SVEUČILIŠTE U ZAGREBU
FILOZOFSKI FAKULTET
ODSJEK ZA INFORMACIJSKE I KOMUNIKACIJSKE ZNANOSTI
Ak. God. 2016./2017.

Petar Zadro

HTTPS protokol

Završni rad

Mentorica: dr. sc. Vjera Lopina

Zagreb, 2017.

Sadržaj

Uvod

Danas se gotovo više od polovine stanovništva u svijetu koristi internetom.[1] Zbog naglog razvoja interneta, bilo je potrebno omogućiti krajnjim korisnicima da sigurno razmjenjuju i pohranjuju osjetljive podatke poput lozinki, brojeva bankovnih računa, kreditnih kartica i sl. preko mreže. Naime, u današnje vrijeme najveći broj komunikacija, transakcija i poslova se obavlja na webu. Jedan od glavnih čimbenika razvoja enkriptirane komunikacije putem interneta je HTTPS protokol. Cilj ovog rada je istražiti problematiku sigurnosti razmjene podataka putem sigurnih i nesigurnih kanala preko mreže te ukazati na važnost HTTPS-a kao čimbenika koji omogućuje sigurnu vrstu digitalne komunikacije.

HTTPS zapravo sam po sebi nije protokol, već je to kombinacija dvaju protokola: HTTP i TLS/SSL. Ukratko, HTTP koji se provodi preko TLS/SSL protokola čini HTTPS. U prvom dijelu ovog završnog rada objasniti će se svrha i način rada tih dvaju protokola. Također će se ukazati na važnost, odnosno razlog zašto je bilo nužno razviti sigurnu i enkriptiranu vrstu komunikacije preko mreže te pojasniti od koga se zapravo trebamo štititi. Naime, postoje različiti oblici napada na našu privatnost preko interneta ako je naša mreža nesigurna.

Nakon definicije i opisa rada tih dvaju protokola, u drugom dijelu bit će prikazane glavne značajke HTTPS protokola, usluge koje pruža krajnjim korisnicima te važnost njegove primjene na mreži.

[1] Internet World Stats. Internet growth statistics. URL: http://www.internetworldstats.com/emarketing.htm (3.9.2017.)

1. HTTP protokol

S ozbirom da je HTTPS protokol nastao kombinacijom dvaju protokola HTTP i TLS/SSL kako bi komunikacija preko mreže bila enkriptirana, važno je objasniti što je HTTP, kako se razvijao i koje su njegove funkcije.

HTTP je najkorišteniji internetski protokol na svijetu te je temelj komunikacije i razmjene podataka preko World Wide Weba. Za početak, HTTP je skraćenica od „HyperText Transfer Protocol“ što bi u prijevodu značilo „protokol za prijenos hiperteksta“. HTTP se najprije koristio upravo za to, prijenos hiperteksta, no tijekom godina se razvijao što je omogućilo i prijenos slika, teksta, videa, xml datoteka, audio zapisa i drugih tipova podataka i informacija putem weba. HTTP protokol se odvija na aplikacijskom sloju TCP/IP modela koji omogućava mrežnim aplikacijama da međusobno komuniciraju i razmjenjuju podatke.

1.1. Povijesni razvoj HTTP-a

1.1.1. HTTP 0.9

Razvoj HTTP protokola se veže uz sami razvoj hiperteksta i interneta. Tim Berners Lee je započeo inicijativu dizajniranja protokola koji bi se koristio za komunikaciju i slanje hiperteksta zajedno sa HTML-om i drugim pripadajućim tehnologijama. Cijeli taj projekt je poznatiji još pod nazivom World Wide Web.[2] Prve korake je započeo 1991. godine kada je zacrtao osnovne ciljeve novog protokola i zajedno sa svojim timom izradio prvi prototip nazvan HTTP 0.9. Ta rana verzija omogućavala je klijentsko-serversku vezu na bazi zahtjev-odgovor protokola, koristio se ASCII protokolom koji se vodio preko TCP/IP veze. Bio je dizajniran za prenošenje hipertekstualnih datoteka (HTML) te da se veza između klijenta i servera zatvara nakon svakog novog zahtjeva.[3]

1.1.2. HTTP 1.0

Period između 1991. i 1995. godine obilježava razdoblje razvoja HTML specifikacija kada nastaju prvi web preglednici te se razvija javna internetska infrastruktura orijentirana prema krajnjim korisnicima. Tako je u svibnju 1996. godine organizacija pod nazivom HTTP Working Group (HTTP-WG) razvila HTTP 1.0. Cilj im je bio proširiti protokol sa dodatnim mogućnostima, operacijama, bogatijim metapodatcima te još mnogim drugim opcijama. Ovim

[2] Čop, Julian. HTTP/2 - protokol prilagođen modernom webu : završni rad. Rijeka: Julian Čop, 2016. str. 3

[3] Grigorik, Ilya. High Performance Browser Networking. Brief History of HTTP. HTTP 0.9: The One-Line Protocol. 2013 URL: https://hpbn.co/brief-history-of-http/#http-09-the-one-line-protocol (3.9.2017.)

razvojem HTTP prestaje biti protokol samo za prijenos hiperteksta već postaje protokol za prijenos hipermedija, no naziv HTTP, *HyperText Transfer Protocol*, je i dalje ostao isti.[4]

1.1.3. HTTP 1.1

U periodu između 1995. i 1999. je razvijen HTTP 1.1. koji je nastao kao revizija uspješnog HTTP 1.0. protokola. Službeno je objavljena 1997. godine u sklopu RFC 2068. HTTP 1.1 je u odnosu na prijašnje verzije uveo mnogo promjena, no glavna svojstva ovog protokola bile su brže učitavanje web stranica i smanjeni web promet. Jedna od najvažnijih značajki koja je to omogućila je bila sposobnost slanja više istodobnih zahtjeva preko jedne veze.[5]

1.1.4. HTTP/2

Nakon što se HTTP 1.1 koristio gotovo 16 godina kao internet standard bez ikakvih većih promjena i ažuriranja, zbog drastičnog napretka interneta i povećanja broja njegovih korisnika došlo je do potrebe za razvojem novog protokola. Specifikacija novog HTTP/2 protokola objavljena je kao RFC 7540 u svibnju 2015. godine. Cilj ovog novog protokola bio je uklanjanje svih nedostataka prethodnog HTTP 1.1 protokola te prilagođavanje novog protokola modernom webu. Većina web preglednika kao što su Google Chrome, Mozilla Firefox, Internet Explorer, Opera i Safari su već krajem 2015. godine prihvatili HTTP/2 i počeli ga koristiti kao novi protokol.[6]

1.2. Funkcije HTTP protokola

1.2.1. Klijentsko-poslužiteljski model HTTP protokola

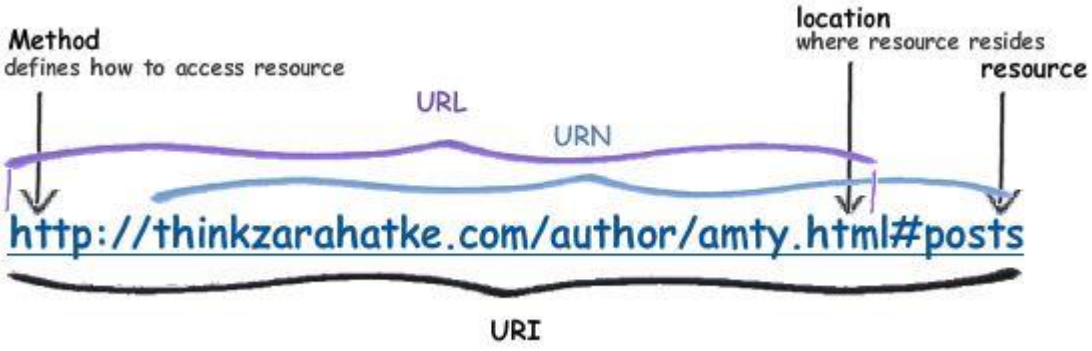
HTTP je već od najranije verzije HTTP 0.9 bio zasnovan na klijentsko-poslužiteljskom (*client-server*) modelu. To znači da HTTP poslužitelj, odnosno web poslužitelj, prima zahtjeve klijenata za dokumentima koje posjeduje. Svaki dokument kojim poslužitelj raspolaže je opisan s tri parametra, a to su:

- identifikator (URI – Uniform Resource Identifier)
- adresa (URL – Uniform Resource Locator)
- naziv (URN – Uniform Resource Name)

[4] Grigorik, Ilya. Nav. dj. URL: https://hpbn.co/brief-history-of-http/#http10-rapid-growth-and-informational-rfc (3.9.2017.)

