Фундаментална киберсигурност за Уеб услуги(Web Services)

Божидар Николаев Захов

фак. № 121217024, група 37

e-mail: bzahov1998@gmail.com

LinkedIn: https://www.linkedin.com/in/bzahov98/

Ключови думи:

Web Services, Security, XML, SOAP, REST, Web Services requirements, Transport Layer Security (TLS), Encryption, Authentication, and Integrity

1. Въведение

В тази статия ще разгледаме основните причини, защо е важно, но и как да поддържаме нашите системи, използващи Уеб услуги да бъдат надеждни, добре подсигурени и да отговарят на високите изисквания за сигурност на бизнеса.

Ако една система е много добре изпълнена, от към бизнес функционалност, покрива всички изисквания на клиента, но не е добре защитена и позволява злонамерена манипулация отвън, което поставя личните данни и устойчивостта на системата под риск, то тя ще носи повече вреди, отколкото ползи.

В такъв случай бизнес клиентите и потребителите няма да използват даденото решение, колкото и полезна работа да върши, а ще предпочетат по-слаб от към функционалност продукт, но по-добре защитен, от към външни кибер заплахи, тъй като ще предпочетат техните данни, да бъдат в безопасност.

1.1. Какво е Уеб Услуга (Web Service)

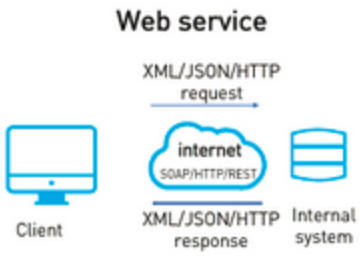
Уеб услугата (услуга) е програма, която организира взаимодействие между сайтове. Те включват всякакъв вид софтуер, който предоставя стандартизиран уеб протокол (web protocol) , като HTTP или HTTPS, който да взаимодейства, комуникира и обменя данни, посредством интернет съобщения, обикновено базирани на XML. [4]

С други думи, те са XML-базирани даннообменящи системи, които използват интернет, за да поддържат комуникации и интерфейси от A2A тип "приложение към приложение" (application-to-application) – което означава, че приложения на различни места в мрежата, могат да бъдат интегрирани, по такъв начин, че да работят като части, от една сложна софтуерна система. [3][4]

Ето един пример за това как работи:

Една уеб услуга стои между две среди, използващи различен технологичен стек от езици (например java, .net (C#) или PHP приложения) и предоставя начин, тези приложения да комуникират, помежду си, чрез мрежа. Например от едната страна, е java приложение, което взаимодейства с .net и/или PHP приложения, от другата страна, чрез уеб услуга, посредством независим език като XML. [3]

Примери за такива приложения са автоматизирани бизнес транзакции и директен достъп от настолни (Desktop) и мобилни приложения (Mobile) (без браузър) до системи за резервации, стокови борси, и системи за следене на поръчки. [3]



Фигура. 1 Устройство на уеб услуга[5]

1.2. Какви са различните видове уеб услуги?

От основите на уеб услугите са се оформили няколко ключови стандарта:

- XML (Extendable Markup Language),
- UDDI (Universal Description, Discovery, and Integration).
- SOAP (Simple Object Access Protocol)
- WSDL (Web Services Description Language) [3]

1.2.1. За отдалечено извикване на процедура XML-RPC

Това е стандарт за отдалечено извикване на процедура XML-RPC (Remote Procedure Call), който използва възможно най-опростеният XML протокол, чрез който може да се обменят данни, между широк набор от устройства в мрежата. Той използва HTTP за бърз и лесен трансфер на данни и комуникация между сървъра и клиента. [4]

1.2.2. За универсално описание,откриване и интегриране UDDI

Стандартът за универсално описание, откриване и интегриране UDDI (Universal Description, Discovery, and Integration) е XML-базиран и се използва за подробно описание, публикуване, и откриване на уеб услуги.

Той представлява прост, интернет регистър за бизнесите по света.

