Зборник радова Факултета техничких наука, Нови Сад

UDK: 007:004.056
DOI: https://doi.org/10.24867/16BE35Milovanovic

ОБЕЗБЕЂИВАЊЕ ИЗВОРНОГ КОДА КОЈИ ОБРАЂУЈЕ ПЛАТФОРМА ЗА АНАЛИЗУ КВАЛИТЕТА КОДА

SECURING SOURCE CODE PROCESSED BY A CODE QUALITY ANALYSIS PLATFORM

Милан Миловановић, *Факултет техничких наука, Нови Сад*

Област – СОФТВЕРСКО ИНЖЕЊЕРСТВО И ИНФОРМАЦИОНЕ ТЕХНОЛОГИЈЕ

Кратак садржај – *У раду је приказан модел претњи за платформу за анализу изворног кода. Такође је представљен дизајн за безбедно унапређење платформе који укључује обфускацију, употребу дигиталних сертификата и виртуелне приватне мреже.*

Кључне речи: *моделовање претњи, безбедност, поверљивост, обфускација*

Abstract – *The paper describes a threat model of a platform designed to analyze source code. The design for secure platform enhancement that includes obfuscation, the use of digital certificates, and a virtual private network was presented.*

Keywords: *threat modeling, security, confidentiality, obfuscation*

1. УВОД

У данашње време, изворни код као интелектуална својина представља најбитнију имовину многих организација. Доспеће пословних тајни у погрешне руке може изазвати огромну материјалну штету. Из тог разлога је заштита поверљивости корисничких података кључан аспект који треба размотрити при изради софтвера за анализу изворног кода. У овом раду ће, са аспекта безбедности, бити извршена анализа софтверског решења за анализу квалитета изворног кода – Clean CaDET платформе. Акценат је стављен на заштиту изворног кода који платформа обрађује.

Следеће поглавље ће бити посвећено безбедносној анализи дизајна и моделовању претњи на основу креираног дијаграма тока података. У трећем поглављу ће бити представљене митигације за претходно поменуте претње. Последње поглавље закључује рад и наводи предлоге за унапређење.

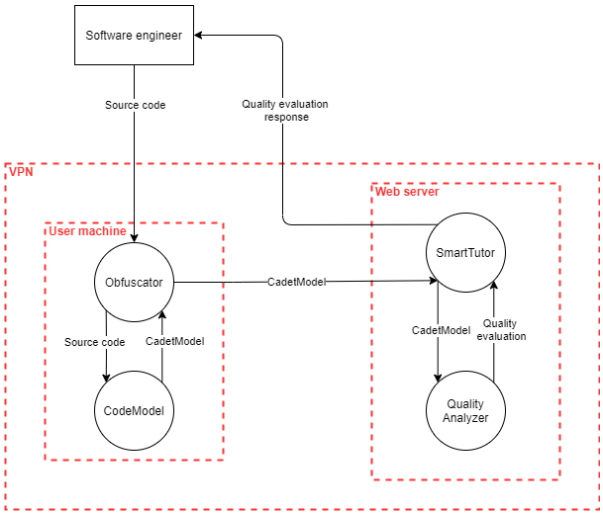
2. БЕЗБЕДНОСНА АНАЛИЗА ДИЗАЈНА

Примарна функционалност Clean CaDET платформе је детекција проблема унутар изворног кода (енгл. *code smells*) неког програма користећи моделе вештачке интелигенције. Затим, кориснику нуди персонализоване предлоге у облику образовног садржаја који ће му помоћи у решавању идентификованих проблема. Платформа има улогу дигиталног асистента софтверским инжењерима и интегрише се у њихова развојна окружења како би анализирала изворни код [1].

С обзиром да се у великом делу софтверских компанија изворни код сматра пословном тајном, а анализа се врши тако што се сам изворни код шаље са корисничких машина на Clean CaDET сервере, изворни код се може дефинисати кључним ресурсом у систему и потребно га је заштити од пада у погрешне руке.

Clean CaDET платформа ради са апстрактним моделом података који се креира парсирањем изворног кода – CaDETModel. У процесу парсирања игнорише се екстерни код (на пример, код из System простора имена) и креирају се објекти само од класа које су послате на Clean CaDET платформу [2]

Слика 1. приказује дијаграм тока података Clean CaDET платформе.