[5] Isto. URL: https://hpbn.co/brief-history-of-http/#http11-internet-standard (3.9.2017.)

[6] Isto. URL: https://hpbn.co/brief-history-of-http/#http2-improving-transport-performance (3.9.2017.)



Slika 1 - URL, URN i URI struktura

Na osnovu ta tri parametra poslužitelj odlučuje na koji će način odgovoriti na postavljeni zahtjev. Zahtjev klijenta sadrži naredbu koja definira željenu akciju (GET, POST, DELETE itd.), adresu dokumenta, verziju HTTP protokola, te odgovarajuća zaglavlja kroz koja su definirani parametri klijenta. Odgovor poslužitelja na zahtjev sastoji se najprije od odluke hoće li prihvatiti komunikaciju s klijentom i uspostaviti vezu ili ne. U slučaju pozitivne odluke na zahtjev za podatcima (GET), poslužitelj šalje odgovor klijentu koji se sastoji od zaglavlja i podataka. Zaglavlje prethodi informaciji namijenjenoj korisniku, a sadrži parametre o samom poslužitelju, o podatcima i klijentu. Primljene podatke klijent prihvaća, izdvaja informacije namijenjene korisniku i prezentira mu ih. Umjesto podataka, klijent može dobiti obavijest o pogrješci, kojoj uzrok može biti na strani klijenta ili na strani poslužitelja. Najčešće poruke o pogrješci su "Datoteka nije pronađena" (404 - File not found) ili "Pristup dokumentu nije dopušten" (403 - Forbidden).[7]

1.2.2. Kodiranje znakova

HTTP koristi MIME (Multi-Purpose Internet Mail Extensions) definiciju skupa znakova i omogućava razmjenu dokumenata koji sadrže znakove različitih svjetskih jezika definiranjem skupa znakova primijenjenog u dokumentu. Kodiranje znakova se primjenjuje kako bi se slijed okteta (niz od 8 bitova) mogao ispravno protumačiti kao slijed znakova. Oznake skupova znakova definira IANA (Internet Assigned Numbers Authority). Ako nije navedena oznaka za tip znakova, podrazumijeva se ISO-8859-1. Znakovlje hrvatskog jezika definirano je kao ISO-8859-2. Danas se najčešće koristi UTF-8.[8]

[7] Mujarić, Eldis. Računalne mreže. HTTP protokol. URL: http://mreze.layer-x.com/s050100-0.html (3.9.2017.)

[8] Isto. URL: http://mreze.layer-x.com/s050100-0.html (3.9.2017.)

1.2.3. Označavanje sadržaja

Označavanjem sadržaja HTTP protokol omogućuje ukazivanje na transformaciju primijenjenu nad podatcima, kao što je komprimiranje (npr. zip, rar, itd.), ili kriptiranje. Time se postiže opis sadržaja koji nije čisti ASCII tekst, kao što su datoteke generirane nekim od programa (npr. doc, ppt, xls, pdf, itd.).[9]

1.2.4. Prijenos podataka različitih protokola

HTTP protokol omogućava komunikaciju između drugih protokola, kao što su SMTP (za razmjenu elektroničke pošte), NNTP (Usenet), FTP (prijenos datoteka) i sl. Ovim je omogućena dostupnost najčešće korištenih mrežnih usluga uporabom samo web preglednika. U slučaju kada se HTTP protokolom prenose informacije drugih protokola, primjenjuje se ili postupak tuneliranja (gdje se informacije prosljeđuju, bez analize o kojem se protokolu radi), ili postupak prevođenja kojeg obavljaju poveznici (engl. gateway). U takvim slučajevima, veza između klijenta i poslužitelja odvija se preko posrednika. Posrednik može biti i proxy poslužitelj, koji ima ulogu rasterećenja prometa od poslužitelja na lokalnoj mreži prema ostatku interneta. Klijent postavlja zahtjev proxy poslužitelju koji provjerava sadrži li traženu informaciju u svom međuspremniku (engl. cache) i ako je pronađe vraća je natrag klijentu. Ako nema traženu informaciju, proxy poslužitelj umjesto klijenta postavlja upit web poslužitelju. Odgovor web poslužitelja prosljeđuje korisniku koji je postavio zahtjev, ali ga proxy poslužitelj pohranjuje i u svoj međuspremnik, kako bi pri sljedećem upitu postigao brži odziv. Protokol za pristup dokumentu, poslužitelj, kao i mjesto dokumenta na poslužitelju definira jedinstvena adresa dokumenta - URL.[10]

[9] Mujarić, Eldis. Nav. dj. URL: http://mreze.layer-x.com/s050100-0.html (3.9.2017.)

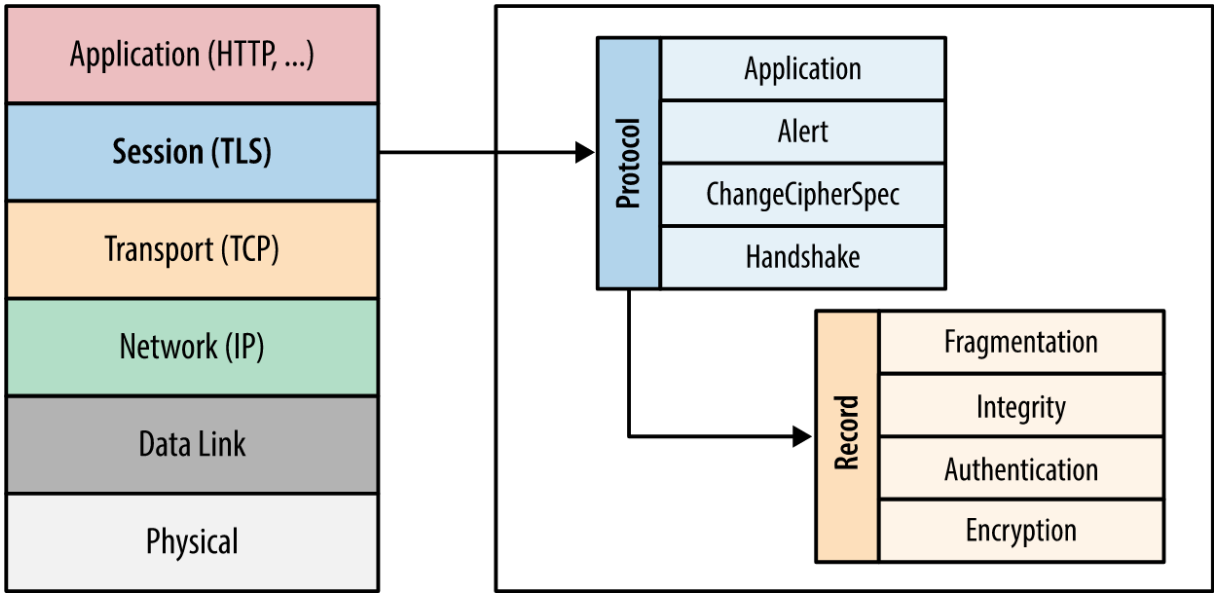
[10] Isto. URL: http://mreze.layer-x.com/s050100-0.html (3.9.2017.)

