Зборник радова Факултета техничких наука, Нови Сад

UDK: 4.4
DOI: https://doi.org/10.24867/32BE16Tosic

ПРИМЕНА *IN-TOTO* ПЛАТФОРМЕ ЗА УНАПРЕЂЕЊЕ БЕЗБЕДНОСТИ СОФТВЕРСКИХ ЛАНАЦА ИСПОРУКЕ

APPLICATION OF THE *IN-TOTO* PLATFORM FOR ENHANCING SECURITY OF SOFTWARE SUPPLY CHAINS

Николина Тошић, *Факултет техничких наука, Нови Сад*

Област– ЕЛЕКТРОТЕХНИКА И РАЧУНАРСТВО

Кратак садржај – *Рад анализира безбедносне изазове у софтверским ланцима испоруке и испитује како in-toto платформа може да помогне у њиховом решавању. Пружа преглед функција које платформа нуди за праћење и обезбеђивање животног циклуса развоја софтвера, уз разматрање предности и ограничења.*

Кључне речи: *аутоматизација, интегритет софтверског ланца испоруке, верификација софтверских артефаката, безбедност софтвера*

Abstract – *The paper analyzes security challenges in software supply chains and examines how the in-toto platform can help address them. It provides an overview of the platform's feature for monitoring and securing the software development lifecycle, along with a consideration of its advantages and limitations.*

Keywords: *Automation, integrity of the software supply chain, verification of software artifacts, software security*

1. УВОД

Процес развоја софтвера је прилично сложен и укључује бројне кораке, алате и актере, и сходно томе јављају се и изазови у обезбеђивању интегритета и безбедности софтвера током свих фаза тог развоја и испоруке. Софтверски ланац испоруке (енгл. *Software supply chain*) обухвата све активности од писања кода, преко тестирања и интеграције, до крајње дистрибуције производа и свака од ових фаза ланца представља могућу тачку напада или манипулације која компромитује сигурност и квалитет софтвера [1].
Главна претња код софтверских система јесу управо напади на софтверски ланац. Један од скорашњих напада, *SolarWinds hack* [2], показује како нападачи могу убацивати злонамерни код у поуздане процесе, што доводи до огромних последица где је на хиљаде организација погођено. Напади попут замењивања зависности (енгл. *Dependency confusion*) [3] користе слабости у управљању пакетима тако што убацују злонамерне верзије библиотека, што додатно

НАПОМЕНА:
Овај рад проистекао је из мастер рада чији је ментор био др Горан Сладић, ред. проф.

Многе организације немају адекватне провере за безбедност компоненти које користе, ослањајући се на поверење уместо криптографске верификације.
In-toto платформа [4] кроз праћење и верификацију свих корака у софтверском ланцу испоруке пружа решење за ове изазове. Омогућава евидентирање свих корака који су предузети током развоја, тестирања и дистрибуције софтвера, пружајући могућност да се сваки артефакт и његова историја измена прате и верификују. Циљ *in-toto* је да осигура да сваки корак у процесу развоја буде извршен на предвиђен начин и да нема неовлашћених или злонамерних измена [5].
Отворени пројекти и компаније попут *Debian, Datadog* и *Gitlab* су имплементирали *in-toto* у своје *CI/CD* процесе како би осигурали интегритет свог софтвера. Ово указује на практичну примену овог решења и његов потенцијал за широку употребу у различитим софтверским екосистемима.

2. СОФТВЕРСКИ ЛАНАЦ ИСПОРУКЕ

Софтверске апликације састављене су од мреже компоненти и библиотека отвореног кода, где већина наслеђује функционалности из других извора. Овај ланац зависности омогућава брз развој софтвера, али такође излаже организације рањивостима које уводе промене изван њихове контроле. Софтверски ланац представља велики, растући, међусобно повезани систем технологија, људи и процеса, који представљају могућа места напада [1].
Како се спољне услуге (енгл. *outsourcing*) и повећана употреба комерцијалних готових производа шире, а крајњи корисници користе прилике да реконфигуришу или изврше додатне допуне на имплементираним производима и системима, тако сигурносни ризик у ланцу расте. Уобичајене грешке код софтвера лако могу бити искоришћене од стране неовлашћених лица како би променили сигурносна својства и функционалност софтвера у злонамерне сврхе. Такве грешке могу бити случајно или намерно убачене у софтвер у било којој фази његовог развоја или употребе, а крајњи корисници имају ограничене начине да пронађу и исправе те грешке како би избегли њихову злоупотребу.

