Зборник радова Факултета техничких наука, Нови Сад

UDK: 004.42:004.738.12
DOI: https://doi.org/10.24867/15BE37Dragisic

УПОТРЕБА TLS ПРОТОКОЛА ЗА БЕЗБЕДНУ КОМУНИКАЦИЈУ У SPRING РАДНОМ ОКВИРУ

USING TLS PROTOCOL FOR SECURE COMMUNICATION IN THE SPRING FRAMEWORK

Јелена Драгишић, *Факултет техничких наука, Нови Сад*

Област – ПРИМЕЊЕНЕ РАЧУНАРСКЕ НАУКЕ И ИНФОРМАТИКА

Кратак садржај – *Обрада особина TLS протокола, начина успостављања и прекида TLS везе. Акценат је стављен на HTTPS протокол, где су описане особине протокола, а начин функционисања је приказан кроз демонстрацију комуникације клијента и сервера која се одвија преко TLS везе.*

Кључне речи: *TLS, SSL, HTTPS, протокол*

Abstract – *Process the features of the TLS. Emphasis will be placed on the HTTPS protocol, where the properties of the protocol will be described, and the way how it works will be shown through the demonstration where the communication between client and server will be happened above TLS.*

Keywords: *TLS, SSL, HTTPS, protocol*

1. УВОД

Задатак овог рада јесте обрада *Transport Layer Security (TLS)* протокола. *TLS* протокол је развијен од стране међународне организације за стандарде, под називом *Internet Engineering Task Force (IETF)*, а прва верзија протокола објављена је 1999. године. Најновија верзија јесте *TLS* 1.3 и објављена је 2018. године. *TLS* протокол је сигурносни мрежни протокол који омогућава сигурну комуникацију на мрежи, сигурност података који се размењују и очување њиховог интегритета. Подржава и аутентификацију обе стране које учествују у комуникацији. Основна употреба овог протокола јесте шифровање комуникације између веб апликација и сервера.

У другом поглављу дат је кратак увод у криптографске протоколе, тачније описани су одређени безбедносни појмови који се користе у овом раду. Треће поглавље рада се односи на презентацију *TLS* протокола, начина успостављања и прекида сигурне комуникационе везе, а такође су описане предности и мане овог протокола. Комбинација *TLS*-а са *HTTP* протоколом презентована је у четвртом поглављу, које садржи и демонстрацију примене *TLS* протокола. Пето поглавље садржи смернице даљег развоја и представља закључак овог рада.

НАПОМЕНА:
Овај рад проистекао је из мастер рада чији ментор је био доцент др Жељко Вуковић.

2. УВОД У КРИПТОГРАФСКЕ ПРОТОКОЛЕ

Криптографски протоколи представљају дефинисан скуп правила за решавање одређених ситуација као и спречавање потенцијалних проблема при самој комуникацији. *TLS* је управо криптографски протокол. С обзиром на то да ови протоколи почивају на коришћењу одређених криптографских техника, у овом поглављу је направљен увод у криптографске протоколе и објашњена су значења неких основних појмова који су коришћени кроз рад.

Пре свега, криптографија заправо представља научну дисциплину која обезбеђује очување тајности порука. Појам енкрипције се односи на шифровање података, односно превођење тих података из људски, разумљивог облика у низ насумичних карактера који не носе семантику. Декрипција је обрнут поступак, где се шифрован текст преводи у оригиналан. Сертификат представља на неки начин личну карту учесника комуникације. Обезбеђује могућност аутентификације, односно потврду идентитета самог учесника. Иза сваког валидног сертификата треба да стоји одређено сертификационо тело, које је тај сертификат заправо издало и потписало [1].

3. TLS ПРОТОКОЛ

TLS протокол је у *Open System Interconnection (OSI)* моделу смештен између петог и шестог слоја. Ради на слојевима изнад траспортних протокола попут *TCP* проткола, а испод апликацијских попут *HTTP* протокола [2]. Положај *TLS* протокола приказан је на слици 1.

Слика 1. *Положај TLS протокола*

TLS протокол је криптографски протокол који успоставља сигурну, комуникациону сесију између

пријемне и предајне стране. Основни задатак јесте очување приватности током комуникације и размене података као и заштита интегритета података који се траснпортују.

TLS је двослојни проткол, тачније састоји се из два подпроткола и то протокола руковања и протокола записа [3] . Током механизма руковања пријемна и предајна страна се договарају о начину шифровања комуникације, врши се аутентификација страна као и размена кључева. Затим, на договорен начин се криптују и на тај начин обезбеђују подаци који се преносе кроз *TLS* тунел, претходно успостављен између комуникационих страна.