2. TLS/SSL

TLS/SSL protokol je nastao s ciljem zaštite komunikacija i razmjene podataka putem mreže. Ukratko, ovaj protokol omogućuje da dvije strane, odnosno dva računala međusobno komuniciraju i razmjenjuju podatke bez da itko drugi može znati o čemu je riječ. Čak i da netko upadne u njihov sigurnosni komunikacijski kanal i domogne se njihove poruke, ne bi je mogao pročitati jer je ta poruka enkriptirana. Drugim riječima, samo te dvije strane mogu pročitati tu poruku, jer je poruka bez odgovarajućeg ključa nečitljiva.[11]

2.1. Povijesni razvoj TLS/SSL protokola

SSL (Secure Socket Layer) protokol je razvio Netscape s namjerom kako bi online trgovinu preko weba učinio sigurnijom, radi toga osobni podatci kupca trebali su biti enkripitrani. Da bi to ostvarili, SSL protokol se morao provoditi kroz aplikacijski sloj, na vrhu TCP/IP modela, kako bi protokoli iznad njega (HTTP, e-pošta, brzo slanje poruka i dr.) mogli neometano raditi i sa sigurnošću. Ako je SSL pravilno konfiguriran, promatrač koji nije dio komunikacijskog kruga može samo razaznati kranje točke veze, vrstu enkripcije te frekvenciju i približnu količinu podataka koji su razmijenjeni, ali nikako ne može doći do samih podataka ili ih izmijeniti.[12]



Slika 2 - Transport Layer Security (TLS)

[11] Symantec Corporation. What is an SSL Certificate? The Ultimate Guide.
URL: https://www.symantec.com/page.jsp?id=ssl-information-center (3.9.2017.)

[12] Grigorik, Ilya. Nav. dj. URL: https://hpbn.co/transport-layer-security-tls/ (3.9.2017.)

SSL 2.0 je bila prva javno dostupna verzija protokola, no ubrzo je bila zamijenjena verzijom SSL 3.0 kako bi nadomjestili par sigurnosnih nedostataka prethodne verzije. Pošto je SSL bio u vlasništvo Netscapea, IETF (Internet Engineering Task Force) je zatražio da se protokol standardizira, te je tako u siječnju 1999. godine nastao TLS (Transport Layer Security) 1.0 u sklopu RFC 2246. Od tada se protokol razvijao pod vodstvom IETF-a s ciljem da se poboljša njegova sigurnost i prošire njegove opcije te je tako TLS 1.1 objavljen u travnju 2006. godine u sklopu RFC 2246, TLS 1.2 u kolovozu 2008. godine u sklopu RFC 5246, dok najnoviji TLS 1.3 samo što nije objavljen.

TLS i SSL su najpoznatiji po tome što se koriste kao baza HTTPS-a za siguran način transakcije putem interneta između web preglednika i web poslužitelja. TLS/SSL se također mogu koristiti i za druge protokole na aplikacijskom sloju, kao što su „File Transfer Protocol“ (FTP), „Lightweight Directory Access Protocol“ (LDAP) i „Simple Mail Transfer Protocol“ (SMTP).

Valja napomenuti da iako postoji razlika između TLS i SSL protokola, u današnje vrijeme se ta dva pojma koriste naizmjenično. Prije se smatralo da je TLS 1.0 bio samo neznatno sigurniji od tadašnjeg SSL 3.0 protokola, no ipak se ispostavilo da je SSL 3.0 bio puno zastarjeliji nego što se smatralo. Moderniji web preglednici će uvijek nastojati upozoriti svoje korisnike u slučaju da posjećuju stranice koje se koriste zastarjelim verzijama ovog protokola, jer te verzije nisu više sigurne za korištenje. U svrhu ovog završnog rada, daljnje će se samo koristiti pojam TLS.[13]

2.2. Funkcije TLS protokola

Bez obzira što postoji više vrsti TLS protokola, svi moderni web pretraživači u sebi imaju ugrađenu opciju da se uvijek koriste najnovijom verzijom TLS protokola radi što veće sigurnosti. Kako bi TLS protokol bio siguran, on je dizajniran s ciljem da svim aplikacijama koje se koriste ovim protokolom pruži tri glavne usluge, a to su:

- Ovjeravanje autentičnosti
- Enkripcija poruka i podataka
- Očuvanje integriteta poruke

[13] Kangas, Erik. The LuxSci FYI Blog. SSL versus TLS – What's the difference?. 19.7.2016. URL: https://luxsci.com/blog/ssl-versus-tls-whats-the-difference.html (3.9.2017.)

Ovjeravanje autentičnosti je mehanizam pomoću kojeg provjeravamo identitet web poslužitelja nekog servera. Enkripcija je mehanizam s kojim skrivamo sadržaj koji prenosimo s jednog računala na drugo, dok je očuvanje integriteta podataka mehanizam s kojim provjeravamo jesu li podatci koji su poslani bili mijenjani ili krivotvoreni.[14]

Da bi se uspostavio kriptološki siguran podatkovni kanal između dvaju računala, mora se unaprijed dogovoriti koja vrsta enkripcije će se koristiti te koji će se ključevi koristiti za dekripciju podataka. TLS protokol obavlja ovu razmjenu s takozvanim TLS „rukovanjem“, no o tome će biti nešto više riječi u idućem poglavlju. Glavni razlog zašto ovo „rukovanje“ funkcionira uspješno je zato što se koristi kriptografija s javnim ključem, odnosno asimetrična kriptografija. Ona omogućuje dogovor dvaju računala oko zajedničkog tajnog ključa bez da prethodno znaju išta jedno o drugome, i to mogu obaviti preko veze koja nije enkriptirana.

TLS „rukovanje“ također omogućuje računalima međusobno ovjeravanje identiteta. Ovim postupkom možemo preko nekog web preglednika ovjeriti je li server na koji se želimo spojiti zaista taj server, na primjer server od naše banke, a ne netko tko se pretvara da je server od naše banke tako što koristi isti naziv ili IP adresu. Serveri također mogu ovjeriti identitete svojih klijenata, na primjer server neke tvrtke može potvrditi da se zaista radi o nekom njihovom zaposleniku jer svaki od zaposlenika ima svoj jedinstveni certifikat koji je potvrdila njegova tvrtka.

Osim enkripcije i ovjere autentičnosti, TLS protokol koristi i tehniku očuvanja integriteta poruka poslanih preko ovog protokola korištenjem vlastitog mehanizma za potpisivanje svake pojedine poruke s takozvanim „kodom za ovjeravanje poruke“, (eng. MAC - message authentication code). MAC algoritam je jednosmjerna kriptografska hash funkcija[15], čiji su ključevi dogovoreni od strane korisnika veze. Svaki put kada se pošalje TLS zapis, nastaje MAC vrijednost koja se pridodaje na tu poruku, tako da primatelj može preračunati i potvrditi MAC vrijednost čime bi dokazao cjelovitost i vjerodostojnost poruke.

[14] Grigorik, Ilya. Nav. dj. URL: https://hpbn.co/transport-layer-security-tls/#encryption-authentication-and-integrity (3.9.2017.)

[15] „Hash funkcija je svaki algoritam koji od podataka proizvoljne dužine stvara podatke fiksne dužine. Vrijednost koja je izlazna je fiksne dužine bez obzira na ulaznu vrijednost podataka.“ Learn Cryptography. What Are Hash Functions. URL: https://learncryptography.com/hash-functions/what-are-hash-functions (5.9.2017.)

2.2.1. TLS „rukovanje“

Razmjena podataka između klijenta i servera počinje utvrđivanjem verzije TLS protokola koji će se koristiti, vrstu enkripcije i ako je nužno i potvrdu certifikata. Također je važno da se koristeći asimetričnom enkripcijom dođe do zajedničkog tajnog ključa kako bi se izbjegao problem distribucije ključa. Na žalost, svaki od ovih koraka zahtijeva ponovno slanje paketa između klijenta i servera što uzrokuje lagane zastoje između svakog pokretanja TLS veze.[16] Koraci za uspostavljanje TLS veze TLS rukovanjem su sljedeći:

1. TLS klijent šalje „client hello“ poruku gdje navodi kriptografske podatke poput verzije TLS protokola kojim se koristi te po želji klijenta navodi vrste enkripcije podržane od strane klijenta. Poruka također sadrži niz bajtova koji će biti korišteni u kasnijim proračunima. Protokol dozvoljava da „client hello“ poruka navede i metode kompresije podataka podržane od strane klijenta.
2. TLS server odgovara sa „server hello“ porukom koja sadrži vrstu enkripcije odabranu sa liste podržanih vrsti od strane klijenta, ID broj sesije te još jedan nasumični niz bajtova. Server šalje svoj certifikat te ako zahtijeva digitalni certifikat za potvrdu autentičnosti klijenta, onda mu šalje „client certificate request“ koji obuhvaća listu tipova certifikata i istaknutih imena podržanih od strane agencije za izdavanje certifikata.
3. Klijent potvrđuje digitalni certifikat servera.
4. Klijent šalje svoj nasumični niz bajtova koji omogućuje klijentu i serveru stvaranje tajnoga ključa koji će biti korišten za enkripciju poruka i podataka koji slijede. Taj sami niz bajtova je enkriptiran s javnim ključem servera.
5. Ako je server slao „client certificate request“, klijent onda šalje nasumični niz bajtova enkriptiran privatnim ključem klijenta, zajedno s digitalnim certifikatom klijenta ili po mogućnosti šalje „no digital certificate alert“. To se koristi samo kao upozorenje, ali u nekim slučajevima TLS rukovanje neće proći uspješno ako je potvrda autentičnosti klijenta obavezna.
6. Server potvrđuje certifikat klijenta.
7. Klijent šalje serveru „finished“ poruku koja je enkriptirana s tajnim ključem što ukazuje da je klijent obavio svoj dio TLS rukovanja.
8. Server šalje klijentu „finished“ poruku koja je enkriptirana s tajnim ključem što ukazuje da je i server obavio svoj dio TLS rukovanja.