3. СТАЊЕ У ОБЛАСТИ

Безбедност развоја софтвера [6] постаје све значајнија тема због све веће зависности од трећих страна (енгл.

third-party libraries) и отворених библиотека (енгл. *open-source libraries*). Како се софтверски екосистеми усложњавају и повезују, тако се повећава и број напада који искоришћавају слабе тачке овог ланца.

3.1 Значајни инциденти и претње

У *SolarWinds* нападу [2], нападачи су компромитовали један од *Orion* сервера и убацили код у један од модула ажурирања. Позадинско ажурирање које је било дигитално потписано, испоручено је за отприлике 18.000 *SolarWinds* купаца укључујући НАТО, *Microsoft, Intel* и бројне друге компаније као и владине организације. Овај код познат као *Sunburst*, био је део легитимног ажурирања које су корисници инсталирали без икаквих знакова компромитације. Напад је открио безбедносне слабости у управљању софтверских ланаца снабдевања јер је нападачима омогућено да се прикрију унутар легитимног софтвера и добију приступ осетљивим информацијама.

Замењивање зависности је још један начин напада на ланац развоја. У овим нападима актери покушавају да искористе уобичајену праксу коришћења трећих страна и менаџера пакета за аутоматско преузимање и инсталирање неопходних пакета за њихову примену [7]. Најчешће се на јавни репозиторијум постави малициозни пакет са истим називом као познати пакети. Пример овог напада јесте напад на *PyTorch,* где је нападач поставио зависност *torchtrition* на *PyPI* која је била названа исто као и у званичној *PyTorch* библиотеци. Како *PyPI* обично има предност при преузимању зависности у *Python* екосистему, злонамерни пакет је повучен током инсталације уместо *PyTorch* легитимног пакета.

3.2 Постојећа решења за безбедност ланца испоруке

Током година развијено је неколико решења за побољшање безбедности софтверског ланца. Једно од најранијих решења је коришћење криптографске верификације, као што је потписивање артефаката помоћу *GPG* (енгл. *GNU Privacy Guard*) кључева. Ови криптографски потписи омогућавају програмерима да провере интегритет и аутентичност софтвера пре него што га имплементирају. Иако ова метода обезбеђује основни ниво безбедности, управљање великим бројем кључева може постати компликовано, нарочито у великим системима [8].

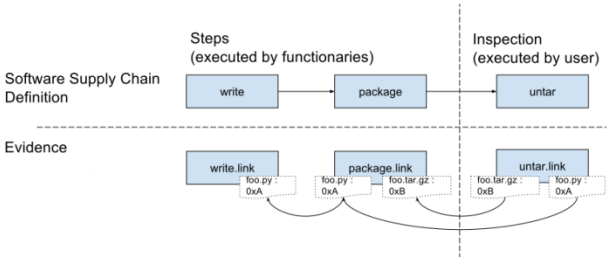
Други приступи укључују развој специфичних оквира као што је *The Update Framework* (*TUF*). *TUF* је дизајниран да обезбеди сигурност процеса ажурирања софтвера, штитећи га од напада. Његов главни циљ је обезбеђивање валидности софтверских артефаката који се преузимају и анализирају, да долазе из поузданих извора и да нису компромитовани. Ипак, *TUF* је фокусиран пре свега на процес ажурирања, док постоји потреба за решењима која могу покрити шири спектар развојног процеса.