Слика 1 – *дијаграм тока података система у коме се CaDETModel парсира на корисничкој машини*

Парсирање изворног кода и креирање апстрактног модела се врши на корисничкој машини, затим се над SourceCode пољима елемената модела примењују трансформације обфускације. Обфускован модел се шаље на сервер и над њим се врши анализа квалитета.

2.1. Моделовање претњи

Shostack [3] моделовање претњи дефинише као употребу апстракција које помажу у размишљању о ризицима. Идеја која стоји иза моделовања претњи је

разумевање потенцијалних безбедносних претњи систему, утврђивање ризика и успостављање одговарајућих митигација. За идентификовање претњи користиће се метода под називом STRIDE.

У табели 1. налази се приказ идентификованих и класификованих претњи, као и делови система на које се поменуте претње односе. Уз сваку претњу је представљен и кратак опис претње.

Табела 1 - *Категоризација потенцијалних претњи*

ИД	Категорија	Мета	Опис
1.1	*Spoofing / Information disclosure*	Корисничка машина	Нападач може да се лажно представи као сервер, што омогућава да се изворни код са корисничке машине шаље директно нападачу
1.2	*Spoofing*	Сервер	Нападач може да се представи као обичнан корисник и оствари комуникацију са сервером.
1.3	*Information disclosure*	Комуникација са сервером	Нападач може да чита изворни код који се шаље на анализу ка серверу тако што прислушкује комуникацију између клијента и сервера преко мреже (*Man-in-the-middle* напад [4])
1.4	*Denial-of-service*	Сервер	Нападач може да оптерети сервере слањем великог броја захтева за анализу изворног кода
1.5	*Information disclosure*	Сервер	Инсајдер унутар Clean CaDET система може да дође у посед података користећи неки програм за читање радне меморије
1.6	*Repudiation*	Сервер	Инсајдер унутар Clean CaDET система може да порекне да је вршио малициозне радње над машинама

3.2 Решавање претњи

У табели 2. приказани су потенцијални начини на који ће се идентификоване претње решити.

Табела 2.- *Потенцијални начини за решавање претњи*

ИД	ИД Претње	Митигација
2.1	1.1	Аутентификација обе стране путем дигиталних сертификата приликом комуникације између клијента и сервера
	1.2	
2.2	1.3	Имплементација HTTPS (енгл. *Hypertext Transfer Protocol Secure*) протокола [5] у комуникацији између клијента и сервера
2.3	1.4	Остваривање везе између клијента и сервера путем виртуелне приватне мреже (енгл. *Virtual Private Network* – VPN [6])
2.4	1.5	Генерисање лог записа, анализа и управљање логовима
2.5	1.6	Имплементација механизама заштите лог датотека

3. ДИЗАЈН ЗА БЕЗБЕДНО УНАПРЕЂЕЊЕ ПЛАТФОРМЕ

У овом поглављу ће конкретније бити представљене митигације за претходно поменуте претње. Биће описан начин комуникације између клијентског и серверског дела платформе и кратак опис инфраструктуре јавних кључева. Затим ће бити приказане смернице за правилно конфигурисање виртуелне приватне мреже. На крају поглавља ће се анализирати појединачна поља апстрактног модела података са циљем да се за свако проблематично поље пронађу адекватне мере прикривања.

3.1. Безбедна комуникација између клијента и сервера

Сва комуникација између клијента и сервера је шифрована уз помоћ TLS (енгл. Transport Layer Security) протокола [7]. Ради повећања безбедности система, није довољно да клијентска апликација зна коме шаље изворни код, већ и да сервер зна од кога изворни код заправо пристиже. Зато се за комуникацију између клијената и сервера користи клијентска аутентификација [8].

3.2. Употреба инфраструктуре јавних кључева

Да би се осигурала безбедна комуникација између актера у Clean CaDET систему потребно је имплементирати инфраструктуру јавних кључева (у даљем тексту PKI).

3.2.1. Издавање сертификата

Свакој машини која користи услуге Clean CaDET система ће бити потребан приватни и јавни кључ. Уколико се приватни и јавни кључ генеришу на серверима и шаљу клијенту, увек ће постојати шанса да ће нападач на неки начин доћи у посед приватног кључа. Најбољи начин да се избегну проблеми са транспортом приватног кључа јесте да се он уопште

не транспортује преко мреже, већ да се генерише на корисничкој машини и никада не напушта систем.