[16] Grigorik, Ilya. Nav. dj. URL: https://hpbn.co/transport-layer-security-tls/#tls-handshake (3.9.2017.)

9. Za vrijeme trajanja TLS sesije, klijent i server mogu međusobno razmjenjivati poruke i podatke koji su simetrično enkriptirani korištenjem zajedničkog tajnog ključa.[17]

2.2.2. RSA, Diffie-Hellman algoritam i savršena tajnost prema unaprijed

Zbog raznih povijesnih i komercijalnih razloga, RSA[18] rukovanje je bio dominantni tip razmjene ključeva u većini TLS implementacija. RSA algoritam funkcionira tako što klijent generira simetrični ključ, enkriptira ga serverovim privatnim ključem i odgovorom ga šalje natrag serveru kako bi taj ključ dalje koristili kao simetrični ključ za već utvrđenu sesiju. Zauzvrat, server koristi svoj privatni ključ za dekripciju simetričnog ključa poslanog od strane klijenta čime je razmjena ključeva gotova. U tom trenu klijent i server koriste dogovoreni simetrični ključ kako bi enkriptirali svoju sesiju.[19]

Rukovanje RSA algoritmom ima jednu kritičnu manu: isti javno-privatni par ključeva se koristi za ovjeravanje autentičnosti servera te za enkripciju simetričnog sesijskog ključa koji je poslan serveru. Kao rezultat toga, napadač koji dobije pristup privatnom ključu servera i prisluškuje razmjenu između klijenta i servera, može dekriptirati cijelu njihovu sesiju. Ako napadač trenutačno ne posjeduje privatni ključ, on i dalje može snimiti enkriptiranu sesiju i dekriptirati je jednom kada se domogne privatnog ključa.

Za razliku od RSA rukovanja, razmjena ključeva preko Diffie-Hellman[20] algoritma omogućava klijentu i serveru da se slože oko zajedničkog tajnog ključa bez da ga eksplicitno navode u fazi rukovanja. Privatni ključ se koristi za potpisivanje i potvrđivanje rukovanja, no dogovoreni simetrični ključ ostaje cijelo vrijeme kod klijenta i servera te ga se pasivni napadač ne može domoći čak ni da ima pristup privatnom ključu.

---

[17] IBM Knowledge Center. An overview of the SSL or TLS handshake. URL: https://www.ibm.com/support/knowledgecenter/en/SSFKSJ_7.1.0/com.ibm.mq.doc/sy10660_.htm (3.9.2017.)

[18] „RSA kriptosustav je prvi i najšire korišteni kriptosustav s javnim ključem, koji je dobio ime prema svojim izumiteljima (Rivest, Shamir, Adleman).“ TechTarget. RSA algorithm (Rivest-Shamir-Adleman). URL: http://searchsecurity.techtarget.com/definition/RSA (5.9.2017.)

[19] Grigorik, Ilya. Nav. dj. URL: https://hpbn.co/transport-layer-security-tls/#rsa-diffie-hellman-and-forward-secrecy (3.9.2017.)

[20] „Diffie-Hellmanov postupak razmjene ključa je kriptografski protokol koji omogućuje osobama koje se ne poznaju da razmjene simetrični tajni ključ preko nezaštićenog komunikacijskog kanala. Ime je dobio prema svojim izumiteljima (Diffie, Hellman).“ Nacionalni CERT. Diffie-Hellman protokol. str. 11 URL: http://www.cis.hr/www.edicija/LinkedDocuments/NCERT-PUBDOC-2009-12-284.pdf (5.9.2017.)

Najbolje od svega, Diffie-Hellman razmjena ključeva smanjuje rizik ugrožavanja prijašnjih komunikacijskih sesija. Naime, postoji mogućnost stvaranja novih jednokratnih ključeva i odbacivanje starih svaki put kada se razmjenjuju ključevi. Kao rezultat toga, jednokratni ključevi nikad nisu priopćeni i tehnički se svaki put ponovo ugovore za svaku novu sesiju. U najgorem slučaju, napadač može ugroziti klijenta ili server te se domoći sesijskih ključeva trenutačne ili budućih sesija. Međutim, posjedujući privatni ili jedan od trenutačnih jednokratnih ključeva napadaču ni na koji način ne pomaže da dekriptira neke od prijašnjih sesija.

Razmjena ključeva Diffie-Hellman algoritmom i upotrebom jednokratnih ključeva omogućuje „savršenu tajnost prema unaprijed“ (engl. Perfect Forward Secrecy): svojstvo protokola za razmjenu ključeva da razotkrivanje tajnih ključeva neće ugroziti ključeve koji su prije toga izvedeni iz tih trajnih ključeva. Drugim riječima, napadač koji se domogne jednog od takvih ključeva neće moći pročitati prošle, a ni buduće poruke u tom komunikacijskom kanalu.

2.3. TLS/SSL certifikati

Izdavanje TLS/SSL certifikata spada u nadležnost Agencije za izdavanje certifikata (CA). Postupak dobivanja certifikata obuhvaća davanje podataka CA o identitetu web stranice i tvrtke. Nakon podnesenog zahtjeva koji čini podatkovnu datoteku o serveru i ovjere od strane CA, serveru se dodjeljuje TLS certifikat. Novonastali TLS je usklađen s privatnim ključem servera. Sve veze između web poslužitelja servera i web preglednika klijenta su enkriptirane. Time je stvoren privatni i javni ključ.[21]

Podatci TLS certifikata sadrže naziv domene i naziv tvrtke servera te informacije o adresi, nazivu grada, općine i države. Također navodi podatke o datumu isteka roka trajanja TLS certifikata te pojedinosti o CA koja je izdala certifikat. Svaki put kad neki web preglednik uspostavi vezu s web stranicom zaštićenom TLS protokolom, preglednik će prvo provjeriti je li TLS certifikat stranice i dalje važeći. Web preglednik također provjerava pouzdanost CA te autentičnost certifikata. Ukoliko jedna od ovih provjera ne uspije ili nije valjana, web preglednik će korisniku odnosno klijentu prikazati upozorenje da web stranicu koju žele posjetiti nema valjani TLS certifikat te stoga nije sigurna.

