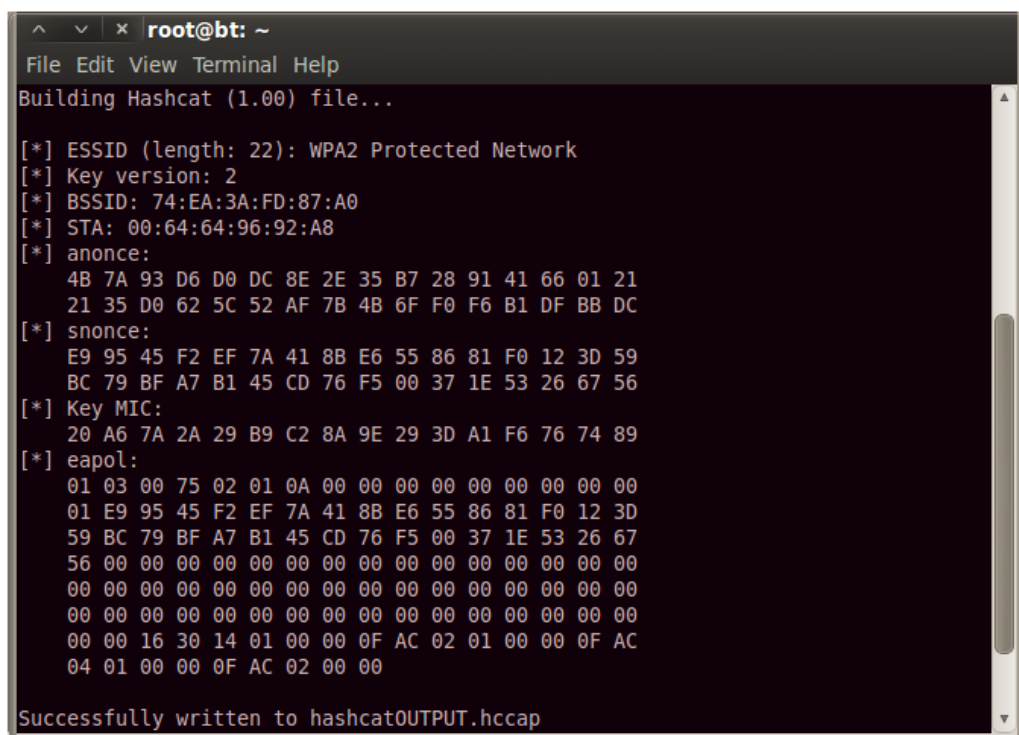
root@bt:~# aircrack-ng -J hashcatOUTPUT OUTPUT*.cap
```

Slika 47: Prikaz komande za izvoz podataka iz poruka hendšejka

Svi podaci se nalaze u četiri prikupljene hendšejk poruke. Može se doći do njih analizom sadržaja kroz neki od softverskih alata kao što je vajeršark (eng. Wireshark), ili se

⁵³ BackTrack 5 Wireless Penetration Testing Beginner's Guide; Vivek Ramachandran (Sep 9, 2011)

može upotrebiti alat koji se nalazi u Backtrack Linuks distribuciji pod nazivom aircrack-ng. Aircrack-ng poseduje veliki broj funkcija od kojih je jedna izvoz podataka prikupljenih prisluškivanjem hendšejk protokola u određeni format. S obzirom da postoje gotove, besplatne i komercijalne aplikacije, koje omogućavaju iscrpnu pretragu ili pretragu po rečniku na osnovu gore navedenih podataka, aircrack-ng poseduje mogućnosti za izvoz podataka direktno u neki od formata za aplikacije tog tipa. Komercijalni softver tog tipa je EWSA (Elcomsoft Wireless Security Auditor), a besplatni softver, kojim će se kasnije testirati je hešket (eng. Hashcat). Na slici ispod je prikazan izvoz podataka za Hashcat aplikaciju korišćenjem Aircrack-ng alata⁵⁴.



```
^ v x root@bt: ~
File Edit View Terminal Help
Building Hashcat (1.00) file...

[*] ESSID (length: 22): WPA2 Protected Network
[*] Key version: 2
[*] BSSID: 74:EA:3A:FD:87:A0
[*] STA: 00:64:64:96:92:A8
[*] anonce:
4B 7A 93 D6 D0 DC 8E 2E 35 B7 28 91 41 66 01 21
21 35 D0 62 5C 52 AF 7B 4B 6F F0 F6 B1 DF BB DC
[*] snonce:
E9 95 45 F2 EF 7A 41 8B E6 55 86 81 F0 12 3D 59
BC 79 BF A7 B1 45 CD 76 F5 00 37 1E 53 26 67 56
[*] Key MIC:
20 A6 7A 2A 29 B9 C2 8A 9E 29 3D A1 F6 76 74 89
[*] eapol:
01 03 00 75 02 01 0A 00 00 00 00 00 00 00 00 00
01 E9 95 45 F2 EF 7A 41 8B E6 55 86 81 F0 12 3D
59 BC 79 BF A7 B1 45 CD 76 F5 00 37 1E 53 26 67
56 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 16 30 14 01 00 00 0F AC 02 01 00 00 0F AC
04 01 00 00 0F AC 02 00 00

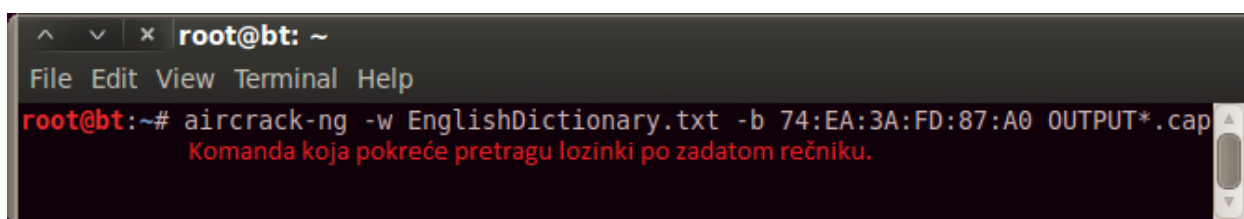
Successfully written to hashcatOUTPUT.hccap
```

Slika 48: Prikaz podataka sadržanih u hendšejk porukama

⁵⁴ Hacking Exposed 7: Network Security Secrets & Solutions, Seventh Edition; Stuart McClure, Joel Scambray and George Kurtz (Jul 11, 2012)

4.2.1. WPA i WPA2 napad korišćenjem rečnika (CPU)

Sledi prikaz napada korišćenjem rečnika. Aircrack-ng, alat korišćen za izvoz podatka iz prikupljenih paketa poseduje opciju za isprobavanje lozinki korišćenjem zadatog rečnika. Lozinka za pristup mreži korišćenoj za napad je naravno u radu unapred poznata i predstavlja standardnu reč od osam karaktera koja se može naći u rečniku. Zapravo to i jeste reč engleskog jezika „standard“. Rečnik koji će biti korišćen za testiranje i prikazivanje vremena potrebnog za pronalazak lozinke sadrži 270.000 reči često korišćenih reči engleskog jezika.



Slika 49: Prikaz komande za pokretanje pretrage

Po pokretanju alata izraženo je proteklo vreme, koliko lozinki je isprobano, koja je brzina pretrage u sekundi i sva izračunavanja koja alat mora da uradi kako bi proverio da li je pretpostavljena lozinka zaista prava lozinka. Na neki način, odmah se može zaključiti da je ovo zaista brz alat. Mora se napomenuti da ovaj alat ne koristi ubrzanje grafičkih procesora već isključivo snagu centralnog procesora računara⁵⁵.

⁵⁵ Ethical Hacking and Countermeasures: Secure Network Infrastructures (Ethical Hacking and Countermeasures; EC-Council (Sep 24, 2009)

```

root@bt: ~
File Edit View Terminal Help

Aircrack-ng 1.1 r2178

      Broj isprobanih lozinki do sada
[00:01:18] 101772 keys tested [1330.30 k/s]
      Protoklo vreme      Broj isprobanih lozinki u sekundi.

      Current passphrase: misfitting

Master Key : FE 04 33 D9 4A AA AA AC C7 1E 6C 11 41 A0 60 94
            0E 33 D3 81 79 80 00 29 6F 28 91 79 79 CB D8 21

Transient Key : 7B 82 46 5D E6 51 62 00 50 9D 06 F6 7A 48 64 61
               EA DE 4F 40 16 30 21 FD 82 EE FF 72 AE 90 36 C6
               11 80 C1 31 4B 47 A5 EF BF 09 B2 2C BD ED A4 C7
               38 54 8B 53 D0 01 9F 82 43 AB 48 41 F6 CB 3F B0

EAPOL HMAC : B8 16 94 4C B9 D2 EC DB FD 0C 2E 34 E2 AC 7C 71
  
```

Slika 50: Prikaz procesa pronalaženja lozinki

Reč „standard“ se nalazila na 160.992. mestu u rečniku često korišćenih reči engleskog jezika. Za pronalazak takve, jednostavne lozinke, potrebno je samo 1 minut i 18 sekundi na prenosivom računaru. U narednoj tabeli (Tabela 5) prikazana su prosečna vremena pretrage po rečniku na prenosivom računaru navedenih karakteristika.

Tip rečnika:	Broj reči u rečniku:	Brzina pretrage kroz ceo rečnik:
Često korišćene reči engleskog jezika.	267.795	2 minuta i 30 sekundi
Kompletan rečnik engleskog jezika.	603.593	5 minuta i 28 sekundi
Poseban rečnik sastavljen u svrhe pronalaženja lozinki.	1.213.139.460	Približno 8 dana.

Tabela 5: Prosečna vremena pretrage po rečniku

Iz prikazanih rezultata, vidi se da je pretraga po rečniku bilo koje vrsta izvodljiva čak i na običnom računaru ali je takođe veoma važno još jednom napomenuti da je pretraga po bilo kakvom rečniku ne garantuje uspeh. Usvojena je ideja, koja je navedena u teorijskim osnovama rada, da je verovatnoća pojavljivanja lozinke u bilo kom rečniku $p(\text{rečnik}) = \frac{1}{4}$. Cilj svakog napadača, jeste da obezbedimo siguran pronalazak lozinke po svaku cenu. U narednom segmentu biće predstavljeno ubrzanje pretrage po rečniku korišćenjem grafičkih procesora.

4.2.2. WPA i WPA2 napad korišćenjem rečnika (GPU)

Grafički procesori su po mnogo čemu različiti od centralnih procesora računara. Ono što ih čini neuporedivo boljim za računanja kakva su potrebna radi pronalaženja lozinke jeste veći broj jezgara koja rade paralelno. Računanje potrebno za proveru da li je jedna pretpostavljena lozinka (npr. jedna reč iz rečnika), zaista prava lozinka jeste zahtevno, ali svaki savremeni procesor neće imati nikakvih problema sa izračunavanjem svih potrebnih heš funkcija. Znajući to ubrzanje se može dobiti jedino povećavanjem mogućih računanja u isto vreme – povećavanjem broja jezgara procesora koji radi računanje⁵⁶.

⁵⁶ Effective Programming: More Than Writing Code; Jeff Atwood (Jul 4, 2012)

```

root@bt: ~
File Edit View Terminal Help
s
Session.Name...: cudaHashcat-plus
Status.....: Running
Input.Mode....: File (EnglishDictionary2.txt)
Hash.Target...: WPA2 Protected Network (00:64:64:96:92:a8 <-> 74:ea:3a:fd:87:a0)
Hash.Type.....: WPA/WPA2
Time.Started...: Sat Apr 6 19:08:27 2013 (2 mins, 7 secs)
Time.Estimated.: Sat Apr 6 19:14:50 2013 (4 mins, 15 secs)
Speed.GPU.#1...: 1839/s Broj isprobanih lozinki po sekundi.
Recovered.....: 0/1 (0.00%) Digests, 0/1 (0.00%) Salts
Progress.....: 348370/603508 (57.72%)
Rejected.....: 114898/348370 (32.98%)
HWMon.GPU.#1...: -1% Util, 86c Temp, -1% Fan

[s]tatus [p]ause [r]esume [b]ypass [q]uit =>

```

Slika 51: Prikaz hešket alata u toku rada

Testirana je pretraga po već predloženim rečnicima, samo što će se ovog puta koristiti snaga grafičkog procesora i alat pod nazivom hešket (eng. Hashcat) koji podržava grafičke procesore.

Tip rečnika:	Broj reči u rečniku:	Brzina pretrage kroz ceo rečnik:
Često korišćene reči engleskog jezika.	267.795	1 minut i 23 sekunde
Kompletan rečnik engleskog jezika.	603.593	3 minuta i 55 sekundi
Poseban rečnik sastavljen u svrhe pronalaženja lozinki.	1.213.139.460	Približno 5 dana.

Tabela 6: Pretraga po rečniku korišćenjem grafičkih procesora

Iz priloženih rezultata se može videti da se postiže minimalno ubrzanje koristeći grafički procesor navedenog prenosivog računara. Mora se navesti da grafički procesor koji je ovde korišćen ima performanse daleko ispod performansi grafičkih procesora današnjih desktop računara. Radi dodatnog poređenja, biće prikazani rezultati korišćenjem jedne standardne desktop grafičke kartice.

	NVIDIA GeForce 320M	NVIDIA GeForce 9800GT
Clock	950Mhz	1500Mhz
Broj jezgara	48 (6)	112 (14)
Broj lozinki u sekundi	1918	6411

Tabela 7: Poređenje laptop i desktop grafičkih procesora

Ovim rezultatom se vidi da prelaskom na grafičke procesore dolazi do velikog ubrzanja pretrage. Problem koji ostaje jeste taj što pretraga po bilo kakvom rečniku ne garantuje pronalazak lozinke. Jedini način za siguran pronalazak lozinke jeste korišćenje pretrage svih mogućih kombinacija – „Brute-Force“ napad. Sa druge strane, broj mogućih kombinacija za lozinku od 8 karaktera jeste neuporedivo veći čak i u odnosu na posebne rečnike koji sadrže milione unosa. U sledećim delovima rada obrađuju se isključivo pretraga svih mogućih kombinacija i predlozi za maksimalno ubrzanje takve pretrage.

4.2.3. WPA i WPA2 brute-force napad

Zaključuje se da su brzine pretrage po rečniku iste kao i brzine pretrage po svim mogućim kombinacijama što je svakako i logično. Dakle najveći problem predstavlja postojanje velikog broja kombinacija koje treba isprobati kako bi napadač bio unapred siguran da će pronaći lozinku. Koristeći grafički procesor referentnog računara, optimistički će biti započeto pronalaženje lozinke sastavljene od 8 karaktera pretragom svih mogućih kombinacija⁵⁷.

⁵⁷ Professional Penetration Testing, Second Edition: Volume 1: Creating and Learning in a Hacking Lab; Thomas Wilhelm (Aug 15, 2013)