Протокол руковања обезбеђује сигурност везе. Та веза је приватна а шифровање података који се размењују постижемо коришћењем симетричних криптографских алгоритама. Кључеви који се користе за енкрипцију генеришу се посебно за сваку конекцију у оквиру протокола руковања. Пре саме енкрпције подаци се компримују коришћењем договореног алгоритма. *TLS* веза је такође поуздана јер се при примању поруке проверава њен интегритет коришћем хеш вредности о чему ће више речи бити у наставку. Дакле, протокол руковања омогућује генерисање сигурне сесије између клијента и сервера, а протокол записа сигурну размену података. [4]

3.1. *TLS handshake* протокол

Успостава *TLS* комуникације између пријемне и предајне стране започиње процедуром руковања. Та процедура се назива *TLS handshake* протокол, а омогућава договарање клијента и сервера око пакета шифровања који ће бити коришћен у наставку комуникације. Пакет шифровања обухвата алгоритам за шифровање као и криптографске кључеве. Омогућена је и аутентификацијуа комуникационих страна, тачније потврду њиховог идентитета. Најчешће је то случај са серверском страном.

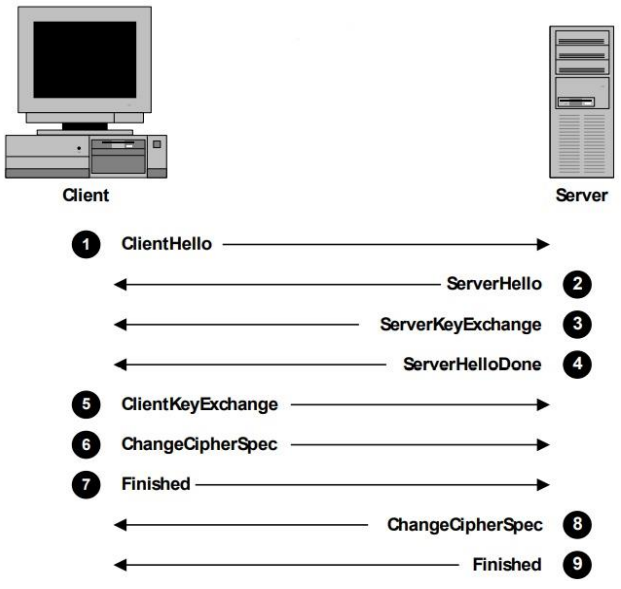
Клијент иницира комуникацију слањем *ClientHello* поруке серверу са којим жели да комуницира. У оквиру ове поруке специфира до које верзије има подршку за *TLS* проткол. Треба узети у обзир то да клијент подржава и раније верзије протокола закључно са оном која је специфирана. Порука садржи *RandomNumber* поље које садржи произвољну вредност дужине 32 бајта која се користи као основа за даља различита криптографска израчунавања. *SessionID* поље поруке је у иницијалној процедури руковања празно, али се користи у надоградњама овог механизма. Списак криптографских алгоритама као и дужине кључева које клијентска страна може да подржи документоване су у пољу *CipherSuites*. *CompressionMethods* поље садржи списак различитих компресионих метода које је са стране клијента могуће користити.

Након што прими ClientHello поруку, сервер одговара слањем ServerHello поруке. У тој поруци поставља верзију TLS протокола по ком ће се даља комуникација заснивати. Верзија протокола за коју се сервер одлучи мора бити у складу са верзијом коју је специфирао клијент у ClientHello поруци. Затим се постављају RandomNumber и SessionID поља. CipherSuite поље садржи неки од криптографских алгоритама са списка који је специфиран у CipherSuites пољу клијентске поруке. Сервер је та страна која доноси коначну одлуку о начинима енкрипције података током даље комуникације. На тај начин се комуникационе стране усклађују око пакета шифровања који ће користити. ServerKeyExchange порука садржи јавни кључ сервера који ће се користити за даљу енкрипцију, а формат кључа зависи од изабраног пакетог шифровања. Након ње, шаље се и ServerHelloDone порука која наговештава клијенту да може прећи у наредну фазу комуникације.

Након што добије јавни кључ од сервера, клијент је у могућности да пошаље ClientKeyExchange поруку. У овој поруци шаље податке које енкриптује јавним кључем сервера, а исте може само прочитати сервер ком је ова порука намењена.Међу подацима у овој поруци је и кључ сесије којим ће се даље шифровати подаци у овој комуникационој сесији.

Након слања *ClientKeyExchange* поруке, клијент серверу шаље и *ChangeCipherSpec* поруку којом указује на то да је успостављен или промењен пакет шифровања и да је даља комуникација обезбеђена на договорен начин. Клијент затим шаље и *Finished* поруку која супротној страни омогућава проверу испуњености договорених, сигурносних параметара у иницијалној фази комуникације.

Сервер одговара клијенту слањем *ChangeCipherSpec* и *Finished* поруке. Процес руковања се на овај начин успешно завршава, а даља комуникација је енкриптована. Поруке које се размењују током овог процеса, као и редослед истих, приказан је на слици 2.