Povjerenje je jedan od glavnik čimbenika kako TLS certifikati funkcioniraju. CA su važne za TLS protokol upravo zato što se proces ovjeravanja autentičnosti oslanja na digitalne

[21] Comodo CA. What is SSL (Secure Sockets Layer)?. URL: https://www.instantssl.com/ssl.html (3.9.2017.)

certifikate. CA su ti koji potvrđuju identitete i jamče sigurnost web poslužitelja koji su zatražili TLS certifikat. Moguće je i ručno provjeravati valjanost TLS certifikata svake pojedine stranice koju posjećujemo, no upravo zato postoje CA koje to rade umjesto nas. Ako je sigurnost neke stranice koja posjeduje TLS certifikat ugrožena, onda CA koja je izdala taj certifikat ima odgovornost da taj isti certifikat opozove.[22]

2.4. Nedostatci TLS protokola

2.4.1. Povjerenje

U prethodnom poglavlju smo spomenuli kako je model po kojemu funkcionira proces izdavanja certifikata zasnovan na uzajamnom povjerenju između CA i web preglednika korisnika. Iako nam TLS protokol jamči da nas nitko ne može pratiti dok komuniciramo preko zaštićene veze, i dalje se svodi na povjerenje da li mi zaista komuniciramo s određenom tvrtkom ili strankom s kojom smo namjeravali komunicirati. CA su ti kojima mi vjerujemo da je identitet stranice koju posjećujemo valjan. Neke „sigurne stranice“ se ne zamaraju s traženjem dopuštenja od strane CA već si samostalno odobre svoje ključeve. Pojedini serveri pak koriste certifikate koji su besplatni ili gotovo besplatni što znači da su uložili veoma malo truda za sigurnost svoje tvrtke odnosno web poslužitelja. U ovakvim slučajevima TLS korisniku pruža malo ili gotovo nikakvo jamstvo da je server s kojim se komunicira zaista server tvrtke s kojom želimo komunicirati, a ne neki haker koji pokušava krivotvoriti identitet tvrtke kako bi se domogao korisnikovih osjetljivih podataka.[23]

Radi zaštite korisnika od ovakvih napada potrebna je velika pažnja na upozorenja koje šalje TLS prilikom spajanja na nečiju stranicu. Neka od tih upozorenja mogu ukazivati na certifikate s isteklim rokom trajanja, krive nazive domena stranice i nepouzdane certifikate čiji javni ključ nije odobren od strane CA kojoj naš web preglednik vjeruje. U ovakvim slučajevima trebali bismo biti na strogom oprezu.

2.4.2. Duljina ključa

Kada kažemo da „samo onaj tko posjeduje privatni ključ može dekriptirati nešto enkriptirano javnim ključem“ onda je ta izjava točna samo u slučaju kada se privatni ključ ne može „pogoditi“. Hakeri su u stanju to učiniti ako isprobaju sve moguće kombinacije nekog ključa

[22] Grigorik, Ilya. Nav. dj. URL: https://hpbn.co/transport-layer-security-tls/#chain-of-trust-and-certificate-authorities (3.9.2017.)

[23] Kangas, Erik. The LuxSci FYI Blog. How Does Secure Socket Layer (SSL or TLS) Work?. 22.7.2013. URL: https://luxsci.com/blog/how-does-secure-socket-layer-ssl-or-tls-work.html (3.9.2017.)

(eng. Brute force attack) ili ako su prethodno saznali nešto o samoj komunikaciji, kao npr. dio običnog teksta ili puno različitih enkripcija nad nekim običnim tekstom. Treća opcija je zlouporaba nedostataka nekog algoritma za enkripciju.

Starije ključeve je moguće probiti sistemom „pokušaja i pogreške“, tj. pokušavanja različitih kombinacije ključeva dok se ne naiđe na onu kombinaciju koja odgovara, no za to je potrebno dovoljno jako računalo ili dovoljno vremena za probijanje svih kombinacija. U današnje vrijeme se u TLS certifikatima koriste ključevi koji su 2048-bitni ili čak jači, što ih čini sigurnim od ovakve vrste napada, barem na neko dulje vrijeme. Isto tako, šifre koje se koriste za simetričnu enkripciju u TLS protokolu su 128-birne ili više, što omogućuje gotovo beskonačnu kombinaciju mogućih zaporki.[24]

2.4.3. Vrsta enkripcije

TLS se koristi mnoštvom različitih „šifri“ (vrstama enkripcije) kako bi omogućio simetričnu enkripciju. Korištenjem loših odnosno slabih šifri može brzo dovesti do ugrožavanja TLS veze. U današnje vrijeme se općenito preporuča uporaba 128-bitne ili jače AES (Advanced Encryption Standard) enkripcije kao zadane vrste enkripcije za TLS.

2.4.4. Povećano opterećenje procesora

Povećano opterećenje procesora niju nužno nedostatak, ali je najznačajnije ograničenje u izvršavanju TLS protokola. Proces kriptografije, pogotovo operacije vezane uz javni ključ, su veoma tehnički zahtijevne. Kao rezultat toga, performanca TLS-a varira od računala do računala. Nažalost, ne može se točno odrediti koliko točno implementacija TLS-a utječe na performanse računala. Performansa se razlikuje ovisno o tome koliko često su veze uspostavljene i koliko dugo traju. TLS koristi najviše resursa u fazi TLS rukovanja, odnosno dok uspostavlja vezu.[25]

[24] Kangas, Erik. The LuxSci FYI Blog. How Does Secure Socket Layer (SSL or TLS) Work?. 22.7.2013. URL: https://luxsci.com/blog/how-does-secure-socket-layer-ssl-or-tls-work.html (3.9.2017.)

[25] Microsoft TechNet. What is TLS/SSL?. URL: https://technet.microsoft.com/en-us/library/cc784450(v=ws.10).aspx (3.9.2017.)

3. HTTPS protokol

HyperText Transfer Protocol Secure (HTTPS) je sigurna inačica HTTP-a korištena za obavljanje sigurnih transakcija na webu. Omogućuje autentifikaciju i enkriptiranu komunikaciju, zbog čega je vrlo koristan u elektroničkom poslovanju. Umjesto komuniciranja otvorenim (nezaštićenim) tekstom (eng. plain text), HTTPS protokol enkriptira podatke tijekom korisničke sesije, koristeći pritom neku inačicu TLS protokola, čime ostvaruje razumnu razinu zaštite od uljeza i napadača. Razina zaštite koju pruža ovisi o ispravnosti implementacije u web pregledniku, o vrsti i obilježjima poslužiteljskog (server) softvera i o podržavanim kriptografskim algoritmima.[26]

Kao korisnik znamo da je neka web stranica zaštićena HTTPS vezom kad na početku web adrese stranice piše *https://* a ne samo *http:///*. Isto tako, ovisno o razini certifikata dodijeljenog nekom web poslužitelju, korisnici će uočiti prisutnost zelenog lokota ili zelene oznake adresne trake koja označava razinu zaštite na toj web lokaciji.

Slika 3 - indikator HTTPS veze

Najveća prednost za korisnika je što HTTPS protokol ovjerava identitet web stranice ili web poslužitelja na koji se klijent želi spojiti te enkriptira gotovo sve informacije koje se šalju između korisnika i web stranice ili usluge. Zaštićene informacije uključuju kolačiće, putove URL-a, podneske obrazaca i parametre niza upita. HTTPS je dizajniran da ove informacije nitko drugi ne može pročitati ili mijenjati za vrijeme prijenosa informacija.[27]

[26] Panian, Željko. Informatički enciklopedijski rječnik. - Sv.1 : @-L / Željko Panian. Zagreb : Europapress holding-Jutarnji list. 2005. str. 262

[27] The HTTPS-Only Standard. URL: https://https.cio.gov/ (3.9.2017.)

Web preglednici su konfigurirani da uvijek vjeruju određenim CA koje mogu izdati kriptološki potpisane certifikate vlasnicima web poslužitelja. Ove potvrde priopćavaju klijentima da je vlasnik web poslužitelja dokazao CA da je on zaista vlasnik svoje domene u vrijeme kad je certifikat izdan. Ovime se sprječava da se nepoznate ili nepouzdane stranice predstavljaju kao neke druge web stranice.

HTTPS protokol ima nekoliko važnih ograničenja. Iako HTTPS enkriptira cijeli proces zahtjeva i odgovora između poslužitelja i korisnika, IP adrese (eng. Internet Protocol address) i nazivi odredišne domene nisu enkriptirani tijekom komunikacije. Čak i enkriptirane veze indirektno otkrivaju neke informacije, kao što su vrijeme provedeno na nekoj web stranici te količina zatraženih podataka ili dostavljenih informacija.[28]

HTTPS protokol osigurava samo integritet veze između dvaju računala, ali ne i samih računala. HTTPS nije dizajniran za zaštitu web poslužitelja od napadača i hakera ili za sprječavanje web poslužitelja da se služi korisničkim podatcima tijekom normalnog rada. Isto tako ako je korisnikovo računalo ugroženo od strane napadača, on je u stanju namjestiti da buduće veze na tom računalu budu pod njegovom kontrolom. Ako su CA koji izdaju certifikate ugroženi ili pak zlonamjerni, onda je i jamstvo HTTPS-a oslabljeno ili skroz nevažeće.