Целта е да се опрости дигиталните транзакции и електронната търговия (e-commerce), между системите на различните компании. [4]

1.2.3. SOAP

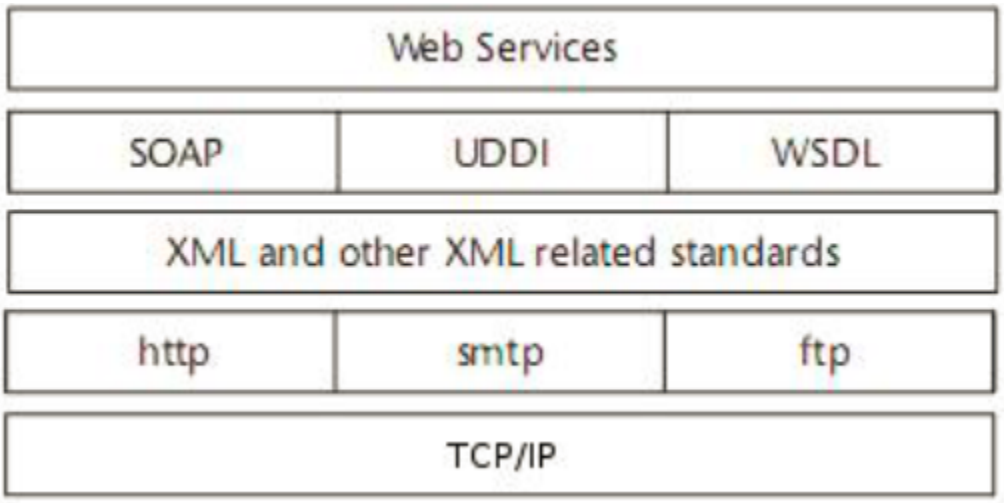
SOAP (Simple Object Access Protocol) е XML-базиран протокол за уеб услуги, чрез който се обменят данни и документи през HTTP или SMTP (Simple Mail Transfer Protocol). Той позволява на независими процеси, работещи на различни системи, да комуникират помежду си, посредством XML. [4]

1.2.4. REST

Акронимът REST (*Representational State Transfer*) означава: "Представителен държавен трансфер". Той осигурява комуникация и свързаност между устройствата и интернет базирани ресурси, които се достъпват чрез API

интерфейсни заявки. Повечето RESTful услуги използват HTTP като поддържащ протокол.

*API - (Приложно-програмен интерфейс/ Application Programming Interface)[4].



Фигура. 2 Протоколен стек на Web услугите [6]

2. Сигурност при уеб услуги

2.1. Условия за сигурност при уеб услуги (Security requirements)

Автентикационните протоколи представляват последователността от необходими действия, за да протече правилно автентикацията на клиента пред сървъра. Основните изисквания за сигурност на уеб услугите са удостоверяване, упълномощаване, защита на данните и невъзможност за отказ. [1]

2.1.1. Удостоверяване (authentication)

Удостоверяването гарантира, че всеки обект, участващ в използването на уеб услуга - заявителят, доставчикът и брокерът (ако има такъв) - е това, което всъщност твърди, че е[1]. С други думи, освен че клиентът се представя пред сървъра, то и сървърът може да се идентифицира пред клиента. [7]

Удостоверяването включва приемане на идентификационни данни от обекта и утвърждаването им спрямо проверяващата функционалност на сървъра.[1]

2.1.1.1. Ниво на защитата при удостоверяване

Пример за слоеве на защита при удостоверяване.

Идентичността на потребителя се проверява въз основа на идентификационните данни, представени от този потребител, като например[2]:

- Нещо, което човек има, например, идентификационни данни, издадени от доверен орган, като паспорт (реален свят) или смарт карта (ИТ свят)[2].

- Нещо, което човек знае, например споделена тайна, като парола[2].
- Нещо, което представлява човек в реалният свят, например биометричната информация[2].

Използването на комбинация от няколко вида идентификационни данни се нарича „силно“ удостоверяване и е предпоставка за високо ниво на защита[2]***.***

Например използване на карта на банкомат (нещо, което човек има) с ПИН или парола (нещо, което човек знае)[2].