Supply-chain Levels for Software Artifacts (SLSA) је још један оквир дизајниран да побољша безбедност софтверског ланца. Основна сврха *SLSA* је да обезбеди систематичан приступ за управљање ризицима који настају током развоја и дистрибуције софтвера, унапређујући сигурност на свим нивоима. Дефинише четири нивоа безбедности, од којих сваки укључује различите захтеве и мере.

За разлику од решења као што је *TUF,* које се фокусира на ажурирање, *in-toto* се фокусира на праћење и проверу интегритета сваког корака у животном циклусу софтвера. Кључне карактеристике *in-toto* платформе укључују праћење порекла, које омогућава верификацију сваког корака у процесу развоја, чиме се обезбеђује да свака промена буде проверена и аутентификована. *In-toto* такође користи криптографске метаподатке који омогућавају да се сваки артефакт који је креиран током развоја може верификовати и проверити његов интегритет.

4. IN-TOTO

У овом поглављу биће представљена *in-toto* платформа која спроводи интегритет софтверског развоја прикупљањем криптографских информација о самом процесу (слика 1).



Слика 1. In-Toto софтверски ланац

Да би се ово постигло, *in-toto* захтева да власник пројекта (енгл. *project owner*) декларише и потпише распоред (енгл. *layout*), ко и на који начин треба да спроводи кораке у процесу развоја. За корак који су извршили, актери снимају радњу и креирају криптографски потписану изјаву (енгл. *link metadata*). Снимљени метаподаци се могу верификовати како би се осигурало да су сви кораци изведени на одговарајући начин и од ауторизоване особе на начин дефинисан у *layout*-у. *Layout* као и метаподаци чврсто повезује улазе и излазе корака у ланцу, што осигурава да не дође до неовлашћеног приступа међу корацима. *Layout* дефинише и додатне захтеве како би се обезбедио квалитет крајњег производа [9].

4.1 Модел претњи

Модел претњи (енгл. *threat model*) односи се на безбедносне претње и ризике који су повезани са ланцем испоруке софтвера, а које *in-toto* покушава да ублажи. Главне претње које *in-toto* адресира су [9]:

- Неовлашћене измене – *in-toto* обезбеђује да сваки корак буде потписан од стране овлашћене особе;
- Инсајдерски напади – *in-toto* смањује ову претњу пружајући запис о томе ко је извршио сваки корак и које радње су предузете;
- Напади поновним коришћењем (енгл. *replay attacks*) – *in-toto* то спречава употребом јединствених метаподатака за сваки корак, осигуравајући да стари или већ коришћени кораци не буду поновљени;

- Компромитовање *build* система - *in-toto* бележи метаподатке на сваком кораку, осигуравајући да није дошло до манипулације;
- Нападачи могу компромитовати треће стране или спољне зависности – *in-toto* помаже да се осигура да су све спољне зависности верификоване и да су исправно проверене пре него што буду укључене у процес израде.

4.2 Безбедносни циљеви

Како би се изградио сигуран ланац испоруке софтвера потребно је испунити следеће циљеве:

- Интегритет ланца софтвера – сви кораци дефинисани у ланцу се изводе по наведеном редоследу, што значи да се ниједан корак не може додати, изменити или уклонити;
- Интегритет тока артефаката – сви артефакти коришћени у корацима не смеју се мењати између њих;
- Провера аутентичности корака – кораке могу изводити само особе које за то имају дозволу
- Постепено смањење безбедности (енгл. *graceful degradation*) – чак и ако су одређени кораци у ланцу угрожени, безбедност система није у потпуности нарушена.

4.3 *In-toto* улоге

Као и други безбедносни системи, *in-toto* користи сигурносне концепте попут делегација и улога како би смањио штету у случају да криптографски кључ буде компромитован. У контексту *in-toto* улога је скуп радњи и дужности које актер треба да изврши [9]. Постоје три улоге у оквиру *in-toto*:

- Власник пројекта – страна задужена за дефинисање ланца испоруке софтвера, дефинише које кораке треба извршити и ко их извршава;
- Функционери – стране које обављају кораке и обезбеђују записе о артефактима који се користе као материјали и њихови производи унутар ланца. Функционери су идентификовани помоћу јавних кључева које користе за потписивање;
- Корисник – људи или алати који користе финални производ, верификују производ.