3.1. Vrste napada i obrane

Bez obzira na koji način se korisnici spajaju na web, uvijek postoji niz ljudi koji ih mogu napasti, bilo da špijuniraju njihovu vezu, oponašaju ih, miješaju im se u vezu ili čine sve troje od navedenog. Operater wifi mreže je u stanju to učiniti ili pak internetski poslužitelj koji stoji između veze klijenta i servera. Ako netko ima pristup wifi ruteru, on je također u stanju napasti našu privatnu vezu.

Firesheep je primjer pasivnog napada na mrežu. On funkcionira tako da prisluškuje sadržaj mrežne komunikacije između web preglednika klijenta i servera, ali ih ne preusmjerava niti modificira. Program XKeyscore je program od NSA (National Security Agency) koji pasivno napada HTTPS promet tako što prikuplja masivne količine podataka iz mrežne komunikacije.[29]

[28] The HTTPS-Only Standard. URL: https://https.cio.gov/ (3.9.2017.)

[29] Electronic Frontier Foundation. HTTPS Everywhere. How to deploy HTTPS correctly. Modes of Attack and Defense. URL: https://www.eff.org/https-everywhere/deploying-https (3.9.2017.)

Napadi na HTTPS vezu se mogu svesti u dvije kategorije: napadi na TLS protokol i napadi na infrastrukturu CA modela. U sklopu ovog rada bit će navedeni samo neki primjeri napada koji su nastali kroz povijest razvoja HTTPS-a te na koji su način ti napadi bili zaustavljeni.

3.2. Napadi na TLS protokol

3.2.1. Slaba enkripcija i duljina ključeva

Nekolicina algoritama za enkripciju koji se danas koriste u TLS-u smatraju se nesigurnima. Bilo koja metoda simetričnog šifriranja s ključem od 40, 56 ili 64 bita podložna je napadu silom, što znači da napadač može isprobavanjem svih mogućih kombinacija ključa doći do one stvarne kombinacije. TLS protokol je prije podržavao DES, RC2 i RC4, vrste algoritama za enkripciju, koji su koristili navedene duljine ključeva. Metoda asimetričnog šifriranja poput RSA podložna je napadima faktorizacije kada se koriste s modulom od 512 bita.[30]

3.2.2. Generator pseudonasumičnih brojeva (PRNG)

Mnoge uobičajene računalne aplikacije (poput video igara) koriste bilo kakve dostupne izvore nasumičnosti kako bi stvorili početnu vrijednost, tzv. „sekvencu“, za generator pseudonasumičnih brojeva (PRNG). PRNG djeluje tako da konstantno preslaže sekvencu. Obično se u početku uzme neki kratak nasumični broj koji PRNG proširuje u duži i veći niz bitova koji se doima nasumičnim. Za jednostavnu video igru, sekvenca se samo mora mijenjati svaki put kad se igra upali što će činiti sekvencu predvidljivom. U kriptografskim aplikacijama, međutim, nepredvidljivost sekvence je neophodna. Ako neki napadač može suziti skup mogućih sekvenci, njegov posao se znatno olakšava.[31]

Većina vrijednosti u TLS protokolu su generirane nasumično, uključujući i tajne ključeve. Postupak snažnog pseudonasumičnog generiranja niza brojeva sa sekvencom visoke entropije omogućuje stvaranje jakih ključeva. Netscape web preglednik se koristio PRNG implementacijom koja nije generirala jake ključeve što je dovelo do predvidljivosti tajnih ključeva korištenih u TLS sesijama.

Učinkovita metoda odabira vrijednosti sekvence za PRNG je bitan dio kriptografskog sustava kao što je TLS. Ako se vrijednosti sekvence za PRNG mogu lako nagađati, razina sigurnosti

[30] SoK: SSL and HTTPS: Revisiting Past Challenges and Evaluating Certificate Trust Model Enhancements. URL: http://ieeexplore.ieee.org/document/6547130/ (3.9.2017.)

[31] Goldberg, Ian.; Wagner, David. Randomness and the Netscape Browser. How secure is the World Wide Web?. URL: https://people.eecs.berkeley.edu/~daw/papers/ddj-netscape.html (3.9.2017.)

koju program nudi znatno se smanjuje jer zahtijeva manje posla napadaču za dekriptiranje poruka koje je presreo u komunikaciji.

3.2.3. RSA enkodiranje

SSL 3.0 s RSA metodom šifriranja koristi jednostavni RSA za prijenos enkodiranog tajnog ključa serveru za vrijeme faze rukovanja. Ako za vrijeme dekripcije i dekodiranja obični tekst nije valjano enkodiran, klijent će dobiti obavijest o pogrješci. U tom trenutku napadač može doći do enkriptiranog tajnog ključa i u zasebnoj fazi rukovanja na isti server poslati promijenjene verzije istoga ključa te dobiva podatak jesu li ta dva ključa odgovarajuća. Tada s tom informacijom može nakon određenog vremena doći do dekriptiranog tajnog ključa. Stoga TLS 1.0 preporučuje da se pogrješke pri kodiranju rješavaju nerazlučivo od uspješnih dekripcija.[32]

3.2.4. Napad degradiranja metode šifriranja

Server i klijent se u procesu TLS rukovanja dogovaraju koju će metodu šifriranja koristiti. U SSL 2.0 „man-in-the-middle“ napadač može utjecati za vrijeme pregovaranja te namjestiti da server i klijent koriste najslabiju moguću zajedničku metodu šifriranja, umjesto da koriste najjaču. Ovaj propust je popravljen u SSL 3.0 i svim verzijama TLS-a tako što klijent šalje ovjereni sažetak prethodnih poruka za rukovanje i čeka ovjerenu potvrdu od strane servera, nakon što je „kod za ovjeravanje poruke“ (MAC) ustanovljen.

3.2.5. Napad degradiranja TLS verzije

Slično tomu postoji i napad degradiranja TLS verzije koja se isto uspostavlja za vrijeme TLS rukovanja. Dok ovakav tip napada nije moguć nad strogim implementacijama TLS protokola, no mnogi serveri reagiraju na određene pogreške servera tako što se pokušavaju ponovno spojiti, ali sa starijom verzijom TLS-a. Napadač ovakve pogreške može namjerno uzrokovati, odnosno krivotvoriti ih. Novije verzije TLS-a sprječavaju bilo kakve pokušaje degradiranja na SSL 2.0. U nekim slučajevima je ipak moguće degradacije sa novije na stariju verziju (npr. s TLS 1.1 na TLS 1.0). Ovakav tip napada je moguće ublažiti ako tijekom TLS rukovanja klijent i server navedu samo najjaču verziju TLS-a koju podržavaju.

[32] SoK: SSL and HTTPS: Revisiting Past Challenges and Evaluating Certificate Trust Model Enhancements, URL: http://ieeexplore.ieee.org/document/6547130/ (3.9.2017.)

3.3. Napadi na infrastrukturu CA modela

3.3.1. Certifikacija

Web certifikat veže javni potpisni ključ uz neki „identitet“. Ispravnost vezivanja se potvrđuje digitalnim potpisom CA od koje se očekuje da će održavati točnost vezivanja tijekom vremena.[33]

Najbitnije svojstvo certifikata nekog HTTPS servera je naziv domene čiji je vlasnik nositelj tog certifikata. U slučaju da netko zatraži certifikat koji glasi na ime domene, CA će tražiti od podnositelja zahtjeva dokaze o vlasništvu zatražene domene. Ovakva provjera se vrši s pretpostavkom da je naziv domene mapiran na točan web server (IP adresu). Takav tip certifikata se još naziva „domain validated“ (DV) certifikat.

Izdani certifikati mogu sadržavati i neke dodatne podatke ovjerene od strane CA, kao što su naziv organizacije i poštanska adresa tvrtke. Postupci provjere valjanosti su s vremenom degradirali, time što neki CA koriste potpuno automatizirane postupke izdavanja certifikata. Kao odgovor na to, CA/Browser forum, dobrovoljna grupa CA osnovana 2005. godine, osnovala je „extended validation“ certifikate te smjernice o postupku njihovog izdavanja.