Ако искаме да вдигнем още новото на сигурност, може да добавим и пръстов отпечатък и/или сканиране на лице (нещо, което представлява човек в реалният свят) и *тогава ще имаме наистина много силна идентификация.*

2.1.1.2. Нуждата от еднократна автентикация

Разрастването на големият брой системи за различни продуктивни нужди, довежда потребители и системни администратори пред проблема за поддръжка на съответния голям брой потребителски имена и пароли (или друг тип автентикационни данни) за всеки един потребител. [7]

Например даден потребител може да притежава няколко регистрации - с клиентски номер и някаква парола за някаква организация (например за плащане на сметки през Интернет), с някакво измислено име и парола за сайт за електронна поща и регистрация с ЕГН и някакъв сертификат за някаква банка например. [7]

Осигуряването на коректен достъп до тези системи понякога може да бъде досадна операция, когато при всяка една заявка към някоя от тези системи, трябва да се предоставят автентикационните данни. [7]

2.1.2. Упълномощаване (authorization)

Упълномощаването определя дали доставчикът на услуги е предоставил достъп до уеб услугата на заявителя. По принцип

упълномощаването потвърждава идентификационните данни на заявителя на услугата.

Той определя дали заявителят на услугата има право да извърши операцията, която може да варира от извикване на уеб услугата до изпълнение на определена част от нейната функционалност на сървъра.[1]

2.1.3. Конфиденциалност на информацията (Confidentiality)

Предаването на данни между клиента и сървъра трябва да бъде осъществено, така че ако се подслушва канала да не може да бъде откраднатa информация, която по-късно да бъде използвана за автентикация.

2.1.4. Защита на данните или (non repudiation,Integrity)

Защитата на данните гарантира, че заявката и отговорът на уеб услугата не са фалшифицирани, по време на преноса на информацията по интернет мрежата. Това изисква потвърждаване, както на целостта на данните, така и на поверителността им. [1]

С други думи пристигналата информация на сървъра, е същата, която е изпратена от самият клиент, т.е. не е била променяна по трансферния канал.[7]

За тази цел обикновено се използват цифрови подписи, които клиентът изпраща заедно със съобщението. При пристигане на съобщението, сървъра проверява дали съобщението отговаря на съпътстващия го цифров подпис. [7]

Струва си да се спомене, че защитата на данните не гарантира самоличността на подателя на съобщението, а само че те не са били променяни. [1][2]

- Невъзможност за отричане (nonrepudiation)

Това свойство доказва, че наистина автентикацията е минала, и клиентът или сървърът не може да отрече, че наистина тя е протекла. Или

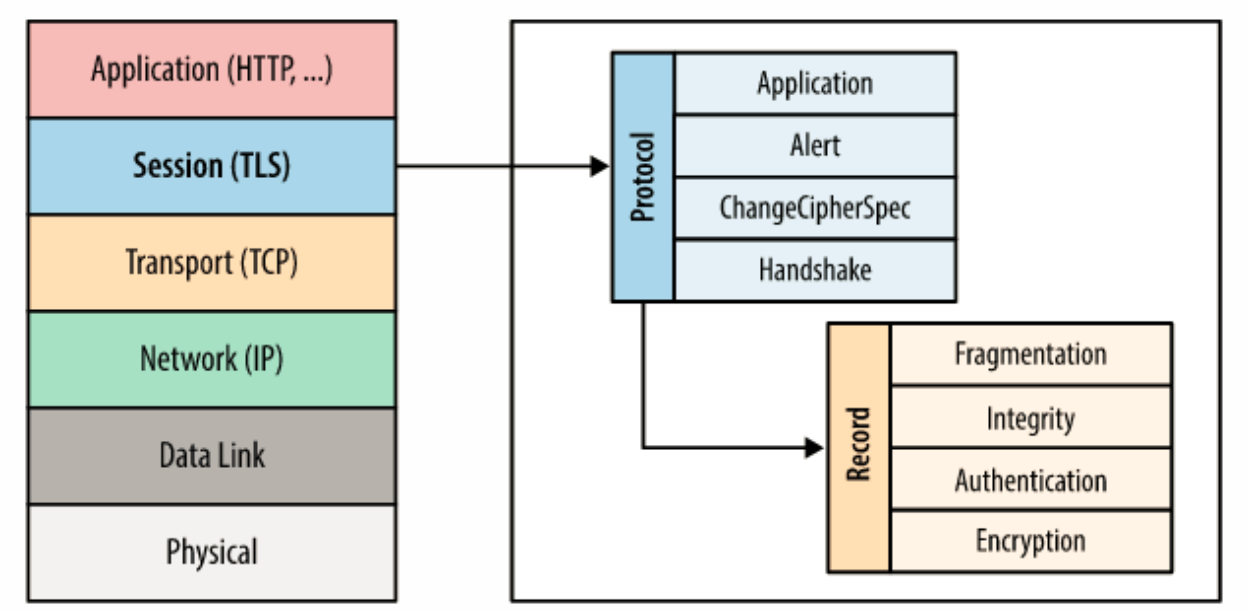
с други думи тя гарантира, че подателят на съобщението е същият като създателя на съобщението[1]. За тази цел отново се използват цифрови подписи.