Слика 2. *TLS handshake [5]*

Претходни модел успостављања *TLS* везе често је неопходно прошити. Тада је потребно обезбедити аутентификацију учесника комуникације, односно потврду њиховог идентитет. У већини случајева то је неопходно учинити за серверску страну. Потврду идентитета клијента могуће је постићи и провером самих података које клијент шаље током комуникације са сервером.

3.2. *TLS* протокол записа

TLS је двослојни проткол, тачније састоји се из два подпроткола и то протокола руковања и протокола записа [2].

TLS протокол записа омогућава сигурну размену података кроз успостављен *TLS* комуникациони канал. Сигурна размена почива на енкрипцији података употребом кључа и криптографског алгоритма. Користи се договорени пакет шифровања, који представља производ претходно описане процедуре руковања. Шифровани подаци се даље предају протоколима нижег нивоа на транспорт, конкретно протоколима транспортног слоја. Између осталог, протокол записа омогућава и проверу интегритета порука и на тај начин спречава нежељену измену података у транзиту.

3.3. Затварање TLS сесије

Када дође до тренутка затварања сесије, неопходно је исту експлицитно затворити. То се постиже слањем *ClosureAlert* поруке супротној комуникационој страни, док друга страна након пријема врши ретрансимисију поруке. На тај начин потврђено је затварање комуникационе сесије [5].

3.4. Предности и мане

Поред основних функција разматрано протокола, једна од главних предности јесте независност протокола која се огледа у употреби протокола вишег нивоа над *TLS* -ом без додатних, неопходних прилагођавања.

TLS протокол омогућава корисницима да безбедно приступе удаљеним ресурсима. Како би истима корисник приступио, неопходно је да се аутентификује. За тај поступак може се искористити TLS конекција. Приступ подацима је могуће ограничити.

Анализом *TLS* проткола долази се до закључка да је временски најзахтевнија фаза процес руковања између клијента и сервера [6]. Неопходна је размена одређеног броја порука у циљу постизања договора око пакета шифровања и успостављања сигурне, комуникационе сесије. Свако ново успостављање TLS везе између пријемне и предајне стране захтева понављање поступка руковања. То свакако представљаману протокола, али временом су креирани различити начини превазилажења ових и сличних недостатака.

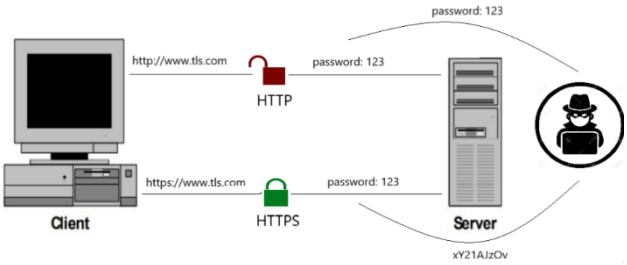
Такође, кеширање сесија на серверу представља потенцијални меморијски проблем за који се опет користе различите технике превазилажења

4. HTTPS ПРОТОКОЛ

HyperText Transfer Protokol (HTTP) је мрежни протокол апликативног слоја *OSI* референтног модела. Као идеја настаје 1989 године у Церну, али је прва верзија овог протокола *HTTP* V0.9 документована 1991. године. Архитектура протокола је клијент-сервер, односно сама комуникација се одвија између клијента и сервера, и то следећим низом трансакција. Клијент шаље поруку захтева, а сервер затим враћа поруку одговора. Порука захтева и порука одговора представљају два типа *HTTP* порука које се шаљу у отвореном тексту на мрежи. Свако ко прати комуникацију између две стране овакав вид отворених порука може да разуме. Ово представља озбиљан безбедносни проблем када се шаљу осетљиви подаци. Сам протокол се користи за обраду, приказивање и испоруку веб страница.

Како би се такви безбедносни ризици смањили, долази до развоја *HTTPS* протокола који представља шифровану, односно сигурну верзију *HTTP* протокола. Употребом *HTTPS* протокола обезбеђује се аутентификација крајњих, комуникационих тачака и поверљивост саме комуникације [7]. Овај протокол заправо представља употребу *HTTP* протокола над *TLS* -ом, где се *TLS* протокол користи за енкрипцију података у оквиру *HTTP* захтева и одговора. На овај начин малициозна трећа страна неће бити у могућности да види отворени текст већ само низ насумичних карактера.

Странице на вебу које користе *HTTP* протокол у оквиру свог *URL*-а садрже идентификатор проткола означен са “HTTP://“, док странице које користе *HTTPS* протокол садрже идентификатор “HTTPS://“. Поређење ова два протокола приказано је на слици 3.