3.3.2. „Sidrenje povjerenja“

Jedna od važnijih funkcija CA je ovjeravanje identiteta osobe navedene u polju SubjectName koja podnosi zahtjev za certifikatom. Polje SubjectName na TLS certifikatu se odnosi na usluge vezane uz DNS (Domain Name System). SubjectName veže certifikat uz određeni server ili naziv domene.[34] Budući da nijedna osoba nema kontrolu nad svim imenima, ne može se odrediti tko bi točno bio najprikladniji za ovakav tip ovjere. Kao rezultat toga, postoji širok spektar CA, pri čemu je većina certifikata izdana od strane komercijalnih CA koji su povezani s tvrtkama za sigurnost i registraciju domena.

Softver trgovci (kao što su Microsoft, Apple, Mozilla, Opera itd.) imaju konfiguriranu listu zadanih CA na svojim operacijskim sustavima i web preglednicima kao „sidro povjerenja“. Svaka HTTPS stranica čiji se certifikat smatra pouzdanim od strane web preglednika se ujedno i korisnicima čini kao pouzdana stranica upravo zato što jedan ili više „sidri

[33] SoK: SSL and HTTPS: Revisiting Past Challenges and Evaluating Certificate Trust Model Enhancements, URL: http://ieeexplore.ieee.org/document/6547130/ (3.9.2017.)

[34] Microsoft TechNet. Understanding TLS Certificates. 11.4.2011.
URL: https://technet.microsoft.com/en-us/library/aa998840(v=exchg.141).aspx (3.9.2017.)

povjerenja“ jamče za sigurnost certifikata te stranice. Prema izvješću SSL Observatorija (projekt koji je zadužen za istraživanje sigurnosti certifikata izdanih svim HTTPS stranicama na webu[35]), između Mikrosoftovog Internet Explorera i Mozillinog Firefoxa, oko 1500 CA certifikata iz gotovo 650 organizacija u skoro 50 zemalja svijeta su podržani od strane njihovih web preglednika.[36]

Bez obzira koliko je pouzdana CA koja izdaje certifikat nekoj web stranici, bez dodatnih poboljšanja sigurnosti, napadač ima priliku birati najslabiju CA kako bi se domogao njihovog certifikata i izbjegao otkrivanje u pokušaju MITM (man-in-the-middle) napada.

3.4. HTTP Strict Transport Security (HSTS)

Većina web preglednika će upozoriti korisnika u slučaju kada pokuša pristupiti nekoj web stranici koristeći HTTP vezu. Ako neka web stranica dopušta preusmjeravanje korisnika prilikom pristupanja preko HTTP veze na HTTPS vezu, on može u tom slučaju u početku komunicirati preko ne-kriptirane verzije web stranice prije samog preusmjeravanja. Na primjer ako korisnik upiše adresu *http://www.primjer.com* ili samo *primjer.com*. Ovakav tip spajanja na neku stranicu dovodi korisnika u potencijalnu opasnost od MITM napada, gdje napadač može proces preusmjeravanja iskoristiti tako što bi korisnika usmjerio na neku zlonamjernu stranicu umjesto na sigurnu verziju originalne web stranice koju je htio posjetiti.[37]

Web stranice i usluge moraju omogućiti HTTP Strict Transport Security (HSTS) kako bi spriječili ovakav tip eksploatacije. HSTS je mehanizam koji štiti korisnike tako što osigurava da njihovi web preglednici mogu pristupiti njihovoj web stranici samo putem osigurane HTTPS veze. Web preglednik će pri prepoznavanju da je na nekoj domeni omogućen HSTS učiniti dvije radnje:

- uvijek će koristiti *https://* vezu, čak i kada korisnik ukuca u adresnu traku *http://* ili ako pritisne hipervezu na kojoj piše *http://*
- onemogućuje korisnicima da ignoriraju upozorenja o nevažećim ili isteklim TLS certifikatima

[35] Electric Frontier Foundation, The EFF SSL Observatory, URL: https://www.eff.org/observatory (3.9.2017.)

[36] SoK: SSL and HTTPS: Revisiting Past Challenges and Evaluating Certificate Trust Model Enhancements, URL: http://ieeexplore.ieee.org/document/6547130/ (3.9.2017.)

[37] Mozilla Foundation. Strict-Transport-Security. 7.6.2017.
URL: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Strict-Transport-Security (3.9.2017.)

Prvi put kada netko pristupi nekoj web stranici koristeći HTTPS vezu, njihovom web pregledniku će biti uzvraćeno Strict-Transport-Security zaglavlje koje u sebi ima naveden rok trajanja koliko dugo će neki web preglednik pamtiti da tu stranicu otvara samo putem HTTPS veze. Web preglednik će tu informaciju sačuvati tako da kad se ubuduće bude spajao na tu web stranicu koristeći HTTP vezu, automatski će se umjesto toga spojiti HTTPS vezom. Kad istekne rok trajanja naveden u zaglavlju, idući put kad netko bude pokušao pristupiti toj istoj web stranici koristeći HTTP vezu, neće se automatski preusmjeriti na HTTPS.

Svaki put kada server pošalje Strict-Transport-Security zaglavlje web pregledniku, preglednik će ažurirati rok trajanja za tu stranicu, kako bi stranica mogla osigurati da rok trajanja nikad ne istekne.

3.4.1. HSTS predučitavanje

Kako bi korisnik iskoristio prednost HSTS mehanizma, njihov web preglednik mora barem jednom učitati HSTS zaglavlje domene koju posjećuju. Korisnici nisu zaštićeni sve dok prvi put ne uspostave sigurnu vezu s određenom domenom. U većini slučajeva, nikad ni ne dođe do prvog posjeta. Mnoge web stranice preusmjeravaju s *http://domena.com* na *https://www.domena.com* ili čak koriste druge stranice kao izvor pa preusmjeravaju s *http://izvor.com* direktno na *https://destinacija.com*.[38]

U svakom slučaju, *https://domena.com* ni jednom nije posjećena, što znači da korisnici nikad neće preko svog web preglednika učitati *includeSubDomains* smjernicu koja se odnosi na cijeli server. Googleov sigurnosni tim uvodi „HSTS listu za predučitavanje“ kako bi riješili ovaj problem. To je lista domeni čiji je HSTS automatski uključen, čak i nakon prvog posjeta. Firefox, Safari, Opera i Edge su također usvojili ovu listu, što ju čini zastupljenom na svim većim web preglednicima.

Iz svega navedenog možemo zaključiti da je HSTS mehanizam veoma koristan u službi zaštite korisnika i njihove veze na mreži. HSTS donekle služi kao garancija da će veza između nekog korisnika i servera uvijek biti zaštićena odnosno da će se provoditi preko HTTPS veze.

[38] The HTTPS-Only Standard. HTTP Strict Transport Security. URL: https://https.cio.gov/hsts/ (3.9.2017.)

3.5. „HTTPS Everywhere“

„HTTPS Everywhere“ (HTTPS posvuda) je besplatna, open source ekstenzija za Mozilla Firefox, Google Chrome i Opera web preglednike koja omogućava korisnicima da se automatski spajaju preko HTTPS veze na neki server umjesto HTTP vezom, ako određeni server podržava HTTPS vezu.[39]

„HTTPS Everywhere“ je nastao kolaboracijom između Projekta „Tor“ i Electronic Frontier Foundation (EFF). Mnoge web stranice nemaju još dovoljno razvijenu podršku za uspostavljanje veze isključivo HTTPS vezom, što otežava njihovo korištenje. Neke stranice koje podržavaju HTTPS i dalje imaju HTTP kao zadanu veze za spajanje. „HTTPS Everywhere“ ekstenzija uklanja taj problem tako što forsira web preglednik da učita neku stranicu HTTPS vezom, ako je moguće.