Например когато плащаме през интернет, ние искаме гаранция, че фирмата, която закупуваме даден продукт, няма да отрече, че той вече е бил платен. [7][2]

- Проблемът с транспортиране на пароли

Голяма част от несигурността при компютърните мрежи идва от факта, че се предават пароли по мрежата (в най-незащитения случай като чист текст).

Това означава, че всеки, който подслушва мрежата може да се сдобие с паролата на даден потребител и да се представя за него. Дори кодиране на паролата, например чрез използване на хеш функции, не може да помогне много. [7]



Фигура. 3 Сигурност на Транспортния Слой - Transport Layer Security (TLS) [10]

2.2. Топ 10 на решаващите фактори за защита

Сега, след като по-горе са разгледани аспектите на защитата на уеб услугите, а и някои от наличните проблеми, в тази глава ще бъдат разгледани десетте, най-важни фактора за сигурност, влияещи, върху внедряването на уеб услуги.

Десетте основни фактора, които определят изискванията за сигурност при използването на уеб услуги, са както следва[1]:

2.2.1. Използва ли се уеб услугата за EAI [1]или B2Bi[2]?

Уеб услугите могат да се използват за два различни домейна:

- интеграция между предприятия (B2Bi) [1].
    - уеб услугите за B2Bi винаги трябва да използват криптиране, а могат да включват и множество нива на удостоверяване
    - Невъзможността за отричане (nonrepudiation) е полезна за уеб услуги в домейна B2Bi
- интеграция на корпоративни приложения (EAI)[1]
    - уеб услугите за EAI трябва да имат едно ниво на удостоверяване и рядко използват криптиране

Изискванията за сигурност за домейна EAI са подмножество на тези за B2Bi, тъй като е много по-лесно да се контролират, управляват, намират, изпълняват и поддържат уеб услуги в интернет, отколкото да се използват през Интернет преминавайки през корпоративната защитна стена (corporate firewall)[1].

Докато уеб услугите за EAI трябва да имат едно ниво на удостоверяване и рядко да използват криптиране, уеб услугите за B2Bi могат да включват множество нива на удостоверяване и винаги трябва да използват криптиране.[1]

И накрая, Невъзможност за отричане (nonrepudiation) е полезна за уеб услуги в домейна B2Bi, тъй като предотвратява злонамерен подател да се откаже по-късно от създаването и изпращането на конкретно

[1] EAI - Интеграцията на корпоративни приложения (Enterprise application integration)

[2] B2Bi - **бизнес-бизнес"** интеграция (business-to-business integration)

съобщение [1]

2.2.2. Каква е целта на уеб услугата?

- Ако уеб услугата предоставя единствено публично достъпни данни, като например прогнозата за времето в даден град, то изискванията за сигурност е възможно да са занижени, спрямо тези за уеб услуга, която предоставя частна бизнес информация, с ограничен достъп[1].

2.2.3. Кои са абонатите на уеб услугата?

- Знанието кои са абонатите на дадена уеб услуга е важно за определяне на функциите и за упълномощаване и удостоверяване[1]

2.2.4. Може ли услугата да бъде извикана през Интернет?

- Уеб услугата ограничена ли е до доверени търговски партньори или може някоя компания, свободно да достъпи уеб услугата през Интернет? [1]
- Това е от решаващо значение за функциите за упълномощаване и удостоверяване на уеб услугата, освен функциите за защита на данните и невъзможността и за отричане(non repudiation) [1]

2.2.5. Колко сигурно е основното приложение?

- Какво ниво на достъп предоставя уеб услугата на основното приложение? [1]
- Достъпът трябва ли да се основава на оторизация и права?[1]
- Колкото по-голям е достъпът до основните приложения, толкова по-големи са изискванията за сигурност при упълномощаването и удостоверяването. [1]

2.2.6. Уеб услугата транзакционно-ориентирана(transaction-oriented) ли е?

- Заплахите за сигурността ще бъдат по-високи, ако транзакцията е разпределена между множество обекти.[1]

2.2.7. Какъв протокол се използва?