```

root@bt: ~
File Edit View Terminal Help
Session.Name...: cudaHashcat-plus
Status.....: Running
Input.Mode.....: Mask (?1?1?1?1?1?1?1?1) Kombinacija od 8 karaktera.
Hash.Target....: WPA2 Protected Network (00:64:64:96:92:a8 <-> 74:ea:3a:fd:87:a0)
Hash.Type.....: WPA/WPA2
Time.Started...: Sun Apr  7 21:41:15 2013 (57 secs)
Time.Estimated.: Tue Dec 25 15:58:43 1990 (-23 years, 261 days)
Speed.GPU.#1...: 1904/s Vreme pretrage
Recovered.....: 0/1 (0.00%) Digests, 0/1 (0.00%) Salts
Progress.....: 98304/218340105584896 (0.00%) Broj isprobanih i preostalih kombinacija.
Rejected.....: 0/98304 (0.00%)
HWMon.GPU.#1...: -1% Util, 66c Temp, -1% Fan
[s]tatus [p]ause [r]esume [b]ypass [q]uit =>

```

Slika 52: Prikaz brute-force napada

Dolazi se do rezultata koji govori da je potrebno više od 20 godina za isprobavanje svih kombinacija. Zaključak je da ovakav način pretrage nije prihvatljiv za lozinke duže od 5 karaktera, za koje je potrebno „izvodljivih“ 10 dana i 14 sati. Još jednom, radi dodatnog poređenja, na desktop računaru sa NVIDIA GeForce 9800GT grafičkom karticom za lozinku sastavljenu od 5 karaktera potreban je 1 dan i 18 sati dok već za lozinku od 16 karaktera 109 dana i 13 sati.

4.2.4. WPA i WPA2 CPU klaster

Iz priloženih rezultata proizilazi zaključak da je moguće razviti sistem koji obezbeđuje siguran pronalazak lozinke do određenog broja karaktera. Za razvoj takvog sistema su potrebna velika sredstva – nekoliko računara sa po više najsavremenijih grafičkih kartica, adaptacija softvera, adaptacija hlađenja takvih računara, napajanja. Na osnovu tih činjenica se dolazi do ideje da je možda moguće iskoristiti veliki broj umreženih računara. Namenska kupovina ovakve opreme za potrebe istraživanja bi bila neosnovana, pa su za testove korišćene postojeće elektronske laboratorij u kojima se odvija redovna nastava na Univerzitetu.

U svrhe testiranja takvog okruženja, razvijen je novi softver u JAVI koji može koristiti i do nekoliko hiljada centralnih procesora umreženih računara. Samim tim što je softver razvijen u JAVI, veoma je fleksibilan i može se koristiti na svakom operativnom

sistemu, dakle nema potrebe za modifikacijama na već umreženim računarima. Framework koji omogućava korišćenje umreženih računara je JPPF Framework (Cohen, 2005).

Dobijeni rezultati nakon navedenog testa nisu ispunili očekivanja. Dva glavna faktora koja utiču na lošije rezultate jesu to što je softver razvijen u JAVI – koristi java virtuelnu mašinu koja interpretira kod za određenu platformu, dakle sporiji rad od jezika poput C-a koji se direktno kompajlira u kod razumljiv procesoru računara. Usporenje se takođe javilo jer se koristi centralni procesori računara koji su, kao što je prethodno navedeno, daleko sporiji od grafičkih procesora.

	C-rešenje (hashcat CPU)	Sopstveno JAVA-rešenje
Opterećenje procesora:	100% (sva raspoloživa jezgra)	100% (sva raspoloživa jezgra)
Ubrzanje sa dodavanjem računara:	Nije moguće	Linearno
Broj računanja u sekundi na jednom računaru:	1200	130

Tabela 8: Poređenje Hashcat i sopstvenog rešenja

U tabeli se vidi da prikazani rezultati dovode do činjenice da je razvijeno rešenje u poređenju sa rešenjem napisanim u C jeziku 9 puta sporije. Ako se zna da je razvijeno rešenje moguće distribuirati na više umreženih računara, i da je porast brzine računanja potpuno linearna sa porastom broja korišćenih računara (procesorskih jezgara), onda se zna da je potrebno 9 umreženih računara za postizanje performansi postojećeg rešenja. Dakle, sa 9 računara se postiže dosadašnji maksimum brzine kod korišćenja centralnog procesora. Ako se u obzir uzme mogućnost korišćenja prostorije koja sadrži 125 umreženih računara, dostiže se brzina pretrage od 16.250 lozinki u sekundi što je veliki napredak kada je u pitanju pretraga korišćenjem centralnog procesora računara.

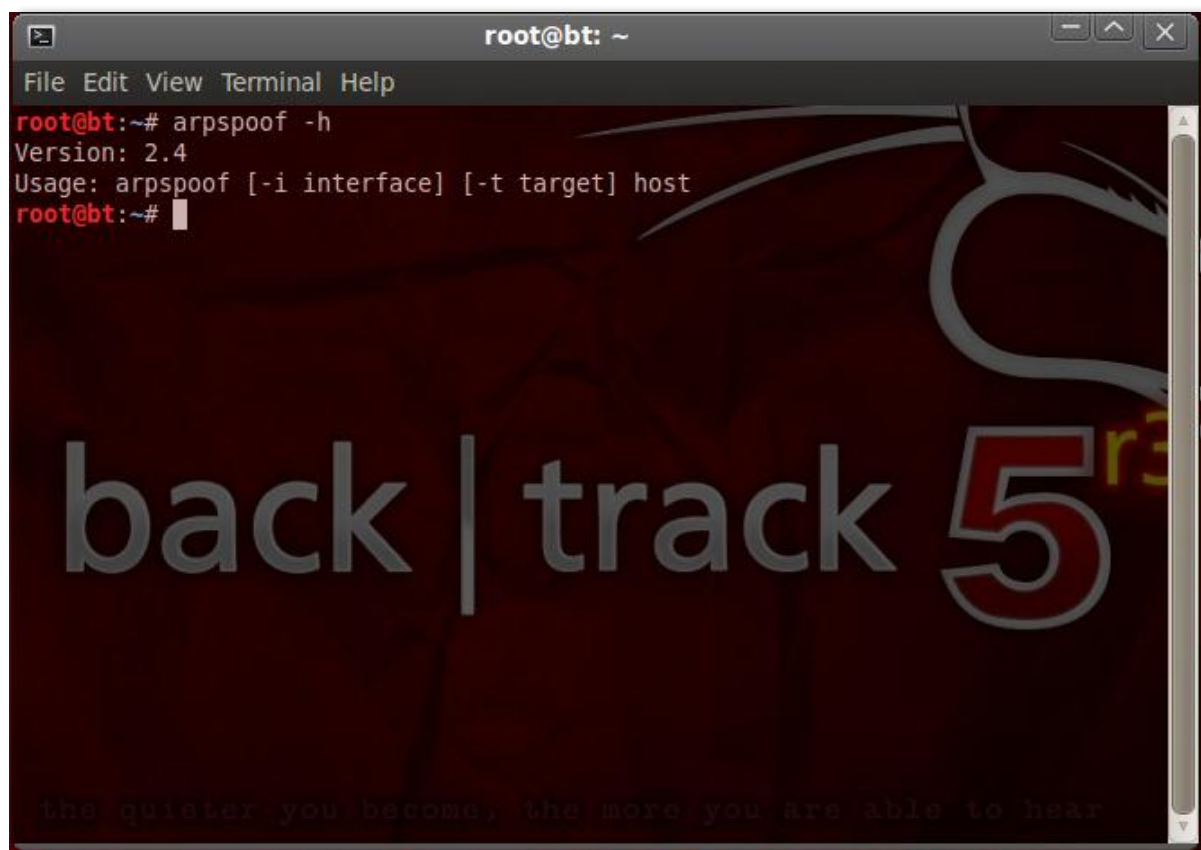
Problem koji ostaje kada je razvijeno rešenje u pitanju je taj što pretraga korišćenjem grafičkog procesora običnog desktop računara dostiže 6411 lozinki po sekundi. Korišćenjem grafičkih kartica umreženih računara bi se napravio veliki pomak u brzini pretrage. Postoji predloženo rešenje koje se bazira na hešket alatu (hashcat - advanced password recovery, n.d.), ali zahteva veoma striktne uslove za rad, određeni operativni sistem, određenu količinu

ram memorije, određen tip mreže – dakle razvoj posebnog sistema kakav je opisan na početku ovog poglavlja.

Predmet daljeg rada će se bazirati isključivo na grafičkim procesorima. Obaviće se pokušaj korišćenje okruženja koje je na raspolaganju, adaptiranje kako bi bilo u skladu sa predloženim rešenjem i provera rezultata takvog okruženja.

4.3. Napad po principu čovek u sredini

Napadi po principu čovek u sredini poznati su od samog nastanka računarskih mreža. Kao i same računarske mreže i napadi po principu čovek u sredini su evoluirali sa razvojem sistema zaštite. Prethodno opisani sistem zaštite WPA2, SSL, TLS i HTTPS standardi više nisu dovoljni za potpunu zaštitu od zlonamernih korisnika. U nastavku je predstavljen mogući scenarijo napada na korisnika u računarskoj mreži, čiji se saobraćaj štiti svim prethodno pomenutim metodama zaštite.



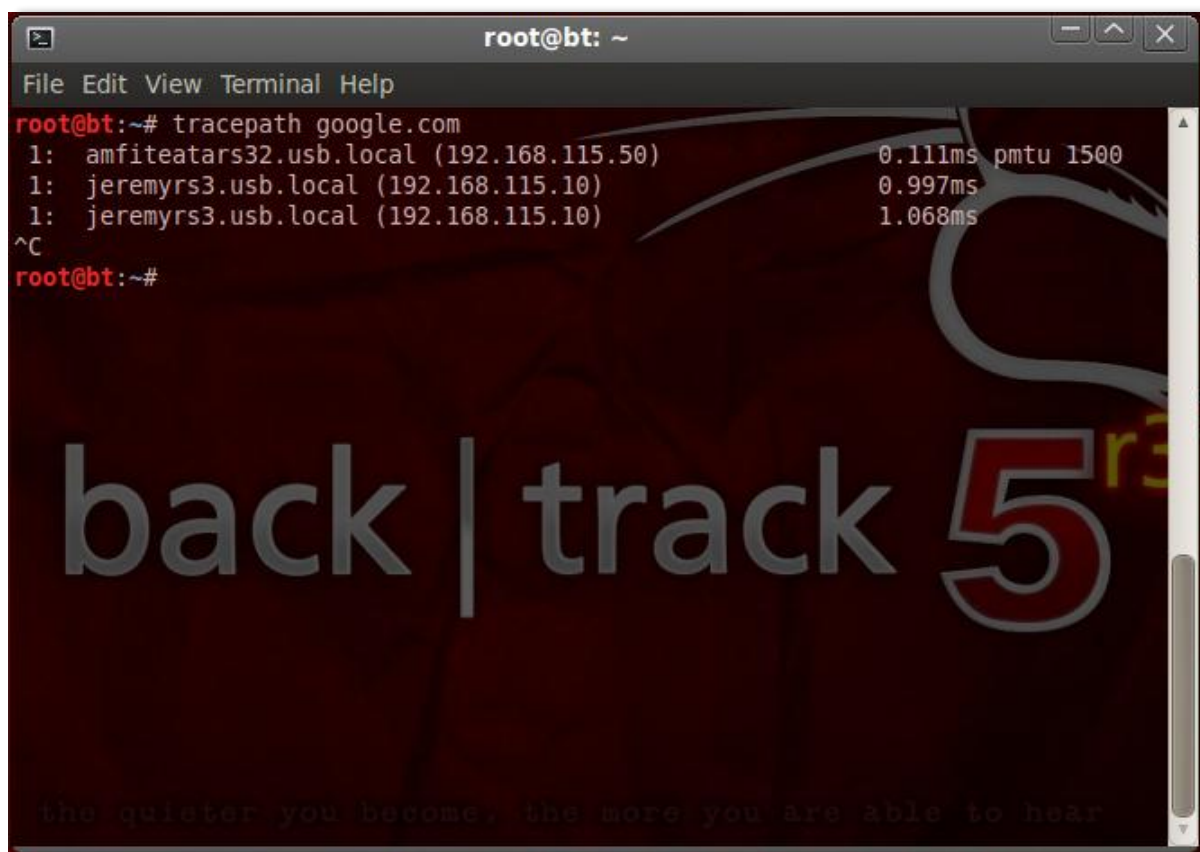
Slika 53: Pokretanje ARPspooft-a i pregled opcionih parametara

Preusmeravanje mrežnog saobraćaja je prethodnica svakog napada, realizacija je najčešće realizovana preusmeravanjem mrežnog saobraćaja. Ilustracija (Slika 53) predstavlja početak napada komandom arpspoof koja se koristi za napad na ARP (Address Resolution Protocol).⁵⁸ ARP spoofing je tehnika u kojoj napadač šalje lažne ARP poruke na u lokalnu mrežu. Generalno, cilj je da se povežu napadačeva MAC adresa sa IP adresom drugog domaćina (npr. default gateway), izazivajući da sav saobraćaj namenjen za tu IP adresu ubuduće bude poslat napadaču.

ARP spoofing može dozvoliti napadaču da presretne okvire podataka na mreži, izmeni saobraćaj, ili potpuno zaustavi saobraćaj. Često se koristi kao napad kojim se mreža otvara za druge napade, kao što su uskraćivanje usluga, čovek u sredini, ili napadi preuzimanja sesije.

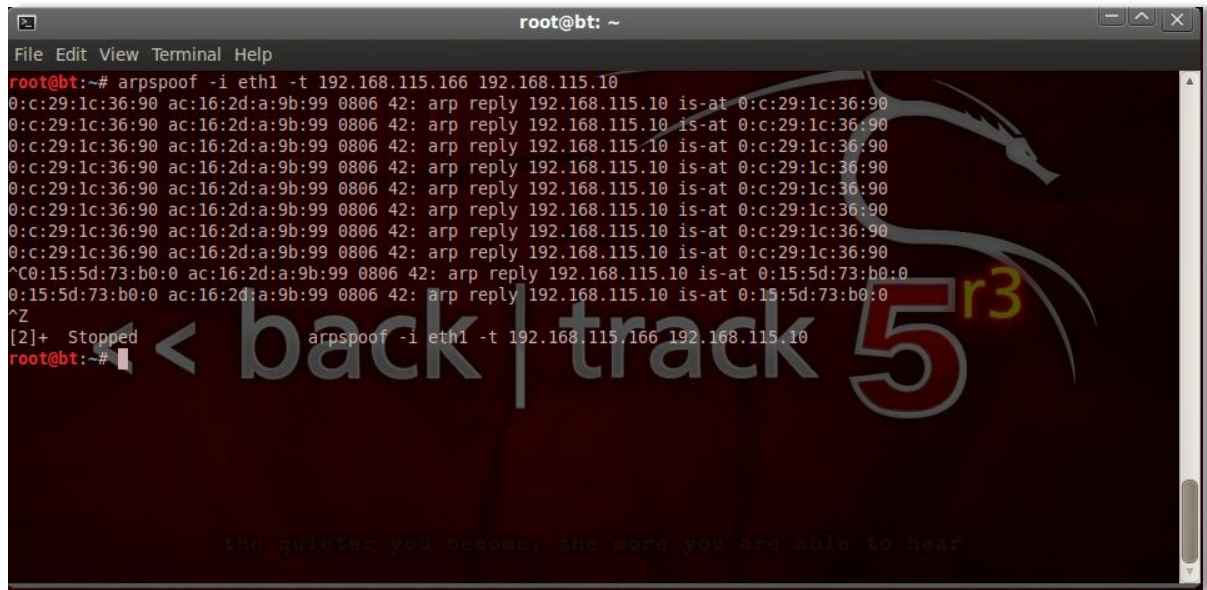
Napad se može koristiti samo na mrežama koje koriste ARP i ograničen je na lokalni segment računarske mreže.

⁵⁸ ARP (engl. Address Resolution Protocol) je protokol trećeg sloja OSI referentnog modela, definiše metode nalaženja fizičke adrese korisnika na osnovu poznate mrežne adrese. S obzirom na najširu upotrebu internet protokola kao protokola adresiranja sloja mreže i etherneteta kao protokola najniža dva sloja, ARP ima najširu primenu pri mapiranju MAC adrese sa IP adresom. Opisan je u RFC dokumentu pod rednim brojem 826.



Slika 54: Utvrđivanje putanje kretanje korisnika kroz lokalnu mrežu ka udaljenoj lokaciji

ARP je protokol koji ne beleži stanje. Mrežni domaćini će automatski keširati sve ARP odgovore koje primaju, bez obzira na to da li su ih tražili. Čak će ARP zapisi koji nisu istekli biti prepisani kada se novi ARP odgovor paketi prime. Ne postoji metod u ARP protokol kojim domaćin može da potvrdi i utvrdi od koga potiče paket. Ovo ponašanje je ranjivost koja omogućava da se dogodi ARP spoofing i napad na računarsku mrežu.

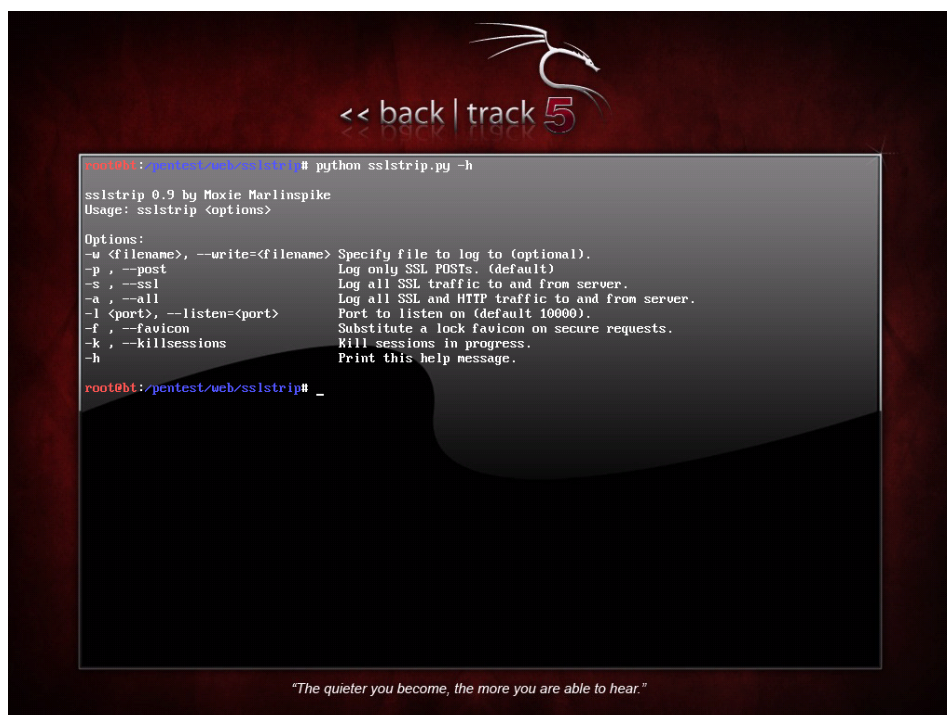


Slika 55: Preusmeravanje saobraćaja korisnika ka računaru napadača metodom arpspoof

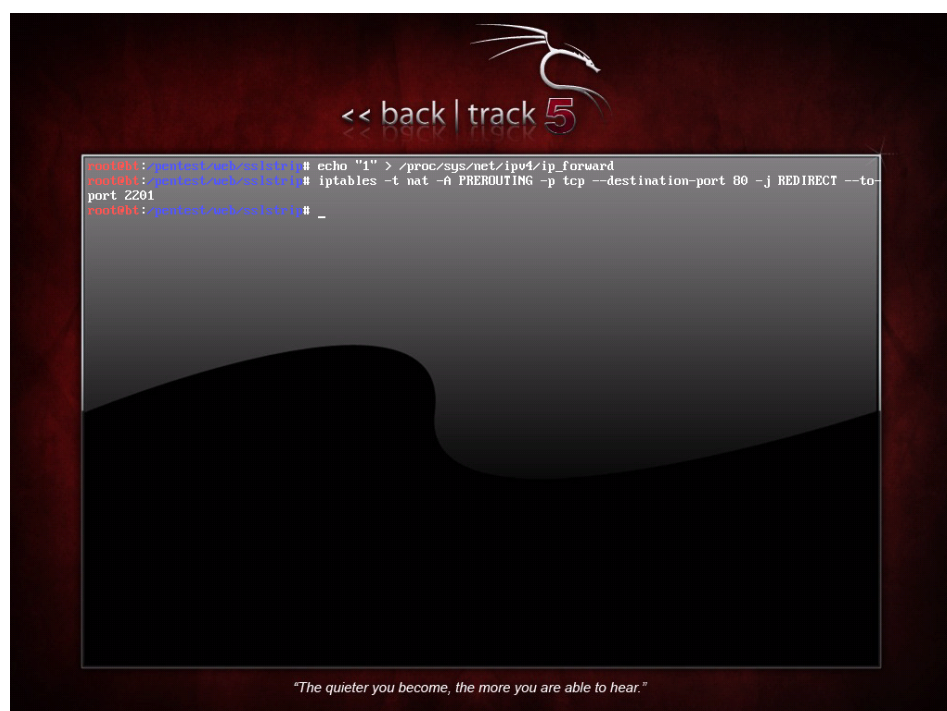
Ilustracija (Slika 55) predstavlja preusmeravanje svog saobraćaja od preporučenog mrežnog izlaza ka napadaču zahvaljujući arpspoof alatu i propustu u ARP zapisu.

Nakon što je saobraćaj preusmeren ka napadaču, može se početi sa uticajima na korisnikove podatke. Zavisno od toga šta napadač želi koristiti neke od do sada obrađenih alata i tehnika. Pri napadu po principu čovek u sredini, napadač najčešće ne želi da korisnik bude svestan napada i ne želi da korisnika spreči u redovnim aktivnostima i korišćenju sopstvenih resursa. Resursa koji su u današnje vreme najčešće štićeni prethodno opisanim SSL/TLS tipovima zaštite. U ovom slučaju najčešće korišćena metoda i alat za obilaženja pomenute zaštite je korišćenjem alata SSLStrip koji ima ulogu da uspostavi sigurnu HTTPS vezu sa validnim serverom ali i da uspostavi regularnu HTTP vezu sa korisnikom koju će napadač dalje moći da prati (Slika 56)⁵⁹.

⁵⁹ Marlinspike je uveo koncept SSL strip, čovek-u-sredini napada u kojima bi napadač mogao da spreči veb pretraživač sa nadogradnjom na SSL veze na suptilan način koji će verovatno proći nezapaženo od strane korisnika. On je takođe objavio alata, sslstrip, koji automatski obavlja ove vrste napada na SSL.



Slika 56: Alat SSLstrip, pregled mogućih opcija

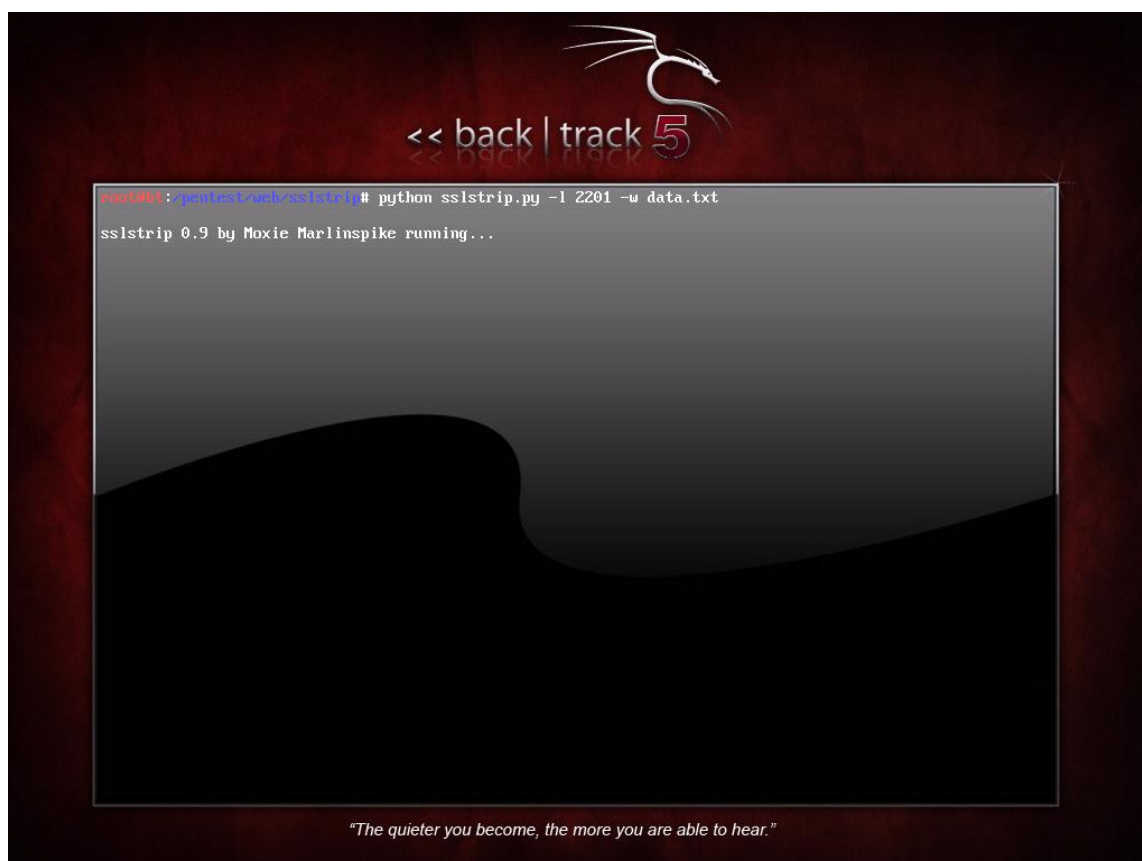


Slika 57: Usmeravanje saobraćaja klijenta nakon pregledavanja od strane napadača

Kroz implementaciju ovakvog napada napadač želi da regularni korisnici računarske mreže nastave sa svojim redovnim aktivnostima bez sumnje da je mreža pod napadom. Kako bi ovo ostvario potrebno je da napadač omogući korisnicima korišćenje resursa računarske mreže preko preporučenog mrežnog izlaza, tj. da sav saobraćaj i informacije koji korisnici

upute lažnom mrežnom izlazu (napadaču) u stvari budu prosleđene validnom mrežnom izlazu koji će obraditi zahteve. Komanda koja napadaču ovo omogućava je predstavljena u okviru ilustracije (Slika 57).

Napadači najčešće ne izvode napade u realnom vremenu, već prikupljaju podatke koje će kasnije obrađivati i koristiti. Ilustracija (Slika 58) predstavlja simulaciju konfiguracije alata sslstrip tako da sluša svu komunikaciju koja se odvija između napadača i validnog mrežnog izlaza i beleži je u okviru dokumenta koji će kasnije biti obrađen.

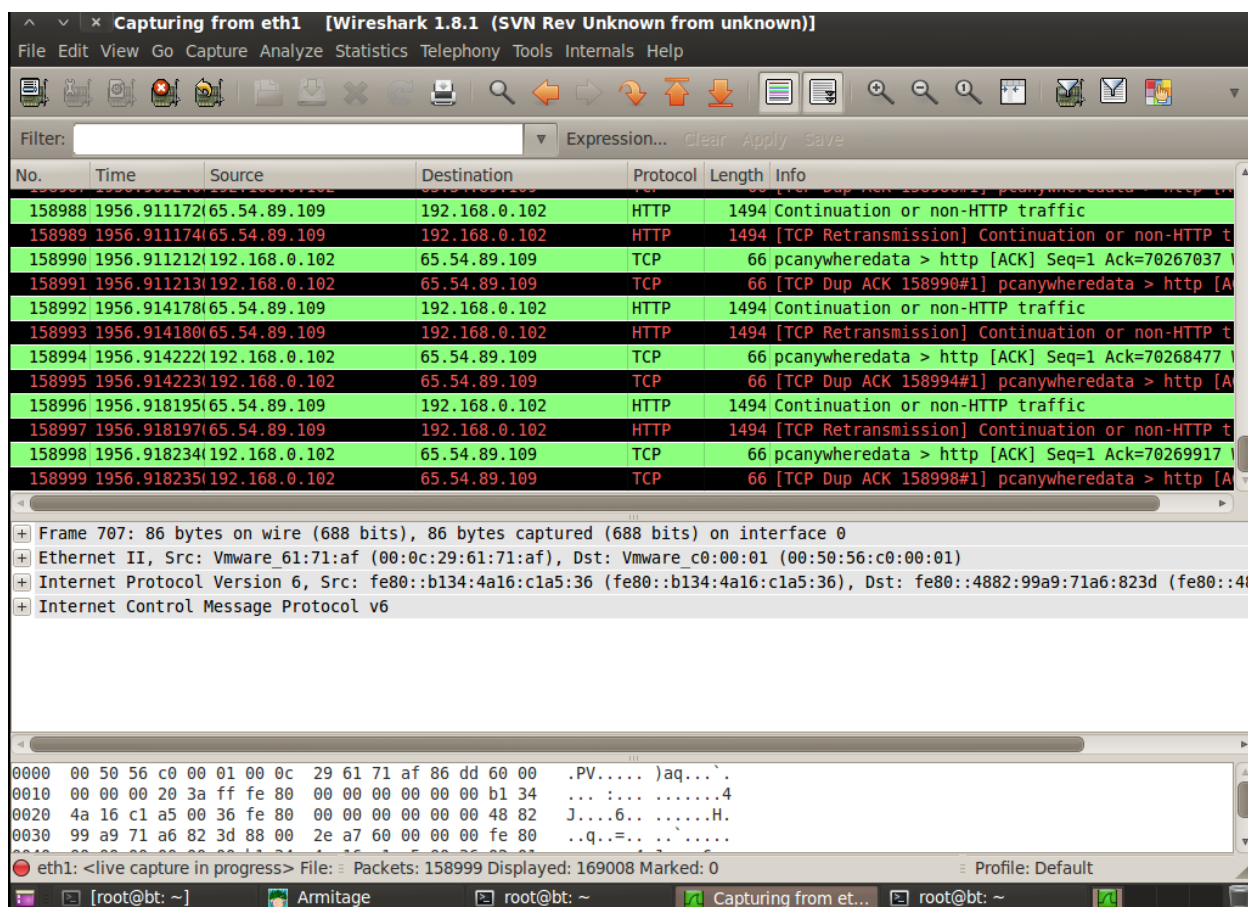


Slika 58: Konfigurisan alat SSL strip tako da beleži aktivnosti korisnika

Za analizu prikupljenih podataka može se koristiti dosta alata od kojih se svakako među najpopularnije ubraja alat Wireshark prikazan na narednoj ilustraciji (Slika 59). Wireshark je softverski alat koji „razume“ strukturu različitih mrežnih protokola. S obzirom da je Wireshark open source alat, relativno je jednostavno implementirati programske dodatke za nove protokole. Podaci se mogu „hvatati“ direktno s aktivne mrežne veze ili se mogu učitati iz datoteke u kojoj su prethodno snimljeni već „uhvaćeni“ paketi. „Uhvaćeni“ podaci mogu biti prikazani preko grafičkog korisničkog interfejsa ili preko terminala (komandne linije) kod korišćenja TSharka.

Wireshark ima, uz konzolu za upis naredbi, grafičko korisničko okruženje koje korisnicima umnogome olakšava rad i snalaženje u alatu. Wireshark podržava sve važnije mrežne protokole i ima mogućnost nadogradnje za nove protokole tako da se do sada broj

podržanih protokola popeo na više od stotinu. Wireshark na vrlo jednostavan i intuitivan način omogućuje korisnicima povezivanje i uvoz i izvoz podataka na druge i s drugih sličnih mrežnih alata, kao što su dSniff, tcpdump i u primeru napada korišćeni sslstrip.



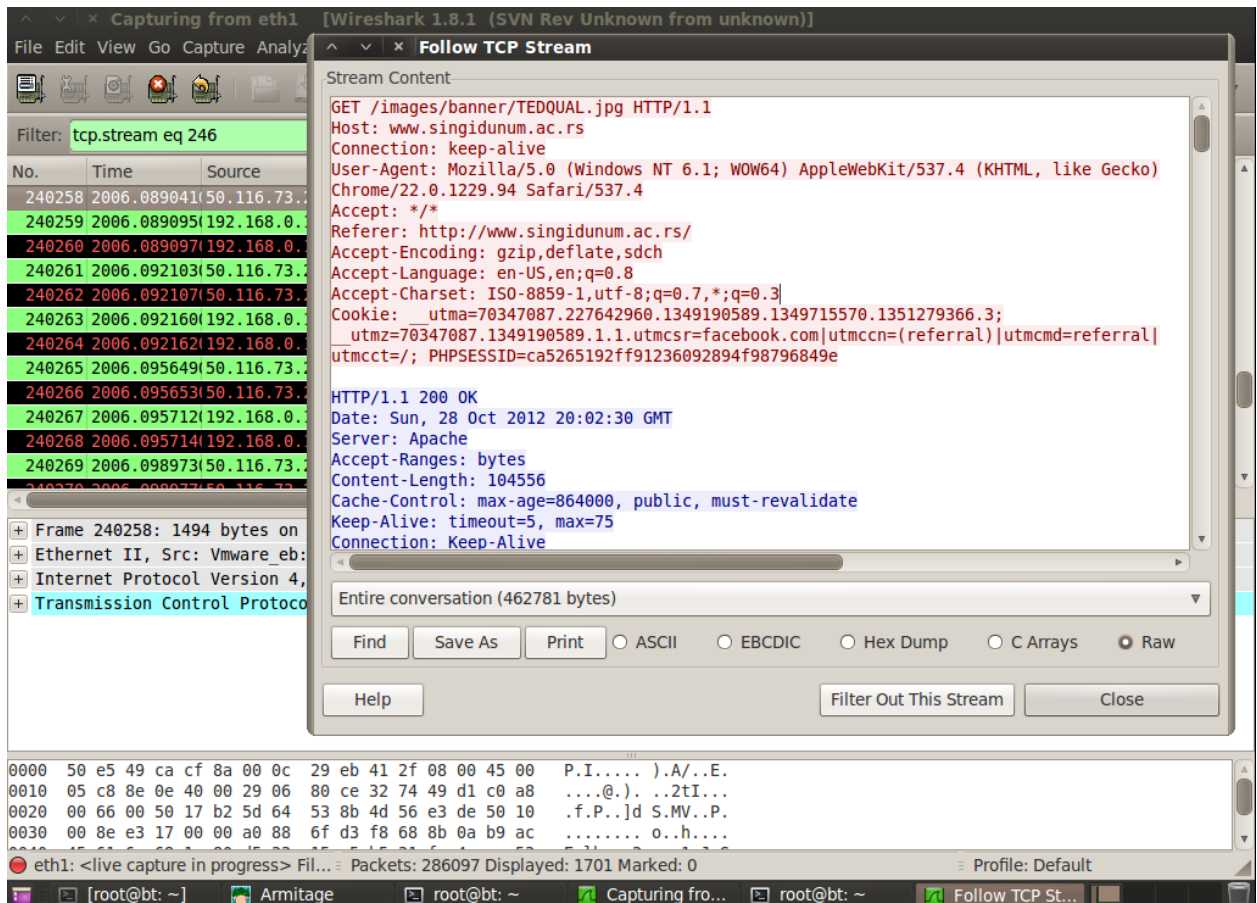
Slika 59: Prikaz alata Wireshark koji se koristi za analizu prikupljenih podataka korisnika

Osim same analize prikupljenih podataka (o kretanju i aktivnostima korisnika) koja je prikazana na prethodnoj ilustraciji (Slika 59), napadači često za cilj imaju i prikupljanje korisničkih podataka za autentifikaciju. Analizom prikupljenih paketa (Slika 60) napadač može doći do podataka prema kojima su korisnici najosetljiviji. Podacima za autentifikaciju (korisničko ime i lozinka) korisnici često štite i svoje najprivatnije i najdragoceneje podatke.

U teorijskoj osnovi rada već je pomenuto da idealni način autentifikacije podrazumeva tri segmenta:

- Nešto što korisnik zna (korisničko ime i lozinka)
- Nešto što korisnik ima (telefon, sms, token, smart card)
- Nešto što korisnik jeste (otisak prsta, glas, iris)

Ovakvi napadi i predstavljene ilustracije upravo i ukazuju da najčešće korišćena zaštita koja podrazumeva samo segment – nešto što korisnik zna, a ostatak bezbednostnih pitanja prepušta sigurnosnim protokolima računarskih mreža jednostavno nije dovoljna u velikom broju primena. U nastavku rada ponuđeno je sopstveno rešenje implementacije i dopune trenutnih rešenja segmentom – nešto što korisnik ima. Takođe predlog rešenja je implementiran dvokanalnom autentifikacijom.



Slika 60: Analiza saobraćaja, kretanja i podataka korisnika uprkos SSL zaštiti

4.4. Višekanalna autentifikacija - realizacija

U prethodnim delovima rada predstavljena je potreba za rešenjem koje će povećati bezbednost samog korisnika u računarskoj mreži i povećati poverenje prema sistemima koja ista pruža. Implementacija ovakvog rešenja moguća je bez obzira na arhitekturu i topologiju koja je primenjena za realizaciju računarske mreže⁶⁰. Praktična realizacija sistema za višekanalnu autentifikaciju je rešena korišćenjem tehnologija otvorenog koda. Za čuvanje podataka o korisniku, tokenima i prethodnim prijavama na sistem je korišćena MySQL baza podataka. Serverski deo sistema je pisan u PHP skripting jeziku i čine ga portal za demonstraciju koji se sastoji iz jedne stranice sa formularom za prijavu i stranice koja predstavlja korisnički panel, kao i sistem za generisanje tokena.

4.4.1. Struktura baze podataka

Za potrebe realizacije sistema za demonstraciju višestruke autentifikacije, podaci su u bazi podataka raspoređeni u tri tabele i to:

- users
- tokens
- logins

4.4.1.1. Tabela users

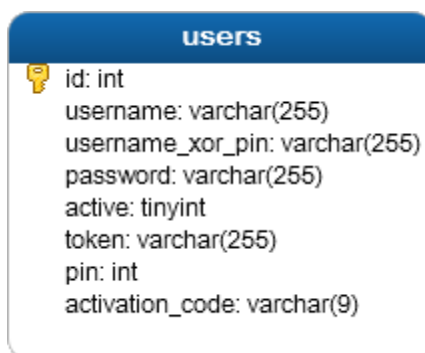
Tabela users pored jedinstvenog id broja korisnika (id), čuva korisničko ime u tekstualnom obliku (username), kao i enkodirano (username_xor_pin) ključem (pin) koji predstavlja PIN kôd kojim korisnik otključava mobilnu aplikaciju. Pored navedenih polja, u tabeli se čuva heš vrednost osnovnog dela lozinke korisnika (password), sha512 heš vrednost stringa sastavljenog od lozinke i najnovijeg generisanog tokena (token), kao i flag polja kojim je korisnički nalog obeležen kao aktivan ili neaktivan (active) polje.

⁶⁰ The New Digital Age: Reshaping the Future of People, Nations and Business; Eric Schmidt and Jared Cohen (Apr 23, 2013)

SQL za formiranje tabele users:

```
CREATE TABLE `users` (  
  `id` int(7) NOT NULL AUTO_INCREMENT,  
  `username` varchar(255) COLLATE utf8_unicode_ci NOT NULL,  
  `username_xor_pin` varchar(255) COLLATE utf8_unicode_ci NOT  
NULL,  
  `password` varchar(255) CHARACTER SET ascii NOT NULL,  
  `active` tinyint(1) NOT NULL DEFAULT '1',  
  `token` varchar(255) CHARACTER SET ascii DEFAULT NULL,  
  `pin` int(6) NOT NULL,  
  `activation_code` varchar(9) COLLATE utf8_unicode_ci DEFAULT  
NULL,  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB AUTO_INCREMENT=3 DEFAULT  
CHARSET=utf8 COLLATE=utf8_unicode_ci;
```

(19)



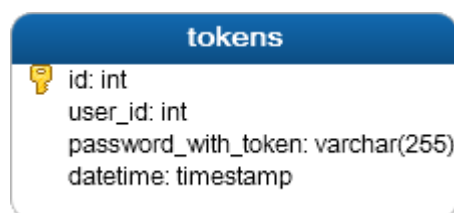
Slika 61: Grafički prikaz tabele users

4.4.1.2. [Tabela tokens](#)

Tabela tokens čuva jedinstveni id broj tokena (id), id broj korisnika za kojeg je token generisan (user_id), sha512 heš vrednost stringa sastavljenog od lozinke i generisanog tokena (password_with_token), kao i datum i vreme generisanja tokena.

SQL za formiranje tabele tokens:

```
CREATE TABLE `tokens` (  
  `id` int(15) NOT NULL AUTO_INCREMENT,  
  `user_id` int(7) NOT NULL,  
  `password_with_token` varchar(255) CHARACTER SET ascii NOT  
  NULL,  
  `datetime` timestamp NOT NULL DEFAULT  
  CURRENT_TIMESTAMP, (20)  
  PRIMARY KEY (`id`),  
  KEY `tokens_user_id_users_id` (`user_id`),  
  CONSTRAINT `tokens_user_id_users_id` FOREIGN KEY (`user_id`)  
  REFERENCES `users` (`id`) ON UPDATE CASCADE  
) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT  
CHARSET=utf8 COLLATE=utf8_unicode_ci;
```



Slika 62: Grafički prikaz tabele tokens

4.4.1.3. Tabela logins

Tabela logins čuva informacije o svim prijavama korisnika i sastoji se od jedinstvenog id broja prijave (id), id broj korisnika koji se prijavio (user_id), datum i vreme prijave (timestamp), IP adresu korisnika (ip), identifikacioni string user-agent-a (agent), identifikacioni kolačić (cookie) i PHP ID sesije (phpsessionid).

SQL za formiranje tabele logins:

```
CREATE TABLE `logins` (  
  `id` int(15) NOT NULL AUTO_INCREMENT,  
  `user_id` int(7) NOT NULL,  
  `timestamp` timestamp NOT NULL DEFAULT  
CURRENT_TIMESTAMP,  
  `ip` varchar(31) COLLATE utf8_unicode_ci NOT NULL,  
  `agent` varchar(255) COLLATE utf8_unicode_ci NOT NULL,      (21)  
  `cookie` varchar(10) COLLATE utf8_unicode_ci NOT NULL,  
  `phpsessionid` varchar(255) COLLATE utf8_unicode_ci NOT NULL,  
  PRIMARY KEY (`id`),  
  KEY `logins_user_id_users_id` (`user_id`)  
) ENGINE=InnoDB AUTO_INCREMENT=7 DEFAULT  
CHARSET=utf8 COLLATE=utf8_unicode_ci;
```

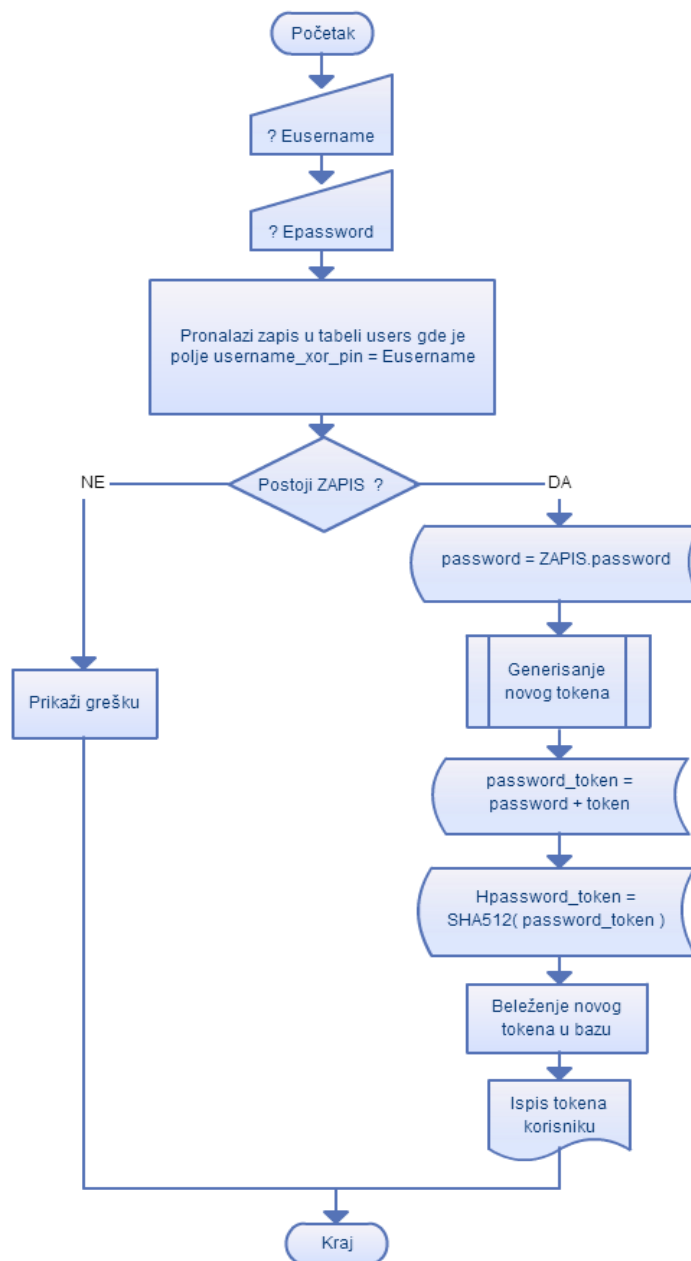


Slika 63: Grafički prikaz tabele logins

4.4.2. Način generisanja tokena

Generisanje tokena se vrši pozivom iz mobilne Android™ aplikacije koju korisnik, da bi je otključao, mora da unese PIN kôd, a zatim da u formular unese korisničko ime i lozinku.

Potvrdom unosa, podaci se kodiraju operacijom XOR karakter za karakter sa nizom karaktera koji predstavlja PIN kôd za otključavanje aplikacije.



Slika 64: Dijagram generisanja tokena

Tako dobijeni podaci (korisničko ime i lozinka) se prenose serveru kao Eusername i Epassword.

Prikazani dijagram toga predstavlja način generisanja tokena na strani servera.

Opis dijagrama:

- Od Android aplikacije se učitaju Eusername i Epassword;
- Aplikacije traži zapis u tabeli users gde username_xor_token ima vrednost polja Eusername;
- Ako zapis nije pronađen, ispisuje se poruka da je došlo do greške;
- Ako je zapis pronađen, pokreće se funkcija za generisanje tokena, kojoj se prosleđuje originalna lozinka korisnika, dobijena XOR dekodiranjem polja Epassword karakter za karakter sa PIN kodom iz zapisa;
- Lozinka i generisani token se nadovežu i izračuna se SHA512 heš vrednost dobijenog stringa;
- U bazu se zabeleži novi token;

Korisniku se nazad u aplikaciju prosledi novi token u obliku tekstualnog podatka, koji je enkodiran operacijom XOR karakter za karakter sa PIN kodom u tekstualnom zapisu, kako bi sa druge strane (na Android aplikaciji), mogao da bude uspešno dekodiran i prikazan, a da u prenosu bude zaštićen, čak i ako se desi presretanje.

PIN kôd je poznat samo korisniku u Android aplikaciji i serverskoj aplikaciji i ni u kom trenutku se ne šalje preko mreže.

4.4.3. Aktivacija naloga

Aktivacija naloga se vrši na sledeći način:

Korisniku se alternativnim kanalom komunikacije saopšti aktivacioni kôd vezan za njegov korisnički nalog. Štampano i zapakovano u zaštitnoj koverti.

Korisnik prilikom prvog pokretanja aplikacije mora da unese aktivacioni kod.

Aktivacioni kod se proverava na strani servera (da li postoji) i ako je zaveden, aktivira se korisnik kome pripada.

Korisniku se u povratnoj poruci na ekranu aplikacije ispisuje PIN kôd koji mora da upiše ili zapamti da bi sa njim u buduće mogao da pokrene aplikaciju i pristupi formularu za zahtev za generisanje tokena za prijavu na portal.

4.4.4. Način prijave na portal

Korisnik prvo mora da izvrši generisanje tokena pomoću Android aplikacije. Generisanje je opisano u prethodnom delu. Nakon što je alternativnim kanalom dobio token, u formular za prijavu na portal, treba da unese svoje korisničko ime i lozinku na koju na kraj nadoveže token koji mu je generisan i koji je dobio preko Android aplikacije. Kada se prijavi, token se poništava i ne može se opet koristiti za prijave.

4.4.5. Bezbednost pristupa portalu

Portal beleži id broj PHP sesije, IP adresu i user agent string koji identifikuje Internet browser kojim se korisnik prijavio. U slučaju da neko presretne komunikaciju korisnika sa serverom nakon uspešne prijave i pokuša imitacijom poziva sa identičnim sadržajem vrednosti PHPSESSID i pomoćnog kolačića za identifikaciju prijave i pokuša da “session hijacking” tehnikom preuzme nalog i pristup zaštićenom sadržaju na portalu, sistem će proveriti još dva dodatna podatka (user agent string njegovog Internet browser-a i IP adresu dolaznog HTTP zahteva) i ako se ta dva podatka ne poklapaju automatski će uništiti sesiju i onemogućiti pristup portalu, a korisnik će biti redirektovan na stranicu za prijavu, tako da će morati da ponovi postupak zahteva za generisanjem novog tokena preko Android aplikacije i zatim da se prijavi na portal kako bi nastavio sa radim.

4.4.5.1. Implementacija aplikacija sa strane korisnika

Korisnik od administratora sistema dobija inicijalni kod za aktivaciju aplikacije. Ovaj kôd generiše se pri otvaranju naloga korisnika i jedinstven je za svakog korisnika. Korisniku se inicijalni kôd dodeljuje u obliku koji neće biti poznat službeniku koji pravi korisničko ime na sistemu već je kod sakriven i od samog službenika, dodeljuje se u papirnom zaštićenom obliku.

Korisnik aplikaciju preuzima sa sajta organizacije čije usluge planira da koristi i kod koje je prethodno otvorio nalog. Pri prvom pokretanju aplikacije biće potreban inicijalni kod.

4.4.5.2. Razmena informacija između aplikacije i servera

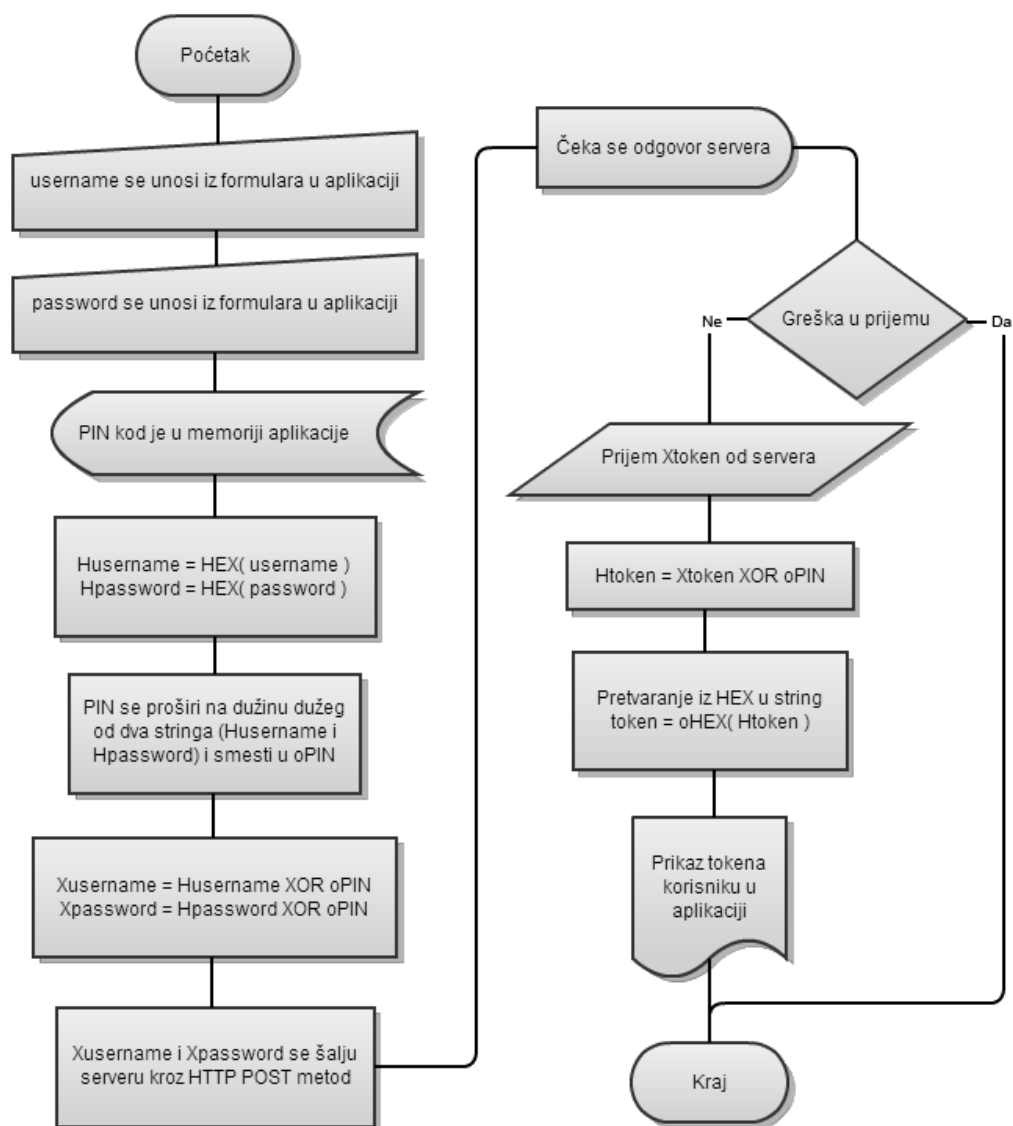
Aplikacija sa serverom komunicira preko Interneta kroz TCP/IP, sa informacijama zapakovanim u HTTP zahtev i odgovor. Sve informacije koje do servera dolaze od aplikacije tako što se svaki karakter poruke slovo za slovo šifruje XOR operacijom sa odgovarajućim karakterom PIN koda. Na strani servera, gde je PIN kôd takođe poznat, pristigli sadržaj se dekodira na isti način i tako dekodiran dalje obrađuje. U smeru od servera ka aplikaciji, postupak je identičan. Ni u jednom slučaju se kroz mrežu ne emituje PIN kôd, već on ostaje poznat samo na strani aplikacije i na strani servera⁶¹.

Razmena koda između Android aplikacije i servera se vrši na sledeći način:

Korisnik unosi korisničko ime i lozinku u tekstualnom obliku, gde je lozinka pri unosu maskirana. Uneti podaci se iz unikodnih tekstualnih vrednosti pretvaraju u ekvivalentne heksadecimalne zapise, takođe u tekstualnom obliku. PIN kod se proširuje prostim ponavljanjem do dužine veće od dužeg od dva stringa heksadecimalnih zapisa korisničkog imena i lozinke. Heksadecimalni zapisi se karakter za karakterom operacijom XOR šifruju proširenim zapisom PIN koda i dobijaju se enkodirani nizovi karaktera. Ovi nizovi karaktera se šalju serveru. Nakon prijema odgovora od servera, ako nije došlo do greške, primljeni token je šifrovan proširenim PIN kodom dobijenim na serverskoj strani na isti način kao u Android aplikaciji. Dekodiranjem operacijom XOR karakter za karakter sa proširenim PIN kodom dobija se heksadecimalni zapis tokena, koji se konverzijom iz heksadecimalnog zapisa pretvara u unikodni niz karaktera. Konačan dobijeni token se prikazuje korisniku.

⁶¹ Two-factor authentication for online banking: here are some key things to consider when trying to satisfy; Randy Schmidt (Sep 25, 2007)

Ceo opisani postupak je ilustrovan u sledećem dijagramu:



Slika 65: Dijagram prijave korisnika na sistem

Aplikacija zahteva definisanje novog koda koji će se ubuduće koristiti za pokretanje aplikacije. Po pravilu nakon tri pogrešna unošenja koda za pokretanje aplikacije korisničko ime i aplikacija se zaključavaju i izdaje se upozorenje administratorima sistema.

4.4.6. Izvorni kod aplikacije

Tehnička realizacija koncepta višekanalne autentifikacije je sprovedena u dva dela. Prvi je serverska strana sistema, koja podrazumeva opisanu bazu podataka i PHP aplikacije koja upravlja komunikacijom sa Android aplikacijom. Drugi deo sistema je Android aplikacija.

Serverski deo je pisan u skripting jeziku PHP i ceo kod se nalazi u prilogu. Android aplikacija je pisana u Java programskom jeziku, korišćenjem Android Source Development Kit-a (ASDK) i pisana je za verziju Android operativnog sistema 2.1 i novije. Aplikacija je rešena u četiri ekrana. Ekran za aktivaciju aplikacije, ekran za definisanje i izmenu PIN koda, ekran za otključavanje aplikacije definisanim PIN kodom i ekran za podnošenje zahteva za generisanjem i dostavom generisanog tokena za prijavu na portal. Komunikacija koja podrazumeva šifrovanje i dešifrovanje sadržaja koji saobraća između servera i aplikacije je podeljena u nekoliko metoda u osnovnim klasama programa. Izvorni kôd aplikacije se nalazi u prilogu. Najznačajniji je deo klase aplikacije koji upravlja logikom prikaza poruka, generisanja zahteva, kodiranja i dekodiranja sadržaja i za upravljanja PIN kodom.

Android aplikacija je klijentska strana sistema koja je zadužena za sledeće funkcionalnosti:

- Slanje zahteva za aktivaciju aplikacije i dodelu PIN koda
- Slanje zahteva za generisanje novog tokena

Komponente android aplikacije Tabela 9:

Activate.java	Klasa zadužena za upravljanje prozorom za aktivaciju koji se pojavljuje pri prvom startovanju instalirane aplikacije.
Pin.java	Zadužen je upravljanje prozorom za unos PIN koda pri startovanju aplikacije nakon što je aplikacija aktivirana.
Main.java	Klasa zadužena za upravljanje prozorom za slanje zahteva za generisanje tokena. Klasa upravlja enkodiranjem zahteva koji se šalju serveru, dekodiranjem odgovora od servera i prikaza generisanog tokena.
Data.java	Apstraktna klasa sa statičkim članovima, podacima koji ukazuju na putanju i nazive resursa na serveru.
Storage.java	Apstraktna klasa sa statičkim metodom za iščitavanje sačuvanog PIN koda u lokalnoj zaštićenoj memoriji aplikacije i statičkim metodom za upis PIN koda u zaštićenu lokalnu memoriju aplikacije.
Factory.java	Klasa koja se po jednom instancira u svakoj od klasa aktivnosti u aplikaciji i koja sadrži metode za enkodiranje, dekodiranje, tokenizaciju i uspostavljanje konekcije ka serveru.

Tabela 9: Komponente android aplikacije i njihova uloga

Serverska strana aplikacije je zadužena za obezbeđivanje sledećih funkcionalnosti:

- Aktivacija korisničkog naloga
- Saopštavanje PIN koda
- Generisanje i saopštavanje tokena
- Simulacija prijave na portal

Komponente serverske strane aplikacije Tabela 10:

Activate.php	Serverska skripta zadužena za preuzimanje poslatog zahteva za aktivaciju iz aplikacije. Skripta preuzima enkodirani aktivacioni kod od aplikacije, proverava u bazi podataka kojem korisniku je dodeljen taj aktivacioni kod, aktivira korisnički nalog i dodeljuje PIN kod. PIN kod enkodira i vraća aplikaciji koja će dekodirati i prikazati isti.
Config.php	Konfiguracioni php fajl sa parametrima za pristup bazi podataka.
Index.php	Serverska skripta za demonstraciju rada višekanalne autentifikacije. Skripta preuzima enkodirano korisničko ime i lozinku proširenu tokenom. Proverava u bazi podataka da li za korisničko ime i lozinku odgovara uneti token. Ukoliko odgovara, poništava token i preusmerava korisnika na panel.php. Ukoliko ne odgovara ispisuje poruku o grešci i vraća korisnika na formular za prijavu.
new_token.php	Osnovna php skripta zadužena za generisanje tokena. Od aplikacije preuzima zahtev za generisanje tokena praćen šifrovanim korisničkim imenom i lozinkom. Skripta dekodira korisničko ime i lozinku, proverava u bazi podataka da li postoje kao par i ukoliko postoje generiše novi token. Generisani token se upisuje u tabelu tokena a svi prethodni se poništavaju. Token se enkodira i vraća aplikaciji za prikaz. Ukoliko korisničko ime i lozinka ne postoje kao par u bazi podataka, skripta enkodira poruku o grešci koju će aplikacija korisniku prikazati.
panel.php	Skripta koja generiše HTML stranicu kada se korisnik uspešno prijavi. Stranica se stilizuje korišćenjem bootstrapa. Ima ulogu da omogućiti odjavljivanje korisnika.

Tabela 10: Komponente serverske strane aplikacije i njihova uloga

4.5. Hardverski uređaji za realizaciju napada

WiFi Pineapple Mark IV - Glavna primena WiFi Pineapple-a se pronalazi u testiranju sigurnosti bežičnih mreža. Kontrola i konfiguracija uređaja od strane napadača je moguća preko bežične ili žične mreže. Web interfejs je dostupan samo unapred definisanoj IP adresi, tako da ostali korisnici mreže ne mogu pristupiti. Jedna od karakteristika uređaja jeste emulacija SSID već sačuvanih (poznatih) mrežnih konekcija, tačnije svaki korisnik na svom računaru/telefonu/tabletu ima spisak već poznatih mreža (mreže sa kojima je već uspešno uspostavljena autentifikacija i konekcija), tako da korisnici uspostavljaju vezu sa uređajem bez njihovog znanja. Sav saobraćaj koji ostali korisnici mreže ostvare rutiraju se prvo preko računara napadača, a tek onda ka krajnjoj destinaciji. Preko web interfejsa je moguće pratiti sve zahteve za deautentifikaciju sa ostalim uređajima, kao i aktivne uređaje na mreži.



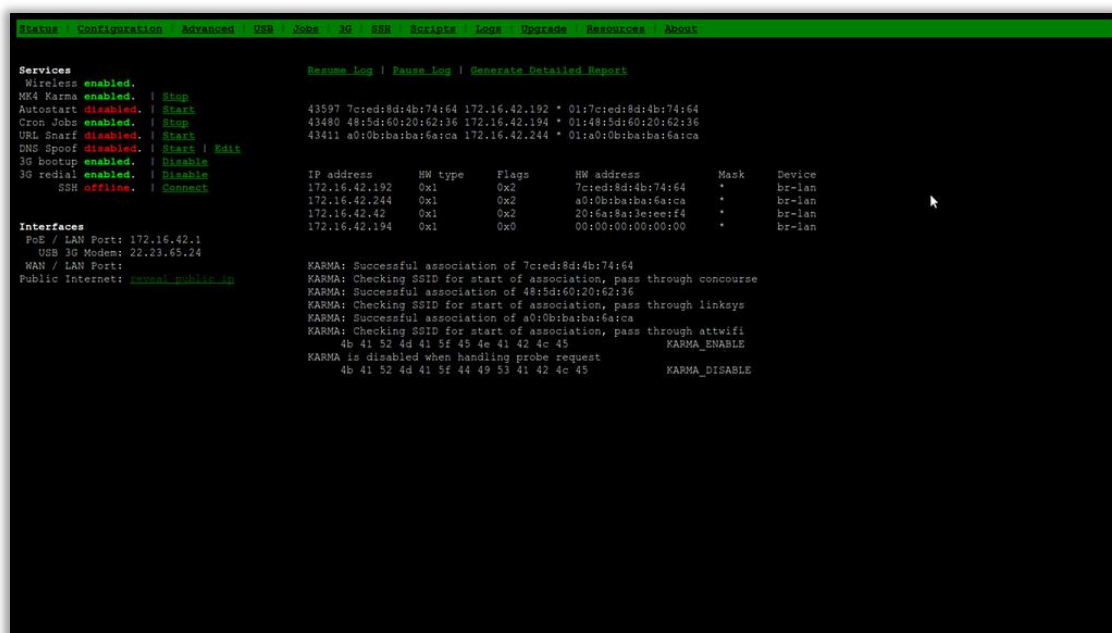
Slika 66: Fizička reprezentacija uređaja WiFi Pineapple Mark IV

4.5.1. Tehnički detalji o uređaju:

- Atheros AR9331 400MHz
- 802.11 b/g/n 150 Mbps
- 2x Ethernet + PoE
- USB 2.0
- OpenWRT (Linux Kernel 3.2)

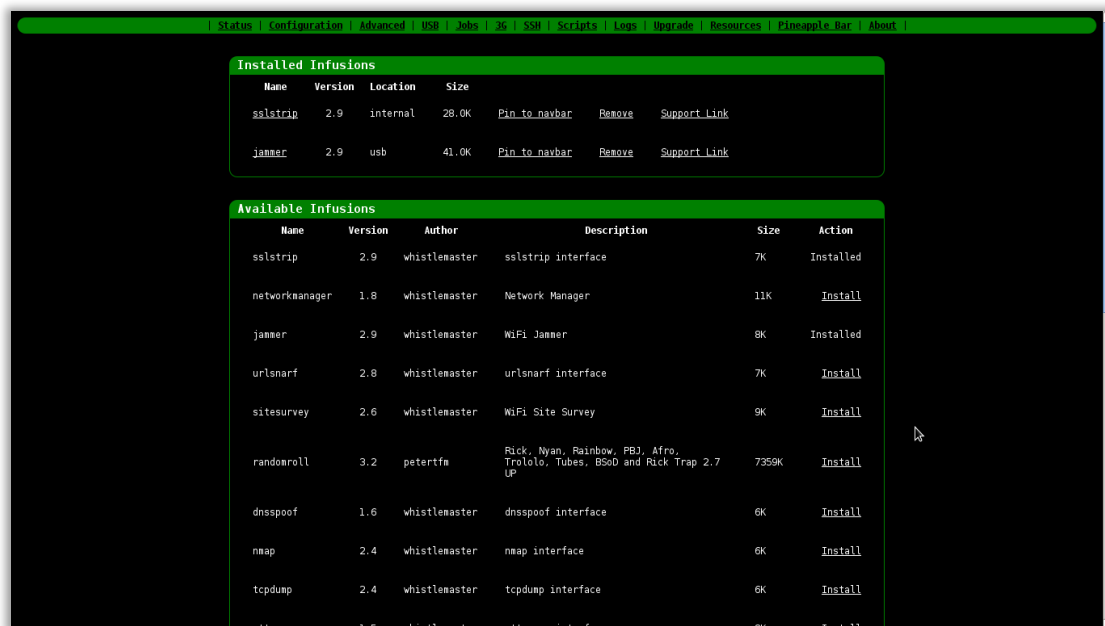
Mogućnosti i primena uređaja, prednost realizacije MITM napada sa WiFi Pineapple-om nad realizacije sa prenosnim računarom su pre svega male dimenzije, mogućnost dodavanja prenosnog napajanja i autonoman rad. Zbog podrške Linux kernela, WiFi Pineapple nudi mogućnost pokretanja različitih servisa kako bi olakšao rad praćenja aktivnosti mreže:

- SSL Strip
- DHCP Server
- HTTP Server
- PHP
- Signal Jammer



Slika 67: Korisnički interfejs WiFi Pineapple Mark IV

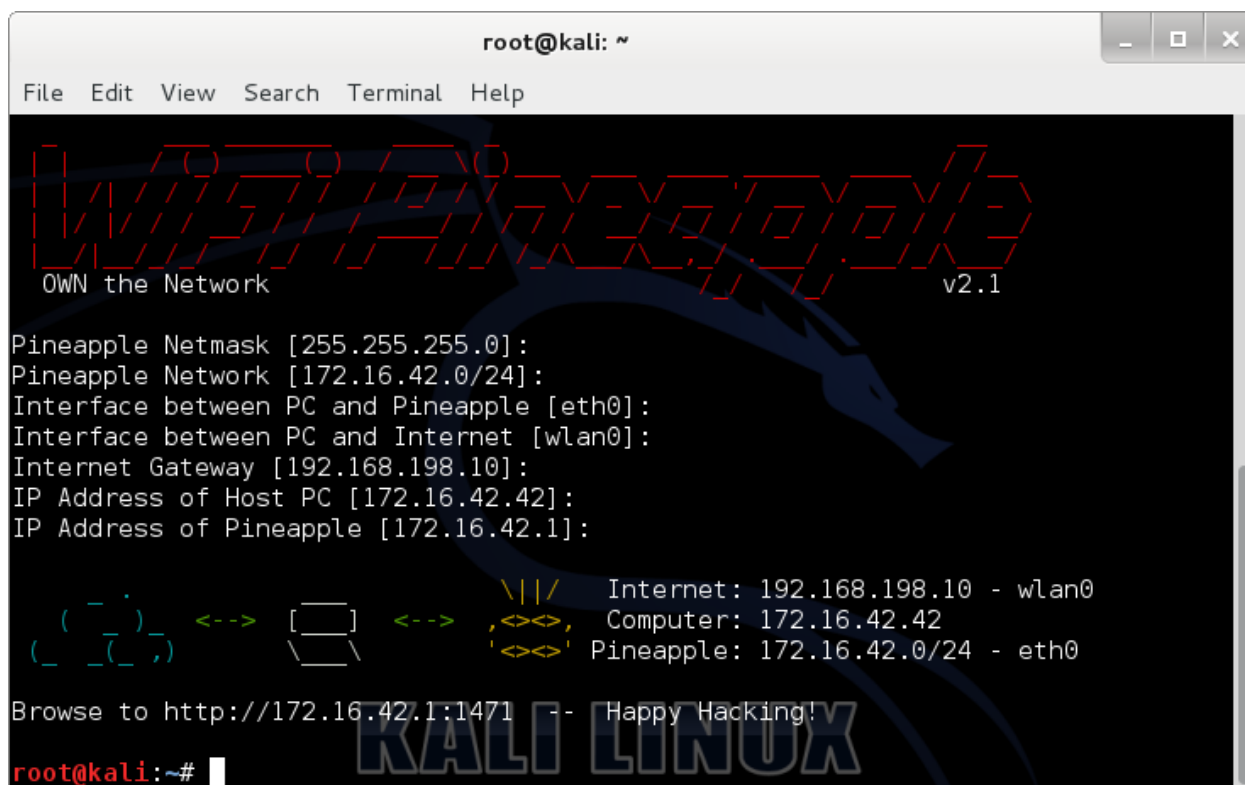
Svi dodatni servisi su dostupni za instalaciju, kako kroz web interfejs, tako kroz CLI opkg menadžer. Paketi servisa se preuzimaju sa wifipineapple.com. Svaki od naknadno instaliranih servisa se pojavljuje u „Pineapple Bar“ delu u web interfejsu.



Slika 68: Instalacija dodatnih servisa WiFi Pineapple Mark IV

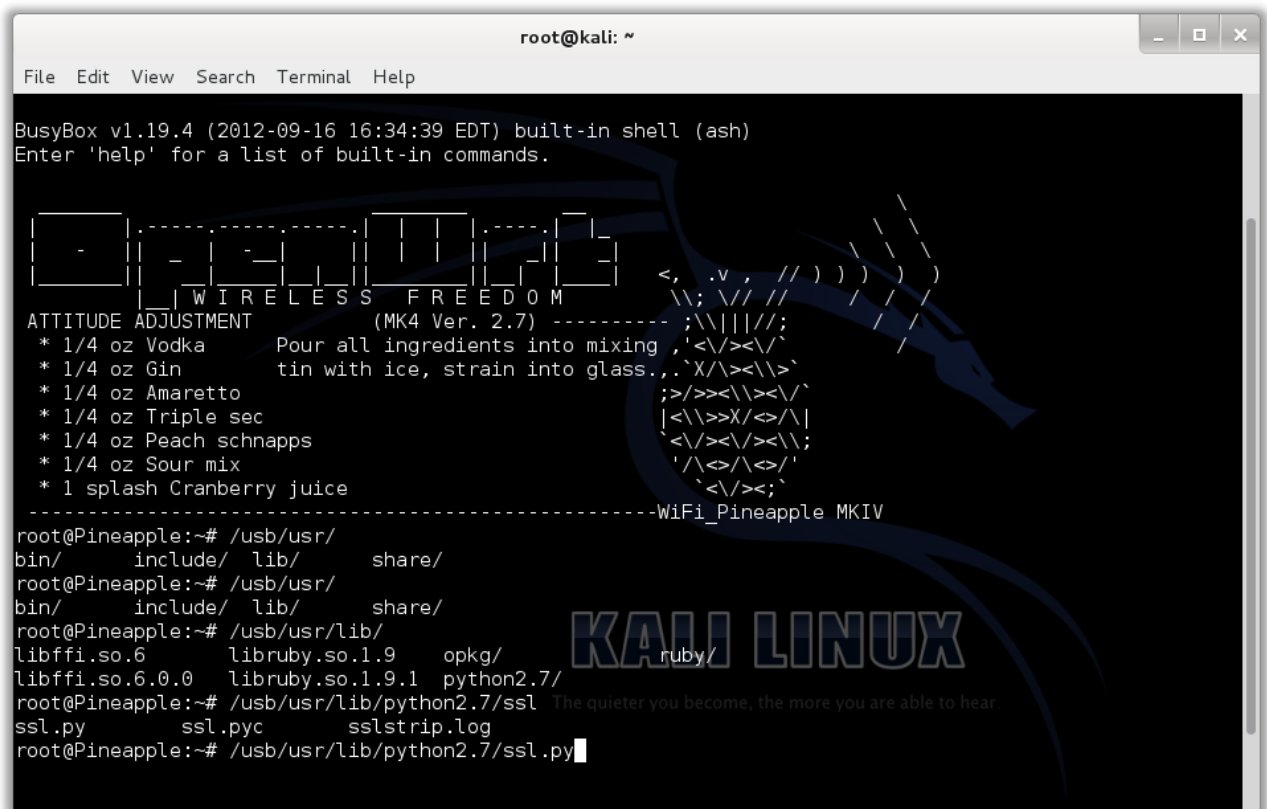
Kompletnu konfiguraciju i dodatne instalacije je moguće obaviti SSH konekcijom na sam uređaj. Ovim putem se dobija pristup kompletom sistemu, tako da je moguće podešavanje sistemskih parametara:

- IP adresa
- Prosleđivanje portova
- Automatsko učitavanje skripti
- 3G Konekcija



Slika 69: Konfiguracija i parametra WiFi Pineapple Mark IV

U slučaju da postoji potreba da se mreža prati na takav način da HTTP protokol bude jedini („strip-ovanje“ HTTPS-a), moguće je instalirati i pokrenuti SSL Strip kroz web interfejs ili CLI. Kasnije u tom postupku, na napadačevom računaru bi se pokrenuo alat za praćenje mrežnog saobraćaja (Wireshark).



Slika 70: Priprema SSL Strip napada korišćenjem WiFi Pineapple Mark IV

Zbog mogućnosti korišćenja 3G konekcije, proširenja memorije preko USB 2.0 interfejsa, poseban deo „Scripts“ korisniku daje opciju da određene skripte izvrši neposredno posle paljenja uređaja kako bi servisi bili dostupni bez potrebe povezivanja na uređaj.

4.5.2. Korišćenja hardverskog rešenja baziranog na WiFi Pineapple Mark IV

Prednost nad „tradicionalnim“ MITM napadom (prenosni računar) su svakako male dimenzije i prenosivost. Dodavanjem eksternog napajanja u vidu litijum-jonskih baterija, aktivan rad uređaja dostiže 4 sata, pritom uređaj može ostati skriven u rancu ili na nekom drugom mestu kako ne bi privlačio pažnju ostalih korisnika mreže.

Izvođenje MITM napada na bežičnu mrežu je dosta lako detektovati instalacijom IDS/IPS sistema, samim tim napad može biti sprečen. Kod MITM napada, u kome se ne koriste specijalizovani uređaji već prenosni računar/tablet, korišćenjem alata za menjanje ARP tabela rutera/sviča, detekcija anomalija je laka, ne retko sami antivirus sistemi korisnika

obaveštavaju da je moguća meta napada. U scenariju gde se napad sprovodi korišćenjem Pineapple uređaja procenat detekcije od strane IDS/IPS uređaja je znatno manji iz razloga što će uređaj biti registrovan kao validni ruter sa kojim korisnici mogu imati interakciju, a ne kao maliciozni korisnik. Sa korisnikove strane se dešava regularna interakcija korisnik – ruter – internet, kako je Pineapple na mreži registrovan kao regularan ruter, nikakve anomalije neće biti detektovane.

Mana koja bi mogla da se istakne jeste nedostatak resursa za održavanje mreže sa više korisnika. Procesor od 400MHz nije dovoljan za opsluživanje više korisnika sa istovremeno pokrenutim modulima (SSL Strip, Signal Jammer...). Zbog nedostatka dodatnog hlađenja i malih dimenzija, uređaj se brzo greje, što zna da uspori celokupan rad.

5. ZAKLJUČAK

Polazna osnova rada bila je potreba za ukazivanjem na bezbednosne nedostatke koji mogu naneti štetu i kompromitovati podatke korisnika savremenih računarskih mreža. Rad je sagledao bezbednosti na bežičnoj wireless mreži. Prikazana je kriptografska arhitektura wireless uređaja, bezbednost, bezbednosne pretnje i mere za ublažavanje tih pretnji. Bežično povezivanje predstavlja odličnu zamenu za kablove, naročito na manjim rastojanjima, ali to sa sobom donosi i određene rizike i pretnje zbog mogućeg neovlašćenog pristupa podacima. Za mnoge komercijalne korisnike bežične računarske mreže predstavljaju idealno rešenje zbog dometa, brzine prenosa, različitih mogućnosti povezivanja i kompatibilnosti. Kroz rad je napravljena relacija između sirunosti bežičnih računarskih mreža i SSL protokola. Poslovne organizacije svoje usluge postavljaju na poznatu infrastrukturu i u najvećem broju slučajeva štite SSL/TLS protokolom uz korišćenje podataka koji su korisniku poznati (korisničko ime i lozinka). Na ovaj način postignut je balans između funkcionalnosti i zaštite, zanemarujući pravila bezbednosti i autentifikacije za nečim što korisnik ima i nečim što korisnik jeste. Kroz rad je predstavljena i razmotrena mogućnosti uvođenja sopstvenog rešenja koje ne umanjuje funkcionalnost uvodeći i drugi princip pravila bezbedne autentifikacije nečim što korisnik ima.

5.1. Sumirani ciljevi istraživanja

Istraživanje u oblasti javilo se kao potreba za sveobuhvatnom analizom sistema bežičnih računarskih mreža, sistema elektronske trgovine i sistema elektronskog bankarstva. Polazna pretpostavka da ovi sistemi često nisu pravilno implementirani, dokazana je eksperimentalnom analizom. Dalja analiza utvrdila je da čak i pravilno implementirani sistemi mogu biti kompromitovani. Nakon završenih analiza i utvrđivanja propusta kao sa teorijske osnove tako i sa praktično ugrožene bezbednosti korisnika ovih sistema, predstavljeni su koncepti mogućih rešenja prevazilaženja ovih problema.

Analizom poslovnih okruženja bezbednosni mehanizmi su uglavnom obezbeđeni preko WEP i WPA2 protokola za zaštitu podataka u različitim formama. Celokupna bezbednost na bežičnoj mreža zavisi od bezbednosti uređaja, bezbednosti protokola i

bezbednosnih parametara koji se koriste u komunikaciji. Na nivou usluga se koriste mehanizmi kontrole pristupa, dok se za bezbednost na nivou veze koriste različiti kriptografski mehanizmi.

U radu su predstavljena trenutno aktuelna open-source i komercijalna rešenja za analizu i napade na bežične računarske mreže štićene WPA2-PSK standardom. Analizom je potvrđena pretpostavka da postoje dva prihvatljiva načina za napad i analizu saobraćaja bežičnih mreža štićenih najaktuelnijom metodom zaštite. Pri način bazira se na napadu zasnovanom na korišćenju rečnika. Drugi tip napada je pretraga po svim mogućim kombinacijama. Prvi tip napada pokazao se značajno bržima međutim bez garancije za dobijanje pristupa bežičnoj računarskoj mreži. Drugi tip napada garantuje pronalazak pristupnim parametrima, nažalost u vremenskim okvirima koji često nisu prihvatljivi.

Nakon analize a u okviru istraživanja predstavljeno je i sopstveno rešenje za prevazilaženje problema vremenskog ograničenja implementiranjem distribuiranog računarstva i paralelne obrade podataka. Rezultati su pokazali da distribuirano računarstvo korišćenjem centralnog procesora više računara može doprineti rešenju vremenskog problema napada na lozinku štićenu WPA2-PSK standardom. Kako WPA2 podržava predefinisani ključ dužine od 8 do 63 karaktera, potrebno je napomenuti da je rešenje primenjivo i racionalno na predefinisanim ključevima od 8 do 12 karaktera. Takođe je bitno napomenuti da predefinisani ključevi od 8 do 12 karaktera upravo i jesu najčešće korišćeni pri implementaciji WPA2-PSK standarda zaštite bežične računarske mreže.

Dalja analiza obuhvaćena je zaštitu sistema elektronske trgovine i elektronskog bankarstva u sprezi sa bežičnim računarskim mrežama. Za zaštitu komunikacije u ovima sistemima bez obzira na tim računarske mreže velike kompanije i ustanove često svoje i poverenje svojih klijenata stavljaju na SSL/TSL protokol. Rad je pokazao da sama po sebi razmena podataka štićena SSL vezom nije dovoljna i da može doći do kompromitovanja podataka klijenta/korisnika.

Elektronsko poslovanje nesporno je donelo mnoge „blagodeti“, ali i brojne opasnosti. Što je njegov rast i zastupljenost veća, to su i posledice zloupotrebe teže. U zemljama u kojima se ovaj oblik poslovanja nalazi u stalnoj ekspanziji, preduzimaju se mnogobrojne aktivnosti u zaštiti, ali i hvatanju i kažnjavanju onih kojima zloupotrebe postaju „zanimanje“.

Nakon analize problema predloženo je sopstveno rešenje zasnovano na teorijskim osnovama autentifikacije koje podrazumevaju da idealna autentifikacija sadrži:

- Nešto što korisnik zna (korisničko ime i lozinka)
- Nešto što korisnik ima (telefon, sms, token, smart card)
- Nešto što korisnik jeste (otisak prsta, glas, iris)

U osnovnu autentifikaciju koja se zasnivala na podacima koje korisnik zna uveden je koncept implementacije segmenta nešto što korisnik ima. Koncept implementacije podrazumeva korišćenje mobilne Android aplikacije korišćenjem drugog kanala komunikacije. Na ovaj način ostvarena su dva principa idealne autentifikacije uz višekanalnu razmenu podataka. Ovime je mogućnost napada svedena na minimum.

5.2. Ostvareni rezultati i doprinos

Pored samog sagledavanja problema bežičnih računarskih mreža rad je sagledao i problem zaštite podataka najsavremenijim metodama. Ostvarena je sveobuhvatna analiza autentifikacije korisnika kroz bežičnu računarsku mrežu ka sistemima elektronske trgovine i elektronskog bankarstva. Kroz rad je prikazano da i u najpouzdanijim metodama koje se koriste u najvećim sistemima, za koje većina smatra da su bezbedni, postoje mana koje zlonamerni korisnici mogu iskoristiti. Od situacije u kojoj su napade sprovodili retki izuzetno upućeni i teorijski dobro potkovani pojedinci, došlo se do situacije da današnji alati za penetracijske testove ne zahtevaju previše znanja od korisnika. Ovakve alate koji su prevashodno napravljeni za penetracijsko testiranje i dijagnostikovanje računarskih mreža zlonamerni pojedinci koriste u cilju dobijanja lične koristi.

Za zaštitu komunikacije bez obzira na tip računarske mreže velike kompanije i ustanove često svoje poverenje stavljaju na SSL/TSL metod. Rad je pokazao da sama po sebi razmena podataka štićena SSL vezom nije dovoljna i da može doći do kompromitovanja podataka klijenta/korisnika. Osnovni doprinosi rada ogledaju se kroz sledeće činioce:

1. Sveobuhvatna analiza sistema bežičnih računarskih mreža. Obavljena je analiza signala i zaštite svih poznatih standarda koji se koriste kako u korisničkim tako i u profesionalnim računarskim mrežama.

2. Analizirani su aktuelni alati i sistemi za penetracijsko testiranje bežičnih računarskih mreža. Aktuelna rešenja implementirana kroz sopstvenu realizaciju dovela su do potrebe za daljom analizom i sopstvenom implementacijom.
3. Implementiran je sopstveni sistem distribuiranog računarstva za analizu i obradu sistema zaštite bežičnih računarskih mreža.
4. Izložena su i analizirana trenutno aktuelna rešenja zaštite bazirana na SSL/TLS protokolu. Ukazano je na propuste u implementaciji istog, ali i na nedostatke čak i pri pravilnoj implementaciji.
5. Na osnovu analiza a u skladu sa teorijskim osnova idealne autentifikacije predložen je sopstveni koncept sa praktičnom implementacijom zaštite veze štićene SSL protokolom.
6. Analizom je potvrđeno da višekanalna autentifikacija značajno smanjuje mogućnost napada na sisteme elektronske trgovine i elektronskog bankarstva, što je dovelo rezultovalo konceptom sopstvenog rešenja implementacije bimodalnog višekanalnog sistema autentifikacije.
7. Razvijeno rešenje je nastalo kao rezultat petogodišnjeg rada i istraživanja u ovoj oblasti. Implementacija je obavljena uz pomoć trenutno najaktuelnije mobilne platforme – Android, dok je sam stepen bezbednosti koju donosi SSL podignut na novi nivo korišćenjem kako tradicionalnih i opštih pravila bezbednosti tako i sopstvenim rešenjem i implementacijom istog.

5.3. Predlog daljeg rada

Deo rezultata i analiza predstavljenih u ovom radu već je izlagan kroz naučne konferencije, stručne skupova i individualna savetovanja. Rezultati savetovanja naišli su na odličan odziv i interesovanje učesnika i predstavljeni su u samom radu. Na taj način ne dvosmisleno je ukazano da problem postoji i da postoji potreba za pravilnom implementacijom odgovarajućeg rešenja.

Predmet daljeg rada u oblasti zaštite komunikacije unutar bežičnih računarskih mreža će se bazirati isključivo na grafičkim procesorima. Budući da isti obezbeđuju značajnije ubrzanje i potencijalno mogu sprovesti napad i na duže predefinisane ključeve u oblasti kompromitovanja razmene podataka.

U oblasti povećanja bezbednosti sistema pažnja treba biti usmerena na implementaciju i prototip predstavljenog rešenja višekanalne bimodalne autentifikacije. Sa sve bržim razvojem mobilnih platformi na čelu sa Android platformom nije ne realno očekivati dalje istraživanje na polju trećeg principa idealne autentifikacije tj. nečega što korisnik jeste.

Trenutni koncept višekanalne višemodalne autentifikacije predstavljen u ovom radu u daljem istraživanju može uključiti i biometrijske metode kao pomoć SSL protokolu u cilju obezbeđivanja sigurnog rada korisnika u sistemima elektronske trgovine i elektronskog bankarstava, ali i svim budućim sistemima koji nas očekuju.

6. PRILOZI I DOKUMENTACIJA

6.1. [Prilog 1. Android aplikacija](#)

Android aplikacija je klijentska strana sistema koja je zadužena za sledeće funkcionalnosti:

- Slanje zahteva za aktivaciju aplikacije i dodelu PIN koda
- Slanje zahteva za generisanje novog tokena

6.1.1. [Activate.java](#)

Klasa zadužena za upravljanje prozorom za aktivaciju koji se pojavljuje pri prvom startovanju instalirane aplikacije.