„HTTPS Everywhere“ štiti korisnika samo ako posjećuje enkriptirane web stranice. Na stranicama koje podržavaju HTTPS vezu će korisnika automatski spojiti HTTPS enkriptiranom vezom na svim dijelovima servera koji podržavaju takvu vezu. Na primjer, ako nečiji davatelj usluga e-pošte ne podržava HTTPS, „HTTPS Everywhere“ ne može ništa učiniti kako bi osigurao vezu.[40]

„HTTPS Everywhere“ ovisi isključivo o razini zaštite web stranice koju korisnici posjećuju. On je zaslužan za aktivaciju sigurnosnih svojstava neke stranice, ne može ih stvoriti ako već nisu ugrađeni u server.

3.6. Google: HTTPS kao signal za rangiranje

Google smatra da je sigurnost na webu jedan od njihovih top prioriteta, stoga su mnogo investirali u razvoj sigurnosti njihovih usluga kao što su Gmail, Google Search, Google Drive i dr. tako što se koriste HTTPS vezom kao zadanim načinom veze.[41]

---

[39] Electronic Frontier Foundation. HTTPS Everywhere. URL: https://www.eff.org/https-everywhere (3.9.2017.)

[40] Electronic Frontier Foundation. HTTPS Everywhere FAQ. URL: https://www.eff.org/https-everywhere/faq (3.9.2017.)

[41] Google Webmaster Central Blog. HTTPS as a ranking signal.
URL: https://webmasters.googleblog.com/2014/08/https-as-ranking-signal.html (3.9.2017.)

U kolovozu 2014. godine Google je započeo proces rangiranja web stranica ovisno o tome koriste li se enkriptiranom HTTPS vezom. Stranice koje koriste HTTPS češće bi se pojavljivale u rezultatima Google Search pretraživanja u odnosu na one koje nisu prisvojile HTTPS.

Google je u siječnju 2017. godine odlučio sve HTTP stranice koje traže od korisnika unos zaporki ili osobnih brojeva kreditnih kartica označiti kao nesigurne i nepouzdane stranice. Ovo rangiranje je dio njihovog dugoročnog plana da sve stranice koje se ne koriste HTTPS vezom budu označene kao nesigurne. Korisnici Google Chrome web preglednika koji pokušaju pristupiti jednoj od ovakvih stranica bit će upozoreni o nesigurnoj stranici.

Zaključak

Današnji internet kao svjetska računalna informacijska mreža sastavljena je od velikog broja manjih međusobno povezanih računalnih mreža koje omogućavaju prijenos informacija. U prijenosu podataka postoji opasnost zlouporabe informacija. Radi toga potrebno je korisnicama osigurati siguran prijenos podataka. HTTPS je protokol koji može pružiti krajnjim korisnicima povjerljivost, integritet poruke i sigurno ovjeravanje autentičnosti. Ovaj protokol enkriptira podatke tijekom korisničkog prijenosa koristeći pritom neku inačicu TLS protokola, čime ostvaruje razumnu razinu zaštite od napadača. Međutim ona nije dovoljna, zaštita podataka s HTTPS protokolom je u stalnom razvoju i u suradnji s web pretraživačima i agencijama za izdavanje certifikata te još uvijek ima dovoljno prostora za unapređenje visoke razine zaštite. HTTPS protokol je opravdao status vodećeg protokola koji može danas osigurati sigurnost digitalne komunikacije.

Literatura

1. Čop, Julian. HTTP/2 - protokol prilagođen modernom webu : završni rad. Rijeka: Julian Čop, 2016.
2. Panian, Željko. Informatički enciklopedijski rječnik. - Sv.1 : @-L / Željko Panian. Zagreb : Europapress holding-Jutarnji list, 2005.
3. Comodo CA. What is SSL (Secure Sockets Layer)?. URL: https://www.instantssl.com/ssl.html (3.9.2017.)
4. Electric Frontier Foundation. The EFF SSL Observatory. URL: https://www.eff.org/observatory (3.9.2017.)
5. Electronic Frontier Foundation. HTTPS Everywhere. URL: https://www.eff.org/https-everywhere (3.9.2017.)
6. Goldberg, Ian; Wagner, David. Randomness and the Netscape Browser. How secure is the World Wide Web?. URL: https://people.eecs.berkeley.edu/~daw/papers/ddj-netscape.html (3.9.2017.)
7. Google Webmaster Central Blog. HTTPS as a ranking signal. URL: https://webmasters.googleblog.com/2014/08/https-as-ranking-signal.html (3.9.2017.)
8. Google Webmaster Central Blog. Moving towards a more secure web. URL: https://security.googleblog.com/2016/09/moving-towards-more-secure-web.html (3.9.2017.)
9. Grigorik, Ilya. High Performance Browser Networking. 2013. URL: https://hpbn.co/ (3.9.2017.)
10. IBM Knowledge Center. An overview of the SSL or TLS handshake. URL: https://www.ibm.com/support/knowledgecenter/en/SSFKSJ_7.1.0/com.ibm.mq.doc/sy10660_.htm (3.9.2017.)
11. Internet World Stats. Internet growth statistics. URL: http://www.internetworldstats.com/emarketing.htm (3.9.2017.)
12. Kangas, Erik. The LuxSci FYI Blog. URL: https://luxsci.com/blog/ (3.9.2017.)

13. Learn Cryptography. What Are Hash Functions. URL: https://learncryptography.com/hash-functions/what-are-hash-functions (5.9.2017.)

14. Microsoft TechNet. Understanding TLS Certificates. URL: https://technet.microsoft.com/en-us/library/aa998840(v=exchg.141).aspx (3.9.2017.)

15. Microsoft TechNet. What is TLS/SSL?. URL: https://technet.microsoft.com/en-us/library/cc784450(v=ws.10).aspx (3.9.2017.)

16. Mozilla Foundation. Strict-Transport-Security. 7.6.2017. URL: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Strict-Transport-Security (3.9.2017.)

17. Mujarić, Eldis. Računalne mreže. URL: http://mreze.layer-x.com (3.9.2017.)

18. Nacionalni CERT u suradnji s LS&S. Diffie-Hellman protokol. URL: http://www.cis.hr/www.edicija/LinkedDocuments/NCERT-PUBDOC-2009-12-284.pdf (5.9.2017.)

19. SoK: SSL and HTTPS: Revisiting Past Challenges and Evaluating Certificate Trust Model Enhancements. URL: http://ieeexplore.ieee.org/document/6547130/ (3.9.2017.)

20. Symantec Corporation. What is an SSL Certificate? The Ultimate Guide. URL: https://www.symantec.com/page.jsp?id=ssl-information-center (3.9.2017.)

21. TechTarget. RSA algorithm (Rivest-Shamir-Adleman). URL: http://searchsecurity.techtarget.com/definition/RSA (5.9.2017.)

22. The HTTPS-Only Standard. URL: https://https.cio.gov/ (3.9.2017.)

Sažetak

U ovom završnom radu opisan je HTTPS protokol, koji se u stvari sastoji od druga dva protokola, a to su HTTP i TLS. HTTP je najkorišteniji internetski protokol na svijetu koji služi za prijenos hiperteksta, slika, video zapisa, xml datoteka i drugih tipova podataka putem weba. TLS protokol nastaje s ciljem da se zaštiti komunikacija i razmjena podataka putem mreže. Ovaj protokol je dizajniran s ciljem da svojim korisnicima pruži tri glavne usluge, a to su: ovjeravanje autentičnosti, enkripcija podataka i očuvanje integriteta poruke. TLS uspostavlja kriptološki siguran podatkovni kanal između dvaju računala takozvanim TLS „rukovanjem“, koristeći se asimetričnom kriptografijom. Zajedno ova dva protokola čine HTTPS protokol koji se smatra najsigurnijim protokolom za obavljanje sigurnih transakcija putem weba.

Ključne riječi: HTTPS, HTTP, TLS, protokol, enkripcija

Abstract

This paper describes the HTTPS protocol which actually consists of two other protocols, namely HTTP and TLS. The HTTP is the most widely used internet protocol for transfering hypertext, images, videos, xml files and other types of data over the web. The TLS protocol was created with the purpose of protecting communication and exchange of data over the web. This protocol is designed to provide three essential services to its users: authentication, encryption and data integrity. TLS establishes a cryptographically secure data channel between two computers with a so called TLS „handshake“ while using asymmetric cryptography. Together, these two protocols make up the HTTPS protocol which is considered the safest protocol for secure transactions over the web.

Key words: HTTPS, HTTP, TLS, protocol, encryption