За поступак пријаве и провере валидности података је задужено регистрационо тело (енгл. *registration authority* [9]) унутар PKI. Захтев за издавање сертификата се може послати путем форме на сајту осигураним TLS протоколом, а генерисани сертификат се кориснику може послати електронском поштом користећи S-MIME протокол [10]. Овај процес се такође може аутоматизовати тако што ће клијентска апликација генерисати парове кључева, послати захтев за издавање сертификата, а затим генерисани сертификат и приватни кључ сачувати у сигурном складишту на машини.

3.3. Употреба виртуелне приватне мреже

Део површине напада Clean CaDET система представљају IP адресе сервера. Како би се избегли *Denial-of-service* напади на Clean CaDET сервере, потребно је смањити површину напада скривањем IP адреса од нападача. Ово се може постићи употребом виртуелних приватних мрежа.

Да би виртуелна приватна мрежа донела предности у безбедности, потребно ју је правилно конфигурисати. Центар за националну сајбер безбедност Уједињеног Краљевства (енгл. *National Cyber Security Centre* – NCSC) [11] препоручује неколико смерница приликом конфигурације виртуелних приватних мрежа [12]:

- Користити IPSec (енгл. *Internet Protocol Security*) протокол [13] како би се добила флексибилност у избору између широког спектра интероперабилних производа;
- Користити аутентификацију базирану на сертификатима;
- Ако је могуће, користити изворни (енгл. *native*) клијент за своје платформе;
- Користити VPN који се аутоматски повезује, тако да корисници уређаја не морају ручно да га укључују;
- Избегавати *split tunneling* да би се смањио ризик од цурења података изван VPN-а [12].;
- Тестирати више VPN провајдера како би се открило који је најбољи за потребе организације и који је добољно робустан и отпоран на нападе.

3.4. Обфускација апстрактног модела података

Анализом апстрактног модела података се дошло до закључка да је могуће анонимизовати или уклонити нека поља у моделу, без утицаја на резултате анализе квалитета изворног кода. Такође, на основу комплексности и количине података које захтева, процес анализе је било могуће поделити на три засебна нивоа:

- Основна анализа,
- Анализа средње комплексности и
- Напредна анализа.

Могуће је извршити основну анализу квалитета изворног кода користећи само метрике. Ова анализа се састоји од примене простих правила на пристигле метрике. Количина информација која може процурити при анализи метрика је минимална, јер метрике не откривају ништа о семантици саме класе. Такође су игнорисане везе ка осталим класама, те је немогуће направити граф зависности класа. Анализа средње комплексности би захтевала поља на основу којих је могуће саставити граф зависности класа. Ова анализа би донела квалитетније резултате, међутим, и количина информација које би процуреле би била већа. Крајње, уколико се не поставе никаква ограничења на апстрактни модел података, над њим ће бити могуће извршити најнапредније анализе, али долази до потенцијалне ситуације где све информације падну у руке нападача, јер се шаље комплетан, непромењен изворни код класе или њених чланова. Најкорисније решење би било да корисник има опцију да изабере једну од три понуђене врсте анализе, у зависности од његових сигурносних потреба.

3.4.1. Анонимизација имена

Независно од нивоа анализе који је у питању, због повратних информација је на сервер потребно слати имена чланова апстрактног модела података. С обзиром да изворно име класе са собом носи потенцијал за цурење информација, потребно га је некако анонимизовати. Табела 3 наводи називе класа и поља која представљају имена чланова.

Табела 3 - *Поља која је потребно анонимизовати*

Класа	Поље
`CaDETClass`	`Name`
`CaDETClass`	`FullName`
`CaDETField`	`Name`
`CaDETMember`	`Name`

Над горе поменутим пољима ће бити извршене трансформације хеширања користећи алгоритам SHA256 (*Secure Hash Alghoritm 2*) [14]. Анонимизација се ослања на постојање посебне структуре података на клијенту – CaDETHashTable, која врши хеширање података и чува податке у облику **кључ: вредност**.

Процес анонимизације имена је следећи: приликом формирања апстрактног модела података, када се наиђе на поље поменуто у **Error! Reference source not found.**, израчунаће се SHA256 хеш вредност тог поља. Оригинална вредност поља се мења хеш вредношћу, а у CaDETHashTable се додаје нови елемент чији је кључ израчунати хеш, а вредност је оригинална вредност поља.