```
public class Activate extends Activity {  
    private static EditText PART1, PART2;  
    private static Button ACTIVATE_BUTTON, CONTINUE;  
    private static TextView RES;  
  
    private static String res = "";  
  
    Handler ha = new Handler() {  
        @Override  
        public void handleMessage(Message msg) {  
            ACTIVATE_BUTTON.setVisibility(Button.GONE);  
            CONTINUE.setVisibility(Button.VISIBLE);  
        }  
    };  
  
    Handler h = new Handler() {  
        @Override  
        public void handleMessage(Message msg) {  
            RES.setText( res );  
        }  
    };  
}
```

```
        }
    };

    public static boolean isNumeric(String str) {
        try {
            Integer.parseInt(str);
        } catch (NumberFormatException nfe) {
            return false;
        }
        return true;
    }

    private StringBuilder inputStreamToString(InputStream is) {
        String line = "";
        StringBuilder total = new StringBuilder();
        BufferedReader rd = new BufferedReader(new InputStreamReader(is));
        try {
            while ((line = rd.readLine()) != null) {
                total.append(line);
            }
        } catch (IOException e) {
            e.printStackTrace();
        }
        return total;
    }

    private class MyAsyncTask extends AsyncTask<Void, Void, Void> {
        ProgressDialog mProgressDialog;

        @Override
        protected void onPostExecute(Void result) {
            mProgressDialog.dismiss();
        }

        @Override
        protected void onPreExecute() {
```

```
mProgressDialog = ProgressDialog.show(Activate.this, "Activating...", "Activating
your account...");
}
@Override
protected Void doInBackground(Void... params) {
    activate();
    return null;
}
}

public void activate() {
    res = "Please wait...";
    h.sendEmptyMessage(0);

    DefaultHttpClient httpclient = new DefaultHttpClient();
    HttpPost httppost = new HttpPost("http://" + Data.domain + Data.activate);

    String code = PART1.getText().toString() + "-" + PART2.getText().toString();

    List<BasicNameValuePair> nameValuePairs = new
ArrayList<BasicNameValuePair>(1);
    nameValuePairs.add(new BasicNameValuePair("code", code));
    try {
        httppost.setEntity(new UrlEncodedFormEntity(nameValuePairs));
    } catch (UnsupportedEncodingException e1) {
        e1.printStackTrace();
    }
    try {
        httpclient.execute(httppost);

        HttpResponse response = httpclient.execute(httppost);
        String r = inputStreamToString(response.getEntity().getContent()).toString();

        if ( !isNumeric(r) ) {
```

```
        res = "There was an error while attempting to retrieve your PIN  
code!";  
  
        } else {  
            Storage.SavePin(r, getApplicationContext());  
            res = "Your PIN code is:\n\n" + r + "\n\nPlease remember it or  
write it down and do not forget it or lose it!";  
            ha.sendMessage(0);  
        }  
        h.sendMessage(0);  
    } catch (Exception e) {  
        res = e.getMessage();  
        h.sendMessage(0);  
    }  
}
```

@Override

```
public void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_activate);  
  
    PART1 = (EditText) findViewById(R.id.PART1);  
    PART2 = (EditText) findViewById(R.id.PART2);  
    ACTIVATE_BUTTON = (Button) findViewById(R.id.ACTIVATE_BUTTON);  
    CONTINUE = (Button) findViewById(R.id.CONTINUE);  
  
    RES = (TextView) findViewById(R.id.RES);  
  
    ACTIVATE_BUTTON.setOnClickListener(new OnClickListener() {  
        @Override  
        public void onClick(View v) {  
            MyAsyncTask m = new MyAsyncTask();  
            m.execute(null);  
        }  
    });  
});
```

```
CONTINUE.setOnClickListener(new OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        Intent i = new Intent(getApplicationContext(), Pin.class);  
        startActivity(i);  
        finish();  
    }  
});  
}  
}
```

6.1.2. [Pin.java](#)

Zadužen je upravljanje prozorom za unos PIN koda pri startovanju aplikacije nakon što je aplikacija aktivirana.

```
public class Pin extends Activity {  
  
    private static EditText PIN;  
    private static Button ENTER_PIN;  
  
    @Override  
    public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
  
        if (Storage.LoadPin(getApplicationContext()) == "") {  
            Intent i = new Intent(getApplicationContext(), Activate.class);  
            startActivity(i);  
        } else {  
            setContentView(R.layout.activity_pin);  
  
            PIN = (EditText) findViewById(R.id.PIN);  
            ENTER_PIN = (Button) findViewById(R.id.ENTER_PIN);  
        }  
    }  
}
```

```
ENTER_PIN.setOnClickListener(new OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        String pin = PIN.getText().toString();  
        if (  
pin.equals(Storage.LoadPin(getApplicationContext())) ) {  
            Intent i = new Intent(getApplicationContext(),  
Main.class);  
            startActivity(i);  
            PIN.setText("");  
        }  
    }  
});  
}  
  
}
```

6.1.3. [Main.java](#)

Klasa zadužena za upravljanje prozorom za slanje zahteva za generisanje tokena. Klasa upravlja enkodiranjem zahteva koji se šalju serveru, dekodiranjem odgovora od servera i prikaza generisanog tokena.

```
public class Main extends Activity {  
  
    public static EditText USER, PASS, SALT;  
    public static Button REQUEST;  
    public static String salt = "";  
  
    private class MyAsyncTask extends AsyncTask<Void, Void, Void> {  
        ProgressDialog mProgressDialog;
```

```
@Override
protected void onPostExecute(Void result) {
    mProgressDialog.dismiss();
}

@Override
protected void onPreExecute() {
    mProgressDialog = ProgressDialog.show(Main.this, "Loading...", "Getting salt...");
}

@Override
protected Void doInBackground(Void... params) {
    setup();
    return null;
}
}

Handler h = new Handler() {
    @Override
    public void handleMessage(Message msg) {
        SALT.setText( salt );
    }
};

@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    if (Storage.LoadPin(getApplicationContext()) == "") {
        Intent i = new Intent(getApplicationContext(), Activate.class);
        startActivity(i);
    } else {
        setContentView(R.layout.main);

        USER = (EditText) findViewById(R.id.USER);
        PASS = (EditText) findViewById(R.id.PASS);
        SALT = (EditText) findViewById(R.id.SALT);
    }
}
```

```
REQUEST = (Button) findViewById(R.id.REQUEST);
REQUEST.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View v) {
        // setup();
        MyAsyncTask m = new MyAsyncTask();
        m.execute(null);
    }
});

}

}

public static boolean isNumeric(String str) {
    try {
        Integer.parseInt(str);
    } catch (NumberFormatException nfe) {
        return false;
    }
    return true;
}

public static String hexadecimal(String input, String charsetName) throws
UnsupportedEncodingException {
    if (input == null) throw new NullPointerException();
    return asHex(input.getBytes(charsetName));
}

private static final char[] HEX_CHARS = "0123456789abcdef".toCharArray();
public static String asHex(byte[] buf)
{
    char[] chars = new char[2 * buf.length];
    for (int i = 0; i < buf.length; ++i)
    {
        chars[2 * i] = HEX_CHARS[(buf[i] & 0xF0) >>> 4];
        chars[2 * i + 1] = HEX_CHARS[buf[i] & 0x0F];
    }
    return new String(chars);
}
```

```
}  
public String xorHex(String a, String b) {  
    try {  
        a = hexadecimal(a, "UTF-8");  
        b = hexadecimal(b, "UTF-8");  
    } catch (Exception e) { }  
    b = b + b + b + b + b;  
    char[] chars = new char[a.length()];  
    for (int i = 0; i < chars.length; i++) {  
        Log.d("LOG", "LOG: Encoding character " + a.charAt(i));  
        chars[i] = toHex(fromHex(a.charAt(i)) ^ fromHex(b.charAt(i)));  
    }  
    return new String(chars);  
}  
private static int fromHex(char c) {  
    Log.d("LOG", "LOG: testing character " + c );  
    if (c >= '0' && c <= '9') return c - '0';  
    if (c >= 'A' && c <= 'F') return c - 'A' + 10;  
    if (c >= 'a' && c <= 'f') return c - 'a' + 10;  
    throw new IllegalArgumentException();  
}  
private char toHex(int nybble) {  
    if (nybble < 0 || nybble > 15) throw new IllegalArgumentException();  
    return "0123456789ABCDEF".charAt(nybble);  
}  
public void setup() {  
    try {  
        HttpClient httpclient = new HttpClient();  
        httpclient.getHostConfiguration().setHost(Data.domain);  
        PostMethod httppost = new PostMethod(Data.page);  
  
        String username = USER.getText().toString();  
        String password = PASS.getText().toString();
```

```
        username = xorHex(username,
Storage.LoadPin(getApplicationContext()));
        password = xorHex(password,
Storage.LoadPin(getApplicationContext()));

        httppost.addParameter( new NameValuePair("username", username ));
        httppost.addParameter( new NameValuePair("password", password ));

        try {
            httpclient.executeMethod(httpppost);
            String res = httpppost.getResponseBodyAsString();
            if ( isNumeric(res) ) {
                salt = res;
                h.sendEmptyMessage(0);
                // SALT.setText( res );
            } else {
                // salt = res;
                salt = "An error has accured!";
                h.sendEmptyMessage(0);
                // SALT.setText("An error has occurred!");
            }
        } finally {
            httpppost.releaseConnection();
        }

    } catch (Exception e) {
        salt = "Error:\n" + e.getMessage() + "\n";
        h.sendEmptyMessage(0);
        // SALT.setText( "Error:\n" + e.getMessage() + "\n" );
        e.printStackTrace();
    }
}
}
```

6.1.4. [Data.java](#)

Apstraktna klasa sa statičkim članovima, podacima koji ukazuju na putanju i nazive resursa na serveru.

```
public abstract class Data {  
    public static String domain = "host.singidunum.ac.rs";  
    public static String page = "/msarac/new_token.php";  
    public static String activate = "/msarac/activate.php";  
}
```

6.1.5. [Storage.java](#)

Apstraktna klasa sa statičkim metodom za iščitavanje sačuvanog PIN koda u lokalnoj zaštićenoj memoriji aplikacije i statičkim metodom za upis PIN koda u zaštićenu lokalnu memoriju aplikacije.

```
public abstract class Storage {  
    public static void SavePin(String PIN, Context context) {  
        String FILENAME = "pin.dat";  
        try {  
            FileOutputStream fos = context.openFileOutput(FILENAME,  
Context.MODE_PRIVATE);  
            fos.write(PIN.getBytes());  
            fos.close();  
        } catch (Exception e) { }  
    }  
    public static String LoadPin(Context context) {  
        String FILENAME = "pin.dat";  
        try {  
            FileInputStream is = context.openFileInput(FILENAME);  
            BufferedReader br = new BufferedReader(new  
InputStreamReader(is));
```

```
        String PIN = br.readLine();
        br.close();
        is.close();
        return PIN;
    } catch (Exception e) {
        return "";
    }
}
}
```

6.1.6. [Factory.java](#)

Klasa koja se po jednom instancira u svakoj od klasa aktivnosti u aplikaciji i koja sadrži metode za enkodiranje, dekodiranje, tokenizaciju i uspostavljanje konekcije ka serveru.

```
public class Factory extends SSLSocketFactory implements ProtocolSocketFactory {
    SSLContext sslContext = SSLContext.getInstance("TLS");
    public Factory() throws NoSuchAlgorithmException, KeyManagementException,
    KeyStoreException, UnrecoverableKeyException {
        super();
        TrustManager tm = new X509TrustManager() {
            @Override
            public X509Certificate[] getAcceptedIssuers() {
                // TODO Auto-generated method stub
                return null;
            }
            @Override
            public void checkServerTrusted(X509Certificate[] chain, String
authType)
                throws CertificateException {
                // TODO Auto-generated method stub
            }
            @Override
```

```
        public void checkClientTrusted(X509Certificate[] chain, String
authType)

                throws CertificateException {
                // TODO Auto-generated method stub

        }

    };

    sslContext.init(null, new TrustManager[] { tm }, null);
}

    public Factory(SSLContext context) throws KeyManagementException,
NoSuchAlgorithmException, KeyStoreException, UnrecoverableKeyException {
        super();
        sslContext = context;
    }

    @Override
    public Socket createSocket(Socket socket, String host, int port, boolean autoClose) throws
IOException, UnknownHostException {
        return sslContext.getSocketFactory().createSocket(socket, host, port, autoClose);
    }

    @Override
    public Socket createSocket() throws IOException {
        return sslContext.getSocketFactory().createSocket();
    }

    @Override
    public String[] getDefaultCipherSuites() {
        // TODO Auto-generated method stub
        return null;
    }

    @Override
    public String[] getSupportedCipherSuites() {
        // TODO Auto-generated method stub
        return null;
    }

    @Override
    public Socket createSocket(String host, int port) throws IOException,
```

```
        UnknownHostException {
            // TODO Auto-generated method stub
            return null;
        }

        @Override
        public Socket createSocket(InetAddress host, int port) throws IOException {
            // TODO Auto-generated method stub
            return null;
        }

        @Override
        public Socket createSocket(String host, int port, InetAddress localHost,
            int localPort) throws IOException, UnknownHostException {
            // TODO Auto-generated method stub
            return null;
        }

        @Override
        public Socket createSocket(InetAddress address, int port,
            InetAddress localAddress, int localPort) throws IOException {
            // TODO Auto-generated method stub
            return null;
        }

        @Override
        public Socket createSocket(String arg0, int arg1, InetAddress arg2,
            int arg3, HttpConnectionParams arg4) throws IOException,
            UnknownHostException, ConnectTimeoutException {
            // TODO Auto-generated method stub
            return null;
        }
    }
}
```

6.2. Prilog 2. Serverska strana aplikacije

Serverska strana aplikacije je zadužena za obezbeđivanje sledećih funkcionalnosti:

- Aktivacija korisničkog naloga
- Saopštavanje PIN koda
- Generisanje i saopštavanje tokena
- Simulacija prijave na portal

6.2.1. Activate.php

Serverska skripta zadužena za preuzimanje poslatog zahteva za aktivaciju iz aplikacije. Skripta preuzima enkodirani aktivacioni kod od aplikacije, proverava u bazi podataka kojem korisniku je dodeljan taj aktivacioni kod, aktivira korisnički nalog i dodeljuje PIN kod. PIN kod enkodira i vraća aplikaciji koja će dekodirati i prikazati isti.

```
<?php
    include 'config.php';

    function MyWayToDie($msg = ") {
        global $con;
        @mysql_close($con);
        die($msg);
    }

    if ( count($_POST) > 0 ) {
        $con = mysql_connect(host, user, pass) or MyWayToDie("Can't connect to database
server!");
        mysql_select_db(base) or MyWayToDie("Can't select the database!");
        $activation_code = mysql_real_escape_string( $_POST['code'], $con );
```

```
$SQL = "SELECT pin, id FROM users WHERE activation_code =
'{$activation_code}';";

$res = mysql_query($SQL, $con);
if ($res and mysql_num_rows($res)) {
    $data = mysql_fetch_assoc($res);
    $user_pin = $data['pin'];
    $id      = $data['id'];
    mysql_free_result($res);

    $SQL = "UPDATE users SET active = 1 WHERE id = '{$id}';";
    $res = mysql_query($SQL, $con);
    if ($res) {
        MyWayToDie( $user_pin );
    } else {
        MyWayToDie( "Could not activate user account!" );
    }

} else {
    MyWayToDie( "User already active or bar activation code!" );
}

} else {
    MyWayToDie( "Illegal function call!" );
}

}
```

6.2.2. [Config.php](#)

Konfiguracioni php fajl sa parametrima za pristup bazi podataka.

```
<?php
define('host', 'localhost');
define('user', 'root');
define('pass', '');
define('base', 'sifra_sarac');
```

6.2.3. [Index.php](#)

Serverska skripta za demonstraciju rada višekanalne autentifikacije. Skripta preuzima enkodirano korisničko ime i lozinku proširenu tokenom. Proverava u bezi podataka da li za korisničko ime i lozinku odgovara uneti token. Ukoliko odgovara, poništava token i preusmerava korisnika na panel.php. Ukoliko ne odgovara ispisuje poruku o grešci i vraća korisnika na formular za prijavu.