- Какъв мрежов протокол обработва удостоверяването и предаването на данни между заявителя на услугата и доставчика? [1]

- Важно е да се знае дали има нужда от сигурност на данните, тъй като всеки може да проследи, както заявката, така и отговора на уеб услугата, чрез middle man атака, тъй като тези данни ще бъдат пренесени по мрежата като обикновени XML документи. [1]
- Ако това е HTTPS, тогава няма нужда от допълнителни алгоритми за криптиране / декриптиране, тъй като HTTPS го предоставя. [1]

2.2.8. Има ли нужда от проверка на подателя/получателите?

Необходимо ли е да се гарантира, че подателят на заявката за уеб услуга и съобщението за отговор е същото, като създателя на съобщението?[1]

2.2.9. Кой участва в услугата?

- Колко различни субекти участват в използването на уеб услугата; т.е. има ли уеб услугата функция за свързване на обекти(entity-chaining)? Ако има повече от един обект, той ще изисква по-високи функции за защита.[1]

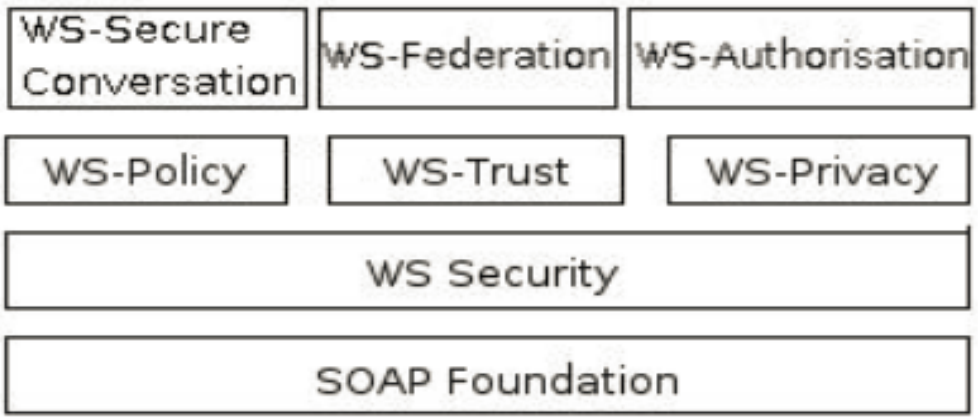
2.2.10. Използва ли се верижно свързване на компоненти(component chaining)?

- Има ли функция за верижно свързване на компоненти (component chaining) в кода за внедряване на уеб услугата? Ако верижното свързване на приложения обхваща корпоративната защитна стена (Corporate firewall), изискванията за сигурност стават по-строги.[1]

2.3. Архитектура на сигурността при Web услугите

Microsoft и IBM съвместно са създали модел за прилагане на политики за сигурност при Web услугите. Тяхната стратегия включва набор от спецификации, способни да осигуряват надеждна среда за предаване на данните,повишена работоспособност при

между-платформена комуникация, която се осъществява през Интернет[6].



Фигура. 4 Спецификация на сигурността при Web услугите[6]

Спецификацията на сигурността, както е показано на *Фигура. 4* се основава на WS-Security (Web Services Security). На същата фигура са показани и два допълнителни слоя, които надграждат WS-Security и предоставят специфична функционалност с предварително дефиниран набор от правила. В тези два слоя се включват набор от спецификации за отделни елементи на сигурността[6]:

2.3.1. WS-Policy

Това е разширение към WS-Security спецификацията, в което се описват политиките на деклариране и използване на дадена

функционалност.[6]

2.3.2. WS-Trust

Спецификацията дефинира рамките за настройка и поддържане на взаимоотношенията на сигурност между страните, участващи в комуникационния процес.[6]

2.3.3. WS-Privacy

Позволява да се добавят организационни модели към дадени Web услуги, за да се позволи имплементирането на конкретни политики.[6]

2.3.4. WS-Authorization

Предоставя процедури и определя изискванията при конфигурирането на комуникацията от край-до-край – определянето на политиките от предходния слой, както и маркерите на сигурността.[6]

2.3.5. WS-SecureConversation

WS-Secure Conversation конкретизира предоставянето на определена функционалност за определянето на идентичността на участниците в комуникацията.[6]

2.3.6. WS-Federation

Позволява да се определят задачите за създаването на сигурна среда за взаимодействие, които ще използват Web услуги.[6]

2.4. Методи за защита на Web услуги

Съществуват три основни механизма за осигуряване на защита [6]:

2.4.1. На ниво транспорт

Защитата на данните се осигурява, само докато се транспортират от една входна точка до друга.