4. ЗАКЉУЧАК

У овом раду је извршена безбедносна анализа и креиран је модел претњи за софтверско решење за анализу квалитета изворног кода. Представљен је дизајн за безбедно унапређење Clean CaDET платформе, где је приказан процес издавања дигиталних сертификата у оквиру инфраструктуре

јавних кључева, представљен начин комуникације између клијентског и серверског дела система и наведене смернице за правилно конфигурисање виртуелне приватне мреже. На основу детаљне анализа апстрактног модела података су пронађена конкретна поља која је потребно анонимизовати.

Ради додатног повећања безбедности Clean CaDET платформе, даље истраживање би се могло усмерити у саму анализу безбедносних митигација, на пример, дубље истраживање проблема који настају приликом коришћења инфраструктуре јавних кључева у самом решењу.

5. ЛИТЕРАТУРА

[1] „Clean-CaDET: Who is it for?," [На мрежи]. Available: https://github.com/Clean-CaDET/platform#what-is-the-problem. [Последњи приступ 5 Октобар 2021].

[2] „Clean CaDET Code Model," [На мрежи]. Available: https://github.com/Clean-CaDET/platform/wiki/Module-Code-Model. [Последњи приступ 07 Август 2021].

[3] A. Shostack, Threat Modeling: Designing for Security, John Wiley & Sons, 2014.

[4] „What is MITM (Man in the Middle) Attack," [На мрежи]. Available: https://www.imperva.com/learn/application-security/man-in-the-middle-attack-mitm/. [Последњи приступ 29 Септембар 2021].

[5] „RFC 2818 – HTTP Over TLS," [На мрежи]. Available: https://datatracker.ietf.org/doc/html/rfc2818. [Последњи приступ 29 Септембар 2021].

[6] „What is a VPN," [На мрежи]. Available: https://www.kaspersky.com/resource-center/definitions/what-is-a-vpn. [Последњи приступ 1 Октобар 2021].

[7] „RFC 5246 - The Transport Layer Security (TLS) Protocol Version 1.2," [На мрежи]. Available: https://tools.ietf.org/html/rfc5246. [Последњи приступ 29 Септембар 2021].

[8] Cloudfare, „Introducing TLS with Client Authentication," [На мрежи]. Available: https://blog.cloudflare.com/introducing-tls-client-auth/. [Последњи приступ 29 Септембар 2021].

[9] „What is a Registration Authority," [На мрежи]. Available: https://www.primekey.com/wiki/what-is-a-registration-authority/. [Последњи приступ 30 Септембар 2020].

[10] „Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 4.0," [На мрежи]. Available: https://datatracker.ietf.org/doc/html/rfc8551. [Последњи приступ 30 Септембар 2021].

[11] „National Cyber Security Centre," [На мрежи]. Available: https://www.ncsc.gov.uk/. [Последњи приступ 1 Октобар 2021].

[12] „Virtual Private Networks (VPNs)," [На мрежи]. Available: https://www.ncsc.gov.uk/collection/device-security-guidance/infrastructure/virtual-private-networks. [Последњи приступ 1 Oktobar 2021].

[13] „RFC 2406 - IP Encapsulating Security Payload," [На мрежи]. Available: https://datatracker.ietf.org/doc/html/rfc2406. [Последњи приступ 1 Октобар 2021].

[14] W. Penard и T. v. Werkhoven, „On the Secure Hash Algorithm," [На мрежи]. Available: https://web.archive.org/web/20160330153520/http://www.staff.science.uu.nl/~werkh108/docs/study/Y5_07_08/infocry/project/Cryp08.pdf. [Последњи приступ 2 Октобар 2021].

[15] A. Chuvakin, K. Schmidt и C. Phillips, Logging and Log Management: The Authoritative Guide to Understanding the Concepts Surrounding Logging and Log Management, Syngress, 2013.

Кратка биографија:

Милан Миловановић рођен је 1. априла 1997. године у Смедереву. 2016 године постаје студент на Факултету техничких наука у оквиру Универзитета у Новом Саду, смер Софтверско инжењерство и информационе технологије. 2020. године је стекао диплому основних студија и уписао мастер студије на смеру Софтверско инжењерство и информационе технологије.