```
<?php
include 'config.php';
session_start();
if (!isset($_SESSION['MESSAGE'])) $_SESSION['MESSAGE'] = "";

function MyWayToDie($msg = ") {
    global $con;
    @mysql_error($con);
    @mysql_close($con);
    $_SESSION['MESSAGE'] = $msg;
}

function AttemptLogin() {
    if ( count($_POST) > 0 ) {
        $con = mysql_connect(host, user, pass);
        if (! $con) {
            MyWayToDie("Can't connect to database server!");
            return;
        }

        if (! mysql_select_db(base) ) {
            MyWayToDie("Can't select the database!");
            return;
        }
    }
}
```

```
$username = $_POST['username'];
$password = $_POST['password'];
$password = hash('sha512', $password);

$_GET['password'] =
'*****';

unset($_POST);

$username = mysql_real_escape_string($username, $con);
$password = mysql_real_escape_string($password, $con);

$SQL = "SELECT `id` FROM `users` WHERE `username` =
'{$username}' and `token` = '{$password}'";
$res = mysql_query($SQL, $con);
if ($res and mysql_num_rows($res)) {
    $data = mysql_fetch_assoc($res);
    $user_id = $data['id'];
    if ( $user_id != "" ) {
        $SQL = "UPDATE `users` SET `token` = NULL
WHERE `id` = {$user_id}";

        mysql_query($SQL, $con);
        $_SESSION['USER'] = $user_id;

        // Get user agent
        $agent = $_SERVER['HTTP_USER_AGENT'];

        // Get IP
        $ip = $_SERVER['REMOTE_ADDR'];

        // Get the active session id
        $phpsessionid = session_id();
        $phpsessionid = mysql_real_escape_string(
$phpsessionid, $con );
```

```
$SQL = "SELECT agent, ip FROM logins WHERE
user_id = '{$user_id}' AND phpsessionid = '{$phpsessionid}';";
$res2 = mysql_query($SQL, $con);
if ($res2 and mysql_num_rows($res2)) {
    $data2 = mysql_fetch_assoc($res2);
    $was_agent = $data2['agent'];
    $was_ip = $data2['ip'];

    // Ako se ne poklapa user agent i IP sa prethodno
    koriscenim za ovu trenutnu vrednost PHP session ID-a, kraj krajeva za tu sesiju!
    if ( $agent != $was_agent || $ip != $was_ip ) {
        ob_clean();
        unset($_SESSION);
        session_destroy();
        header('Location: index.php');
        die();
    }
}

// Get mllic is exists, or generate a new one if it doesn't
or if it is invalid

if ( !isset($_COOKIE['mllic']) ) {
    $mllic = rand(100000, 999999);
} else {
    $mllic = $_COOKIE['mllic'];
    if ( !is_numeric($mllic) || strlen($mllic) != 6 ) {
        $mllic = rand(100000, 999999);
    }
}

// Store current My Last Login Identification Code mllic
cookie

setcookie('mllic', $mllic, time() + 24*60*60 );

// Prepare data for insertion into the database
```

```

                                $agent      = mysql_real_escape_string( $agent,
$con );

                                $ip         = mysql_real_escape_string( $ip, $con
);

                                $mllic      = mysql_real_escape_string( $mllic,
$con );

                                // Insert new login data
                                $SQL = "INSERT INTO logins (user_id, ip, agent,
cookie, phpsessionid) VALUES ('{$user_id}', '{$ip}', '{$agent}', '{$mllic}',
'{$phpsessionid}')";

                                mysql_query($SQL, $con);

                                ob_clean();
                                header('Location: panel.php');
                                die();
                                return;
                        } else {
                                MyWayToDie("Login Failed!");
                                return;
                        }
                                mysql_free_result($res);
                } else {
                                MyWayToDie("Invalid user or password!");
                                return;
                }
        }
        AttemptLogin();
?>
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
```

```
<title>Sign in &middot; Marko Sarac &middot; Secure Login App</title>
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<meta name="description" content="">
<meta name="author" content="Milan Tair, Marko Sarac">
<link href="assets/css/bootstrap.min.css" rel="stylesheet">
<style type="text/css">
body {
  padding-top: 40px;
  padding-bottom: 40px;
  background-color: #f5f5f5;
}

.form-signin {
  max-width: 300px;
  padding: 19px 29px 29px;
  margin: 0 auto 20px;
  background-color: #fff;
  border: 1px solid #e5e5e5;
  -webkit-border-radius: 5px;
  -moz-border-radius: 5px;
  border-radius: 5px;
  -webkit-box-shadow: 0 1px 2px rgba(0,0,0,.05);
  -moz-box-shadow: 0 1px 2px rgba(0,0,0,.05);
  box-shadow: 0 1px 2px rgba(0,0,0,.05);
}

.form-signin .form-signin-heading,
.form-signin .checkbox {
  margin-bottom: 10px;
}

.form-signin input[type="text"],
.form-signin input[type="password"] {
  font-size: 16px;
  height: auto;
  margin-bottom: 15px;
```

```
padding: 7px 9px;
}
</style>
<link href="assets/css/bootstrap-responsive.min.css" rel="stylesheet">
</head>
<body>
<div class="container">
  <form class="form-signin" action="" method="post">
    <h2 class="form-signin-heading">Please sign in</h2>
    <input type="text" class="input-block-level" placeholder="Username"
name="username" required="required" title="You must enter a valid username!" />
    <input type="password" class="input-block-level" placeholder="Password"
name="password" required="required" title="You must enter a valid password!" />
    <button class="btn btn-large btn-primary" type="submit">Sign in</button>
  </form>
  <?php if ( $_SESSION['MESSAGE'] != "" ): ?>
    <div class="form-signin form-signin-heading">
      <h4><?php echo $_SESSION['MESSAGE']; ?></h4>
      <?php unset( $_SESSION['MESSAGE'] ); ?>
    </div>
  <?php endif; ?>
</div>
</body>
</html>
```

6.2.4. [New_token.php](#)

Osnovna php skripta zadužena za generisanje tokena. Od aplikacije preuzima zahtev za generisanje tokena praćen enkodiranim korisničkim imenom i lozinkom. Skripta dekodira korisničko ime i lozinku, proverava u bazi podataka da li postoje kao par i ukoliko postoje generiše novi token. Generisani token se upisuje u tabelu tokena a svi prethodni se poništavaju. Token se enkodira i vraća aplikaciji za prikaz. Ukoliko korisničko ime i lozinka ne postoje kao par u bazi podataka, skripta enkodira poruku o grešci koju će aplikacija korisniku prikazati.

```
<?php
    include 'config.php';

    if ( count($_POST) > 0 ) {
        function MyWayToDie($msg = ") {
            global $con;
            @mysql_close($con);
            die($msg);
        }

        function hexToStr($hex) {
            $string="";
            for ($i=0; $i < strlen($hex)-1; $i+=2) {
                $string .= chr(hexdec($hex[$i].$hex[$i+1]));
            }
            return $string;
        }

        $con = mysql_connect(host, user, pass) or MyWayToDie("Can't connect to database
server!");
        mysql_select_db(base) or MyWayToDie("Can't select the database!");

        $username = $_POST['username'];
        $password = $_POST['password'];
```

```
$username = hexToStr($username);
$password = hexToStr($password);

$username_4_sql = mysql_real_escape_string($username, $con);

$SQL = "SELECT `pin` FROM `users` WHERE `username_xor_pin` =
'{$username_4_sql}' AND `active` = 1;";
$res = mysql_query($SQL, $con);
if ($res and mysql_num_rows($res)) {
    $data = mysql_fetch_assoc($res);
    $user_pin = $data['pin'];
    mysql_free_result($res);
} else MyWayToDie("Couldn't get user data!");

$real_username = "";
$real_password = "";
$user_pin = $user_pin . $user_pin . $user_pin . $user_pin . $user_pin;
for ($i=0; $i<strlen($username);$i++) {
    $real_username .= $username[$i] ^ $user_pin[$i];
    $real_password .= $password[$i] ^ $user_pin[$i];
}

$username = $real_username;
$password = $real_password;

$_POST['password'] = '*****';
unset($_POST);

$password_hash = hash('sha512', $password);

$username = mysql_real_escape_string($username, $con);
$password_hash = mysql_real_escape_string($password_hash, $con);

$user_id = NULL;
```

```
$SQL = "SELECT `id` FROM `users` WHERE `username` = '{$_username}' and
`password` = '{$_password_hash}' and `active` = 1;";

$res = mysql_query($SQL, $con);
if ($res and mysql_num_rows($res)) {
    $data = mysql_fetch_assoc($res);
    $user_id = $data['id'];
    mysql_free_result($res);
} else MyWayToDie("Couldn't get user data!");

if ( $user_id == NULL ) MyWayToDie("User not found!");

$token = strval(rand(1000, 9999));
$password_and_token = $password . $token;
$password_and_token = hash('sha512', $password_and_token);

$password      = '*****';
$password_hash = '*****';
unset($password);

$SQL = "INSERT INTO `tokens` (`password_with_token`, `user_id`) VALUES
('{$_password_and_token}', {$_user_id});";
$res = mysql_query($SQL, $con);
if ( $res ) {
    $SQL = "UPDATE `users` SET `token` = '{$_password_and_token}'
WHERE `id` = {$_user_id}";
    $res = mysql_query($SQL, $con);
    mysql_close( $con );
    MyWayToDie( $token );
} else MyWayToDie("Couldn't store new token!");
}
```

6.2.5. [Panel.php](#)

Skripta koja generiše HTML stranicu kada se korisnik uspešno prijavi. Stranica se stilizuje korišćenjem bootstrapa. Ima ulogu da omogući odjavljivanje korisnika.

```
<?php
    session_start();

    if ( isset($_GET['logout']) ) {
        unset($_SESSION);
        session_destroy();
    }

    if (!isset($_SESSION['USER'])) {
        ob_clean();
        header('Location: index.php');
        die();
    }

?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>Marko Sarac &middot; Secure Login App</title>
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta name="description" content="">
    <meta name="author" content="">

    <!-- Le styles -->
    <link href="assets/css/bootstrap.min.css" rel="stylesheet">
    <style type="text/css">
        body {
```

```
padding-top: 20px;
padding-bottom: 40px;
}

/* Custom container */
.container-narrow {
    margin: 0 auto;
    max-width: 700px;
}
.container-narrow > hr {
    margin: 30px 0;
}

/* Main marketing message and sign up button */
.jumbotron {
    margin: 60px 0;
    text-align: center;
}
.jumbotron h1 {
    font-size: 72px;
    line-height: 1;
}
.jumbotron .btn {
    font-size: 21px;
    padding: 14px 24px;
}

/* Supporting marketing content */
.marketing {
    margin: 60px 0;
}
.marketing p + h4 {
    margin-top: 28px;
}
```

```
</style>
<link href="assets/css/bootstrap-responsive.min.css" rel="stylesheet">
</head>
<body>
<div class="container-narrow">
  <div class="masthead">
    <ul class="nav nav-pills pull-right">
      <li class="active"><a href="#">Home</a></li>
      <li><a href="#">About</a></li>
      <li><a href="?logout">Log out</a></li>
    </ul>
    <h3 class="muted">Marko Sarac - Secure Login App</h3>
  </div>

  <hr>

  <div class="jumbotron">
    <h1>Super awesome Secure Login App!</h1>
    <p class="lead">Cras justo odio, dapibus ac facilisis in, egestas eget quam. Fusce
dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa
justo sit amet risus.</p>
    <a class="btn btn-large btn-success" href="#">Sign up today</a>
  </div>

  <hr>

  <div class="row-fluid marketing">
    <div class="span6">
      <h4>Subheading</h4>
      <p>Donec id elit non mi porta gravida at eget metus. Maecenas faucibus mollis
interdum.</p>
      <h4>Subheading</h4>
      <p>Morbi leo risus, porta ac consectetur ac, vestibulum at eros. Cras mattis consectetur
purus sit amet fermentum.</p>
```

```
</div>
<div class="span6">
  <h4>Subheading</h4>
  <p>Donec id elit non mi porta gravida at eget metus. Maecenas faucibus mollis
interdum.</p>
  <h4>Subheading</h4>
  <p>Morbi leo risus, porta ac consectetur ac, vestibulum at eros. Cras mattis consectetur
purus sit amet fermentum.</p>
</div>
</div>
<hr>
<div class="footer">
  <p>&copy; Company <?php echo date('Y'); ?></p>
</div>
</div>
</body>
</html>
```

6.3. Prilog 3. MySQL upiti korišćeni na strani servera

MySQL upiti za kreiranje potrebnih tabela na serverskoj strani i popunjavanje određenih tabela podacima potrebnim za demonstraciju su:

```
SET FOREIGN_KEY_CHECKS=0;
```

```
DROP TABLE IF EXISTS `logins`;
```

```
CREATE TABLE `logins` (
```

```
  `id` int(15) NOT NULL AUTO_INCREMENT,
```

```
  `user_id` int(7) NOT NULL,
```

```
  `timestamp` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
```

```
  `ip` varchar(31) COLLATE utf8_unicode_ci NOT NULL,
```

```
  `agent` varchar(255) COLLATE utf8_unicode_ci NOT NULL,
```

```
  `cookie` varchar(10) COLLATE utf8_unicode_ci NOT NULL,
```

```
  `phpsessionid` varchar(255) COLLATE utf8_unicode_ci NOT NULL,
```

```
  PRIMARY KEY (`id`),
```

```
  KEY `logins_user_id_users_id` (`user_id`)
```

```
) ENGINE=InnoDB AUTO_INCREMENT=7 DEFAULT CHARSET=utf8
```

```
COLLATE=utf8_unicode_ci;
```

```
DROP TABLE IF EXISTS `tokens`;
```

```
CREATE TABLE `tokens` (
```

```
  `id` int(15) NOT NULL AUTO_INCREMENT,
```

```
  `user_id` int(7) NOT NULL,
```

```
  `password_with_token` varchar(255) CHARACTER SET ascii NOT NULL,
```

```
  `datetime` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
```

```
  PRIMARY KEY (`id`),
```

```
  KEY `tokens_user_id_users_id` (`user_id`),
```

```
  CONSTRAINT `tokens_user_id_users_id` FOREIGN KEY (`user_id`) REFERENCES
```