Слика 3. *Приказ HTTP и HTTPS протокола*

Када се започиње ова врста комуникације, *HTTP* клијент се понаша као тлс клијент. Неопходно је иницирање комуникационе сесије са серверском страном претходно описаним протоколом руковања. Након што се успостави сигурни тлс комуникациони канал, могуће је слање првог *HTTP* захтева [8]. Сви подаци који се преносе путем ових захтева су енкриптовани, док се њихов интегритет проверава. Битан корак за успостављање сигурне *HTTPS* везе јесте провера идентитета сервера. Због тога најважнији део подешавања окружења јесте коришћење сертификата.

4.1. Spring радни оквир

Spring је open source развојно окружење које се користи за развој апликација коришћењем *Java* програмског језика. Садржи екстензију *Spring Boot* која са собом носи бројне погодности у виду руковања конфигурацијом, како би се скренуо фокус на развој саме апликације.*Spring Boot framework* је коришћен за потребе демонстрације истраживања.

4.2. Демонстрација истраживања

За потребе задатка направљена је *Spring Boot* апликација која неће имати никакве

функционалности, већ ради једноставности само један endpoint.

Како бисмо омогућили *HTTPS* у нашој апликацији било је неопходно креирање сертификата.

Сертификат је генерисан путем терминала за шта нам је био потребан алат *keytool*, који можемо преузети са интернета [9] или не морамо уколико имамо инсталиран JDK (*Java Development Kit*), јер се поменути алат налази у оквиру JDK-а. Сертификате можемо генерисати користећи *Let's encrypt* [10]. То је непрофитан *certificate authority (CA)* који нам пружа X.509 сертификате [11] за *Transport Layer Security (TLS)*, без накнаде. То је највећи светски *CA* чији је главни циљ да вебсајтови постану сигурни и да користе *HTTPS*.

Након што се обезбеди сертификат извршена је конфигурација *HTTPS*-а у *Spring Boot*-у. Поред тога било је потребно да извршимо редирекцију на *HTTPS*, уз помоћ *Spring Security*-а. Када користимо *Spring Security* тада можемо да конфигуришемо сервер тако да се аутоматски блокира било који *request* који долази са небезбедног *HTTP* канала и редиректује на *HTTPS*.

5. ЗАКЉУЧАК

Циљ рада био је да се осврнемо на *TLS* протокол, на његове особине а пре свега на начин успостављања, као и прекида конекције. Размотрене су предности, али свакако и мане протокола. Акценат је стављен на *HTTPS* протокол, чије су особине такође описане, а начин функционисања је приказан кроз демонстрацију истраживања.

Експанзијом комуникације на мрежи, сама сигурност података који се размењују добија велики значај. Интернет мрежа постаје водећа инфраструктура када је у питању комуникација, трговина и генерални приступ информацијама. Подаци који се преносе се у свом изворном облику су под великим безбедносним ризиком. Заштиту таквих података који се транспортују путем мреже омогућава управо примена *TLS* протокола. Данас, готово сви веб претраживачи и сервери имају уграђену подршку за овај протокол у циљу обезбеђивања сигурног преноса података између комуникационих страна.

6. ЛИТЕРАТУРА

[1] Uvod u kriptografiju – Bezbednost u sistemima elektronskog poslovanja, Prof. dr Goran Sladić

[2] A survey on MITM and its countermeasures in the TLS handshake protocol, Seung-Woo Han, Hyunsoo Kwon, Changhee Hahn, Dongyoug Koo, Junbeom Hur , Department of Computer Science and Engineering, Korea University, Seoul, Republic of Korea

[3] An Analysis of TLS Handshake Proxying, Douglas Stebila, Nick Sulivan

[4] *TLS protokol CCERT-PUBDOC-2009-03-257*, Hrvatska akademska i istraživačka mreža

[5] SSL & TLS Essentials Securing the Web, Stephen A. Thomas

[6] Design of an enhancement for SSL/TLS protocols, Ashraf Elgohary, Tarek S. Sobh, M. Zaki

[7] The Cost of the "S" in HTTPS, David Naylor, Alessandro Finamore, Ilias Leontiadis, Yan Grunenberger, Marco Mellia, Maurizio Munafò, Konstantina Papagiannaki, Peter Steenkiste

[8] *RFC 2818– HTTP over TLS*, E. Rescorla

[9] Spring Initializr, *(https://start.spring.io/)*

[10] Let's encrypt, *(https://letsencrypt.org//)*

[11]X.509 сертификат, (*https://en.wikipedia.org/wiki/X.509*)

Кратка биографија:

Јелена Драгишић рођена је у Новом Саду 1998. године. Мастер рад на Факултету техничких наука из области Електротехнике и рачунарства – Примењене рачунарске науке и информатика одбранила је 2021. године.

контакт: jelenaa.dragisic@gmail.com