Съществуват три механизма за защита на ниво транспортен слой[6]:

- SSL/TLS,
- първично определяне на идентичността
- Клиентско определяне на идентичността (two way SSL)

2.4.2. На ниво съобщение

- На ниво съобщение защитата се осигурява, като самите съобщения се криптират.[6]
- Тук е възможно различни части от съобщението да бъдат криптирани с различни ключове за различните получатели на съобщението. [6]
- Могат да бъдат използвани цифрови подписи за осигуряване на интегритет на съобщението.[6]
- Три от механизмите които могат да се използват тук са[6]:
    - XML криптиране
    - XML подписване
    - Използване на маркери.

2.4.3. На ниво роли.

- При наличие на потребители изграждането на сигурност включва определяне на идентичността и оторизация. [6]
- Определянето на идентичността е свързано с идентификация[6]:
    - чрез потребителско име и парола
    - чрез сертификат.
- Оторизацията е свързана с определяне на правата, които отделните потребители имат на различните системи[6].
- Най-често използваният механизъм за осигуряване на оторизация е посредством използване на роли[6].

3. Заключение

В днешния дигитално-активен свят, където киберпрестъпленията, кражбите и измамите се увеличават, организациите изискват най-сигурните, точни и надеждни методи за удостоверяване на самоличността, за достъп до данни, физически достъп и обща сигурност.[8]

Подсигуряването на добра защита на Уеб услугите е неделима част от процеса на разработка, експлоатация и поддръжка на, коя да е Уеб система. Когато дадена система е достъпна до публичното пространство, само защитените уеб услуги могат да осигурят оправданата интеграция на продукта, тъй като ползите, които тя предлага на потребителя, би трябвало да надвишават потенциалните рискове. [9]

Сигурността на системите се превърна в предизвикателство с атаки срещу всички фронтове. Уеб услугите са изключително важна част от тези системи и ако дори само едно звено от тях не е достатъчно защитено, то това поставя под изключително сериозен риск, много сфери от нашия онлайн и реален живот, защото тези системи намират все по-голям набор от приложения .

4. Съдържание:

5. Използвана Литература

[1] Gunjan Samtani, “Top 10 Web service security requirements,” *TechRepublic*, 2002. https://www.techrepublic.com/index.php/article/top-10-web-service-security-requirements/

[2] “Understanding Web Services Security Concepts.” - Oracle https://docs.oracle.com/cd/E23943_01/web.1111/b32511/intro_security.htm#WSSEC1500 (accessed Apr. 21, 2021).

[3] Е. Нюкъмър, “Web услуги,” *soft-press.com*. https://www.soft-press.com/book/261/category_28.

[4] “Blog: What Are Web Services? Easy-to-Learn Concepts with Examples,” *Cleo*. https://www.cleo.com/blog/knowledge-base-web-services (accessed Apr. 21, 2021).

[5] Sanjna Verma, MuleSoft Blog, “APIs versus web services,” *MuleSoft Blog*, Jan. 18, 2018. https://blogs.mulesoft.com/dev-guides/apis-versus-web-services/ (accessed Apr. 21, 2021).

[6] Dsn. V-Lab, “WS-Security Implementations.” https://dsnet.tu-plovdiv.bg/website/container/papers/A&I_security.pdf (accessed Apr. 21, 2021).

[7] User, “Microsoft Word - MasterThesis GeorgiDimitrov.” https://research.uni-sofia.bg/bitstream/10506/201/1/MasterThesis%20GeorgiDimitrov.pdf (accessed Apr. 21, 2021).

[8] “S&T - Биометрична идентификация.” http://snt.bg/93908.bg.php (accessed Apr. 21, 2021).

[9] “RixGroenboom-v2.pdf.” https://owasp.org/www-pdf-archive/SecuringWebServices-RixGroenboom.pdf (accessed Apr. 21, 2021).

[10] I. Grigorik, “Networking 101: Transport Layer Security (TLS) - High Performance Browser Networking (O’Reilly),” *O’Reilly*, Oct. 15, 2013.

[11]