```
`users` (`id`) ON UPDATE CASCADE
```

```
) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT CHARSET=utf8
```

```
COLLATE=utf8_unicode_ci;
```

```
DROP TABLE IF EXISTS `users`;
```

```
CREATE TABLE `users` (
```

```
`id` int(7) NOT NULL AUTO_INCREMENT,  
`username` varchar(255) COLLATE utf8_unicode_ci NOT NULL,  
`username_xor_pin` varchar(255) COLLATE utf8_unicode_ci NOT NULL,  
`password` varchar(255) CHARACTER SET ascii NOT NULL,  
`active` tinyint(1) NOT NULL DEFAULT '1',  
`token` varchar(255) CHARACTER SET ascii DEFAULT NULL,  
`pin` int(6) NOT NULL,  
`activation_code` varchar(9) COLLATE utf8_unicode_ci DEFAULT NULL,  
PRIMARY KEY (`id`)  
) ENGINE=InnoDB AUTO_INCREMENT=3 DEFAULT CHARSET=utf8  
COLLATE=utf8_unicode_ci;  
INSERT INTO `users` VALUES ('1', 'msarac', '_CQGQS',  
'c7ad44cbad762a5da0a452f9e854fdc1e0e7a52a38015f23f3eab1d80b931dd472634dfac71cd3  
4ebc35d16ab7fb8a90c81f975113d6c7538dc69dd8de9077ec', '1', null, '200500', '6987-8074');
```

7. BIBLIOGRAFIJA

- [1] Uvod u računarske mreže, Prof. dr Mladen Veinović, Aleksandar Jevremović, Univerzitet Singidunum, Beograd
- [2] Kriptologija I, Osnove za analizu i sintezu šifarskih sistema, Prof. dr Mladen Veinović, Saša Adamović, Univerzitet Singidunum, Beograd
- [3] R.J. Anderson, Security Engineering: A Guide to Building Dependable Distributed Systems. New York: Wiley, 2001.
- [4] Michael Howard, "Man-in-the-Middle Attack to the HTTPS Protocol", IEEE computer society, 2009, p.78-81.
- [5] S.Thomas. SSL and TLS Essentials. New york: Wiley Computing Publishing, 2004.
- [6] Moxie, Sslstrip.<http://www.thoughtcrime.org/software/sslstrip/>
- [7] T. Dierks and C. Allen, The TLS Protocol, IETF RFC 2246, 1999; www.ietf.org/rfc/rfc2246.txt.
- [8] Ponemon Institute© Research Report, Cost of Cyber Crime Study, August 2012
- [9] X. Song, H. Deng, "Taking Flexible and Diverse Approaches to Get Undergraduate Students Interested in Cryptography Course", Proceedings of the First International Workshop on Education Technology and Computer Science, 2009.
- [10] Certified Wireless Analysis Professional Official Study Guide (Exam PW0-205) bPlanet3 Wireless (Paperback - Oct 15, 2004)
- [11] Wilson, J. "The Future of the Firewall." Business Communications Review, May 2005.

- [12] 802.11 Wireless Networks: Security and Analysis (Computer Communications and Networks) by Alan Holt and Chi-Yu Huang (Hardcover - Jul 8, 2010)
- [13] Wireless Network Security (Signals and Communication Technology) by Yang Xiao, Xuemin Shen, and Ding-Zhu Du (Nov 23, 2010)
- [14] Mobile and Wireless Network Security and Privacy by S. Kami Makki, Peter Reiher, Kia Makki, and Niki Pissinou (Nov 4, 2010)
- [15] http://www.intechopen.com/source/pdfs/12460/InTech-Distributed_detection_of_node_capture_attacks_in_wireless_sensor_networks.pdf
- [16] Security and Cooperation in Wireless Networks: Thwarting Malicious and Selfish Behavior in the Age of Ubiquitous Computing by Levente Buttyan and Jean-Pierre Hubaux (Aug 1, 2007)
- [17] Network Security Hacks: Tips & Tools for Protecting Your Privacy by Andrew Lockhart (Oct 30, 2006)
- [18] Hacking Wireless Networks (Computer/Tech) by Kevin Beaver, Peter T. Davis, and Devin K. Akin (Sep 13, 2005)
- [19] Hacking Exposed Wireless, Second Edition by Johnny Cache, Joshua Wright, and Vincent Liu (Jul 9, 2010)
- [20] <http://searchsecurity.techtarget.com/definition/RSA>
- [21] Phishing and Countermeasures Understanding the Increasing Problem of Electronic Identity Theft
- [22] <http://www.ietf.org/rfc/rfc2986.txt>
- [23] Wireless Hacks: Tips & Tools for Building, Extending, and Securing Your Network by Rob Flickenger and Roger Weeks (Nov 22, 2005)
- [24] Security+ Guide to Network Security Fundamentals by Mark D. Ciampa (Nov 11, 2008)

- [25] W. Stallings, Cryptography and Network Security: Principles and Practice (5th Edition), 2010.
- [26] Audio and video conferencing tools in learning management systems, G. Adorni, M. Coccoli, C. Fadda, IEEE conferencing, 2007.
- [27] Implementing 802.1X Security Solutions for Wired and Wireless Networks by Jim Geier (Apr 14, 2008)
- [28] Network Security Essentials: Applications And Standards Fourth Edition, William Stallings, 2011 Pearson Education, Inc.
- [29] Hacking Exposed: Network Security Secrets and Solutions, Sixth Edition by Stuart McClure, Joel Scambray, and George Kurtz (Jan 5, 2009)
- [30] Collaborative Technologies in International Distance Education, R. Anderson, Ruth Anderson, Natalie Linnell, IEEE conferencing, 2009.
- [31] Smart Classrooms Bring Top-Quality Education Around The Globe, Davar Pishva, IEEE conferencing, 2007.
- [32] Video conferencing over IP; configure, secure, and troubleshoot. Gough, Michael
- [33] Publishing, V. H. 2008. IT Governance based on Cobit 4.1 - A Management Guide. Van Haran Publishing.
- [34] INFORMATION SECURITY Principles and Practice Second Edition, Mark Stamp, San Jose State University, San Jose, CA, A JOHN WILEY & SONS, INC., PUBLICATION
- [35] Selig, G. J. 2008. Implementing IT Governance. Van Haren Publishing.
- [36] Senft, S., & Gallegos, F. 2009. Information Technology Control and Audit (Third ed.). Boca Raton, USA: Taylor & Francis Group.
- [37] B. W. Lampson, Computer security in the real world, IEEE Computer, pp. 37-46, June 2004.
- [38] Goeken M., Alter S. 2008. IT governance framework as methods, ICEIS 2008.

- [39] Min, Y. W. 2009. Understanding and Auditing IT Systems. Peking: Lulu.
- [40] 802.11 Wireless Networks: Security and Analysis (Computer Communications and Networks) by Alan Holt and Chi-Yu Huang (Jul 8, 2010)
- [41] D. Lin, Hunting for undetectable metamorphic viruses, Master's Thesis, Department of Computer Science, San Jose State University, 2010, www.cs.sjsu.edu/faculty/stamp/students/lin_da.pdf
- [42] Implementing 802.1X Security Solutions for Wired and Wireless Networks by Jim Geier (Apr 14, 2008)
- [43] Okrugli sto „Digitalna forenzika i IT veštacenje u funkciji borbe protiv visoko tehnološkog kriminala“ - Wireless forenzika
- [44] Hacking Exposed: Network Security Secrets and Solutions, Sixth Edition by Stuart McClure, Joel Scambray, and George Kurtz (Jan 5, 2009)
- [45] Security and Cooperation in Wireless Networks: Thwarting Malicious and Selfish Behavior in the Age of Ubiquitous Computing by Levente Buttyan and Jean-Pierre Hubaux (Aug 1, 2007)
- [46] Network Security Hacks: Tips & Tools for Protecting Your Privacy by Andrew Lockhart (Oct 30, 2006)
- [47] Maximum Wireless Security - Paperback (Dec. 18, 2002) by Cyrus Peikari and Seth Fogie
- [48] Mansoor Alicherry and Angelos D. Keromytis. Doublecheck: Multi-path verication against man-in-the-middle attacks. In ISCC 2009: IEEE Symposium on Computers and Communications, pages 557-563, Piscataway, NJ, USA, 2009. IEEE
- [49] Christopher Soghoian. Caught in the cloud: Privacy, encryption, and government back doors in the web 2.0 era. In Journal on Telecommunications and High Technology Law, Forthcoming.

- [50] Christopher Soghoian. 8 Million Reasons for Real Surveillance Oversight. Slight Paranoia blog, December 1 2009. paranoia.dubfire.net/2009/12/8-million-reasons-for-real-surveillance.html.
- [51] Ian Grigg. Why the browsers must change their old SSL security model. Financial Cryptography (blog), March 24 2010. financialcryptography.com/mt/archives/001232.html.
- [52] Pavni Diwanji. Detecting suspicious account activity. The Official Gmail Blog, March 24 2010. gmailblog.blogspot.com/2010/03/detecting-suspicious-account-activity.html.
- [53] Matthieu Bussiere and Marcel Fratzscher. Low probability, high impact: Policy making and extreme events. Journal of Policy Modeling, 2008.
- [54] Hacking Wireless Networks (Computer/Tech) by Kevin Beaver, Peter T. Davis, and Devin K. Akin (Sep 13, 2005)
- [55] Essential Computer Security: Everyone's Guide to Email, Internet, and Wireless Security - Kindle Edition - Kindle eBook (Jan. 14, 2007) by T. Bradley
- [56] Hacking Exposed Wireless, Second Edition by Johnny Cache, Joshua Wright, and Vincent Liu (Jul 9, 2010)
- [57] Wireless Hacks: Tips & Tools for Building, Extending, and Securing Your Network by Rob Flickenger and Roger Weeks (Nov 22, 2005)
- [58] Security+ Guide to Network Security Fundamentals by Mark D. Ciampa (Nov 11, 2008).
- [59] T. Cipresso, Software Reverse Engineering Education, Master's Thesis, Department of Computer Science, San Jose State University, 2009, reversingproject.info
- [60] AS8015-2005 – "Australian Standard for Corporate Governance of Information and Communication Technology (ICT)", 2005.
- [61] E. Barkan, E. Biham, and N. Keller, Instant ciphertext-only cryptanalysis of GSM encrypted communication, cryptome.org/gsm-crack-bbk.pdf

- [62] M. Bishop, Computer Security: Art and Science, Addison Wesley, 2003.
- [63] M. Cooney, IBM touts encryption innovation: New technology performs calculations on encrypted data without decrypting it, Computer World, June 25, 2009.
- [64] Xiaoyun Wang and Hongbo Yu, "How to break MD5 and other hash functions", Advances in Cryptology – EUROCRYPT, LNCS 3494, Springer-Verlag, pp.19-35, 2005.
- [65] N. Borisov, I. Goldberg, and D. Wagner, Intercepting mobile communications: The insecurity of 802.11, www.isaac.cs.berkeley.edu/isaac/wep-draft.pdf
- [66] U. Erlingsson, Y. Younan, and F. Piessens, Low-level Software Security by Example, to appear in Handbook of Communications Security, Springer-Verlag, 2009.
- [67] W. Feller, An Introduction to Probability Theory and Its Applications, third edition, Wiley, 1968.
- [68] E. Mills, Twitter, Facebook attack targeted one user, CNET News, news.cnet.com/8301-27080_3-10305200-245.html
- [69] P. Priyadarshini and M. Stamp, Digital rights management for untrusted peer-to-peer networks, Handbook of Research on Secure Multimedia Distribution, IGI Global, March 2009, www.es.sj.su.edu/faculty/stamp/papers/Pallavi_paper.doc
- [70] Sheena Mathew, K.Paulose Jacob, "A New Fast Stream Cipher: MAJE4", Proceedings of IEEE, INDICON 2005, pp60-63, 2005.
- [71] M. Stamp, Digital rights management: the technology behind the hype, Journal of Electronic Commerce Research, Vol. 4, No. 3, 2003, www.csulb.edu/web/journals/jecr/issues/20033/paper3.pdf
- [72] D. Mulani, How smart is your Android smartphone?, Master's Thesis, Department of Computer Science, San Jose State University, 2010, www.es.sjsu.edu/faculty/stamp/students/mulani_deepika.pdf

- [73] D. Jao, Elliptic curve cryptography, in Handbook of Communication and Information Security, Springer-Verlag, 2009.
- [74] P. Cerven, Crackproof Your Software: Protect Your Software Against Crackers, No Starch Press, 2002.
- [75] H. Krawczyk, M. Bellare and R. Canetti, RFC 2104 — HMAC: Keyedhashing for message authentication, www.faqs.org/rfcs/rfc2104.html
- [76] Beale Screamer, Microsoft's digital rights management scheme technical details, web.elastic.org/~fche/mirrors/cryptome.org/beale-sci-crypt.html
- [77] National Institute of Standards and Technology (NIST) (2002), FIPS- 180-2: Secure Hash Standard, at http://csrc.nist.gov/publications/fips/fips_180-2/fips_180-2.pdf
- [78] Sheena Mathew, K. Poulouse Jacob, “JERIM-320: A New 320-bit Hash Function with Higher Security”, International Journal of Computers, Systems and Signals, to be published.
- [79] A.Roos, “A Class of weak keys in the RC4 stream cipher”, sci.crypt, 1995.
- [80] D.Wagner, “ My RC4 weak keys”, sci.crypt, September 1995.
- [81] M. Stamp, Risks of digital rights management, Inside Risks 147, Communications of the ACM, Vol. 45, No. 9, p. 120, September 2002, www.csl.sri.com/users/neumann/insiderisks.html#147
- [82] B. A. Forouzan, TCP/IP Protocol Suite, second edition, McGraw Hill, 2003.
- [83] A.I.Grosul and D.S.Wallach, “A Related Key Cryptanalysis of RC4”, Manuscript from Department of Computer Science, Rice University, 6. June 2000.
- [84] Johnathan Nightingale. SSL Question Corner. meandering wildly (blog), August 5 2008. blog.johnath.com/2008/08/05/ssl-question-corner/.
- [85] DI Management, RSA Algorithm, www.dimgt.com.au/rsa_alg.html#pkcslschemes

- [86] Defcon 11, at www.cs.sjsu.edu/~stamp/cv/tripreports/defcon11.html
- [87] Defcon 16, at www.defcon.org/html/defcon-16/dc-16-post.html
- [88] J.Dj.Golic, "Linear statistical Weakness of alleged RC4 keystream generator", *Advances in Cryptology – Eurocrypt 97*, LNCS vol. 1233, pp.226-238, Springer-Verlag, 1997.
- [89] S.R.Fuhrer and D.A.McGrew, "Statistical Analysis of the Alleged RC4 Keystream Generator", *Proceedings of Fast Software Encryption 2000*, LNCS vol. 1978, pp.19-30, Springer-Verlag, 2001.
- [90] C. Swenson, *Modern Cryptanalysis: Techniques for Advanced Code Breaking*, Wiley, 2008.
- [91] J. Zhang, *Improved software activation using multithreading*, Master's Thesis, Department of Computer Science, San Jose State University, 2010, www.cs.sjsu.edu/faculty/stamp/students/zhang_jianrui.pdf
- [92] T. Ylonen, *The Secure Shell (SSH) Transport Layer Protocol*, RFC 4253, at www.ietf.org/rfc/rfc4253.txt
- [93] R. Venkataramu and M. Stamp, *P2PTunes: A peer-to-peer digital rights management system*, *Handbook of Research on Secure Multimedia Distribution*, IGI Global, March 2009, www.cs.sjsu.edu/faculty/stamp/papers/Ramya_paper.doc
- [94] Marc Stevens, Alexander Sotirov, Jacob Appelbaum, Arjen Lenstra, David Molnar, Dag Arne Osvik, and Benne Weger. Short chosen-prefix collisions for MD5 and the creation of a rogue CA certificate. In *Proceedings of the 29th Annual International Cryptology Conference on Advances in Cryptology*, pages 55-69, Berlin, Heidelberg, 2009. Springer-Verlag.
- [95] Lucas I. Password recovery speeds. Lockdown.co.uk. <http://www.lockdown.co.uk/?pg=combi>; 2009.
- [96] Dictionary.com. Obfuscate. Dictionary.com. <http://dictionary.reference.com/browse/obfuscate>; 2010.

- [97] Hadnagy, CJ, Aharoni M, and O’Gorman J. Defcon 18 social engineering CTF – how strong is your schmooze. socialengineer.org. http://www.social-engineer.org/resources/sectf/Social-Engineer_CTF_Report.pdf;
- [98] Laboratory for Systems and Signals. Nessus/OpenVAS comparison test. Faculty of electrical engineering and computing, University of Zagreb. http://www.securisite.org/biblioteca/Fran/documentacion%20curso%20seg/3/Nessus_vs_OpenVAS_en.pdf; 2009.
- [99] S.Mister and S.E.Tavares, “Cryptanalysis of RC4-like Ciphers”, Proceedings of SAC’98, LNCS vol. 1556, pp.131-143, Springer-V0000erlag, 1999.
- [100] L.Knudsen, W.Meier, B.Preneel, V.Rijmen and S.Verdoolaege, “Analysis methods for (alleged) RC4”, Advances in Cryptology – AsiaCrypt 98, LNCS vol.1514, pp.327-341, Springer -Verlag, 1998.
- [101] Mills E. Stuxnet: Fact vs. theory. CNET news. http://news.cnet.com/8301-27080_3-20018530-245.html; 2010
- [102] J. A. Halderman, B.Waters, and E. Felten A convenient method for securely managing passwords, 14th International World Wide Web Conference, 2005.
- [103] Yanjiang Yang, Robert H. Deng, and Feng Bao, “A Practical Password-Based Two Server Authentication and Key Exchange System,” IEEE Transaction on Secure and Dependable Computing, Vol.3, No.2, April-June 2006
- [104] Alexander Sotirov and Mike Zusman. Breaking the Security Myths of Extended Validation SSL Certificates. BlackHat USA, 2009. www.blackhat.com/presentations/bh-usa-09/SOTIROV/BHUSA09-otirov-AttackExtSSL-SLIDES.pdf.
- [105] What are Information Operations. Cyberspace and information operations study center. July 24, 2010, <http://www.au.af.mil/info-ops/what.htm>; 2010
- [106] Muxiang Zhang, Analysis of the SPEKE password-authenticated key exchange protocol, IEEE Communications Letters, Vol. 8, No. 1, pp. 63-65, January 2004.

- [107] Singel R. Top secret: we're wiretapping you. Wired.com. March 05, 2007, <http://www.wired.com/science/discoveries/news/2007/03/72811?currentPage=all>; 2007.
- [108] Z. Zhao, Z. Dong, Y. Wang, "Security analysis of a password-based authentication protocol proposed to IEEE 1363," Theoretical Computer Science, Vol. 352, No. 1, pp.280–287, 2006.

8. OBJAVLJENI RADOVI U NAUČNIM ČASOPISIMA I NA KONFERENCIJAMA

Rezultati rada na ovoj doktorskoj disertaciji objavljeni su u više radova u časopisima međunarodnog značaja i saopšteni na više naučnih skupova u zemlji i inostranstvu. Pojedine ideje su već izložene na međunarodnim stručnim skupovima.

Kao autor ili koautor objavio je 39 radova na domaćim i međunarodnim konferencijama i u časopisima.

Radovi u međunarodnim časopisima:

- [1.] M. Šarac, T. Radojevic, N. Stanišić, S. Adamović, D. Radovanovic, „Safety of e-business Applications in Serbia: Applied Knowledge Based on SSL Traffic“ JIBC // Journal of Internet Banking and Commerce. - December 2012, vol. 17, no.3.
- [2.] D. Radovanovic, T. Radojević, S. Adamović, M. Šarac, „Information technology audit process and cobit“ MI // Metalurgia International, ISBN 1582-2214, Romania, 2013.
- [3.] S. Adamović, M. Šarac, M. Veinović, M. Milosavljević, A. Jevremović, „An Interactive and Collaborative Approach to Teaching Cryptology“ IFETS // Journal of Educational Technology & Society, 1436-452

Udžbenici:

- [4.] Univerzitetski udžbenik: „Operativni sistemi“, Ranko Popović, Irina Branović, Marko Šarac, ISBN: 978-86-7912-377-0, Beograd, Univerzitet Singidunum 2011.

Saopštenja sa domaćeg-međunarodnog skupa – konferencije:

- [5.] Sigurnosna i bezbednosna rešenja u mrežnom okruženju virtualnog datacentra, 16th Telecommunications Forum - TELFOR 2008: Šarac Marko, Veinović Mladen.
- [6.] Komparacija i analiza resursa fizičkog i virtualnog okruženja datacentra, rad objavljen u zborniku radova Sinergija 2009, Bijeljina, BiH. Marko Šarac, Mladen Veinović, Saša Adamović.
- [7.] Zaštita u bazama podataka, rad objavljen u zborniku radova Sinergija 2009, Bijeljina, BiH, Marko Šarac, Mladen Veinović, Saša Adamović.
- [8.] Analiza performansi i bezbednosti fizičkog i virtuelnog datacentra, rad objavljen u stručnom časopisu Singidunum Revija 2009, Beograd. Mladen Veinović, Marko Šarac.
- [9.] Klasifikacija napada i ugrožavanje bezbednosti smart kartica, rad objavljen u stručnom časopisu Singidunum Revija 2009, Beograd. Mladen Veinović, Dalibor Radovanović, Marko Šarac.
- [10.] Analysis and classification of attacks on the security of smart cards with additional emphasis on security pin code, Međunarodna konferencija, Bugarska - Gabrovo - Unitech09, Dalibor Radovanović, Marko Šarac, Saša Adamović, Miloš Milenković.
- [11.] Data protection in databases at the local and server level, Međunarodna konferencija, Bugarska - Gabrovo - Unitech09, Saša Adamović, Marko Šarac, Dalibor Radovanović, Miloš Milenković.
- [12.] Upravljanje X.509 sertifikatima u PKI sistemu, ZITEH, Beograd, Miloš Milenković, Saša Adamović, Marko Šarac, Dalibor Radovanović.
- [13.] Analiza bezbednosnih i sigurnosnih rešenja u mrežnom okruženju virtelnih datacentara, Infotech 2010, Jahorina, Marko Šarac, Dalibor Radovanović, Saša Adamović, Tijana Radojević.

- [14.] IT audit in accordance with Cobit standard – IEEE Conference, MIPRO 2010, May 24-28, 2010, Opatija, Croatia, ISBN 978-1-4244-7763-0, p1137-1141, Radovanović D. Radojević T, Lučić D, Šarac M.
- [15.] Analiza metoda za kontrolu i reviziju informacionih sistema, Infotech 2010, Jahorina , Dalibor Radovanović, Marko Šarac, Dubravka Lučić, Saša Adamović.
- [16.] Generatori slučajnih sekvenci i njihov uticaj na sigurnost, Infotech 2010, Jahorina, Saša Adamović, Marko Šarac, Dalibor Radovanović, Miloš Milenković.
- [17.] Analiza procesa virtualizacije, Sinergija 2010, 19 mart 2010, Bijeljina, Bosna, p. 131-137, Veinović M., Šarac M., Radovanović D.
- [18.] Analysis of Security and Safety Solutions in Network Environment of Virtual Datacenter, Međunarodna konferencija, Litvanija - Kaunas & Vilnius - ELECTRONICS 2010, M. Šarac, D. Radovanović, S. Adamović, M. Milenković, D. Lučić.
- [19.] Smart Card Security, Međunarodna konferencija, Litvanija - Kaunas & Vilnius - ELECTRONICS 2010, D. Radovanovic, M. Sarac, S. Adamovic, D. Lucic, J. Rakovic.
- [20.] Secret Key Agreement and Privacy Amplification by Public Discussion, Litvanija - Kaunas & Vilnius - ELECTRONICS 2010, S. Adamović, M. Šarac, D. Radovanović, M. Milenković.
- [21.] Upravljanje informacionim tehnologijama i IT revizija, YU INFO 2010, Mart 7-10, 2010, Kopaonik, Srbija, Stanišić M., Radovanović D., Lučić D., Šarac M.
- [22.] Rešavanje problema uskog grla virtuelnih servera, virtuelnim iSCSI uređajima za skladištenje podataka, Singidunum revija, Vol. 6, No. 2, Beograd, 2010, ISSN 1820-8819, 193-198., Radovanović D., Šarac M., Adamović S.
- [23.] Analysis of methodology for it governance and information systems audit - 6th International Scientific Conference Business and Management–2010 May

- 13-14, 2010, Vilnius, Lithuania, ISBN 2029-4441, p.943-949, Radovanović D., Radojević T., Lučić D., Šarac M.
- [24.] Integracija Video konferencijskih sistema u edukativne platforme, SEE Business Travel & Meetings magazine, Beograd, 2011., M. Šarac, V. Mihailović.
- [25.] Analiza sigurnosti bežičnih mreža IEEE 802.11. na teritoriji grada Beograda, Infotech 2011, Jahorina, M. Šarac, S. Adamović, D. Radovanović.
- [26.] Video konferencijski sistemi i njihova primena u obrazovnom sistemu, rad objavljen u zborniku radova Sinergija 2011, Bijeljina, BiH, V. Mihailović, M. Šarac, S. Adamović, D. Radovanović.
- [27.] Klasifikacija i analiza napada na sigurnost mikroprocesorskih smart kartica na fizičkom nivou, sa posebnim osvrtom na verovatnoću pogađanja PIN koda, Telfor 2009, Novembar 24-26, 2009, Beograd, Srbija, CD Zbornik, 1237-1240, Radovanović D., Lučić D., Šarac M., Milenković M.
- [28.] Realizacija učenja na daljinu na Univerzitetu Singidunum, rad objavljen u zborniku radova Etran - Društvo za elektroniku, telekomunikacije, računarstvo, automatiku i nuklearnu tehniku, J. Gavrilović, M. Šarac, I.K.Kovačević.
- [29.] Information technology governance – COBIT model – IEEE Conference, MIPRO 2011 - DE, May 23-27, 2010, Opatija, Croatia, ISSN 1847-3938, p. 80-83., Radovanović D., Lučić D., Radojević T., Šarac M.
- [30.] Lučić D., "Necessity of IT Service Management and IT Governance" – IEEE Conference, MIPRO 2011- DE, May 23-27, 2011, Opatija, Croatia, ISSN 1847-3938, p. 84-87., Radovanović D., Šarac M., Adamović S.
- [31.] Analiza sigurnosti SSL saobraćaja u bežičnim računarskim mrežama, Infotech 2012, Jahorina, M. Šarac, M. Veinović, S. Adamović, D. Radovanović, A. Jevremović.
- [32.] Energetska efikasnost u primeni virtualnog datacentra i upotreba ekoloških računara u edukativnim ustanovama, rad objavljen u zborniku radova Sinergija

- 2012, Bijeljina, BiH, D. Stamenković, M. Šarac, S. Adamović, D. Radovanović.
- [33.] The Impact of Franchising on the Development of Tourism in Serbia, The EBES 2012 Istanbul Conference, T. Radojević, M. Šarac, N. Radović, D. Radovanović.
- [34.] Predavanje po pozivu u okviru manifestacije „Mesec dana matematike” u galeriji Srpske akademije nauke i umetnosti (SANU), „Bezbednosni problemi u zaštiti bežičnih komunikacija” Mladen Veinović, Marko Šarac, Saša Adamović.
- [35.] Radovanović D., Lučić D., Adamović S., Šarac M., "What is IT Governance?", UNITECH 2012 - International Scientific Conference, 16 – 17 November 2012, Gabrovo, Bulgaria
- [36.] „E - Agenda as a tool for support of evaluation of ECTS“ - Marko Šarac, TREND 2013, POHORJE, MARIBOR, 18.02- 21.02.2013.
- [37.] Radojević, T, Šarac, M, Radovanović D, Srejić, J,: „UTICAJ UVOĐENJA ELEKTRONSKIH SERVISA NA POBOLJŠANJE KVALITETA USLUGA U LOKALNOJ SAMOUPRAVI“, XII Međunarodni naučni skup Sinergija 2013, Univerzitet Sinergija, Bjeljina, Zbornik radova str.97-105
- [38.] Mladen Veinović, Milan Milosavljević, Marko Šarac, Jovan Jovanović, Saša Adamović, “Analiza bežičnih računarskih mreža zasnovanih na WPA2-PSK korišćenjem sopstvenog rešenja distribuiranog računarstva”, Etran 2013. Rad objavljen u zborniku radova Etran - Društvo za elektroniku, telekomunikacije, računarstvo, automatiku i nuklearnu tehniku.
- [39.] Milan Milosavljević, Jovan Jovanović, Saša Adamović, Marko Šarac, Aleksandar Jevremović i Vladislav Mišković, “Protokol za generisanje i razmenu apsolutno tajnih kriptoloških ključeva putem javnih kanala u savremenim računarskim mrežama”, Etran 2013. Rad objavljen u zborniku radova Etran - Društvo za elektroniku, telekomunikacije, računarstvo, automatiku i nuklearnu tehniku.