4.4 *In-toto* компоненте

In-toto садржи три главне компоненте које раде заједно како би обезбедиле интегритет и сигурност софтверског ланца испоруке:

- Распоред – поставља структуру ланца испоруке који омогућава власницима пројекта да дефинишу захтеване кораке за све фазе;
- Линк метаподаци – представљају запис да су се кораци прописани у распореду десили. Материјали и поља метаподатака користе се како би се осигурало да ниједан међупроизвод није измењен у ланцу пре него што се употреби у кораку;
- Крајњи производ – *in-toto* осигурава да је производ прошао кроз проверене и сигурносне кораке без неовлашћених промена.

4.5 *In-toto* команде

In-toto пружа скуп команди које омогућавају управљање и верификацију токова безбедности софтверског ланца. Неке од главних команди су:

- *in-toto-record*
- *in-toto-verify*
- *in-toto-sign*
- *in-toto-verify-signature*

5. ИМПЛЕМЕНТАЦИЈА *IN-TOTO*

In-toto платформа имплементирана је и тестирана у оквиру *Python* апликације уз коришћење *Jenkins CI/CD* система, где је власник пројекта креирао репозиторијум и потребне датотеке, док су функционери имали задатке да измене, тестирају и пакују код. *In-toto* је инсталирана унутар *Docker* слике што омогућава креирање изолованог и конзистентног окружења. *Docker* омогућава лако управљање зависностима и гарантује да се *in-toto* алат покреће у предвидљивом окружењу. *Jenkins* се користи за аутоматизацију и праћење интегритета целокупног процеса развоја софтвера. *In-toto* пружа транспарентност и сигурност праћењем свих корака у ланцу, док *Jenkins* аутоматизује те кораке.
CryptoSigner класа из модула *securesystemslib.signer* [10] је класа која омогућава потписивање података коришћењем различитих криптографских алгоритама. Креира дигитални потпис за одређени сет података. *CryptoSigner* подржава различите криптографске алгоритме, као што су *RSA, ECDSA* и *Ed25519,* што омогућава флексибилност у избору механизма потписивања.

Layout из модула *in_toto.models.layout* [11] представља структуру која дефинише распоред корака у *in-toto* процесу. Кључни аспекти *Layout* класе су следећи:

- Структура кључева – *Layout* садржи информације о јавним кључевима који се користе за потписивање метаподатака;
- Дефинисање корака – *Layout* омогућава пројектантима да дефинишу низ корака који се морају пратити приликом развоја и испоруке софтвера. Сваки корак има одређене акције за које се очекује да буду извршене, као и кључеве особа којима је дозвољено да их извршавају;
- Провере – осим корака *Layout* може укључивати и инспекције које дефинишу додатне провере које треба обавити пре него што се производ испоручи клијенту.

5.1 Верификација

Верификација је кључни део *in-toto,* који осигурава да су сви кораци у ланцу изведени исправно и у складу са дефинисаним распоредом. Процес се састоји од неколико фаза:

1. Прикупљање метаподатака – када се заврши процес израде, метаподаци се прикупљају и чувају. Они укључују информације о свим корацима, материјалима, производима као и потписима који потврђују аутентичност тих корака;
2. Провера интегритета – *in-toto* користи криптографске хеш функције да би верификовао интегритет материјала и производа као што је претходно поменуто;
3. Верификација потписа – поред провере интегритета, *in-toto* такође проверава потписе повезане са метаподацима. Потписи су генерисани од стране овлашћених лица који су извршили одређене кораке у процесу;
4. Верификација распореда – *in-toto* проверава да ли су сви кораци у складу са распоредом, што укључује да ли су сви очекивани материјали и производи присутни. Уколико неки од корака недостају или нису правилно извршени, процес верификације ће пријавити грешку;
5. Извештавање – након завршетка процеса верификације, кориснику се пружа извештај који укључује информације о успешности верификације, као и све потенцијалне грешке или неусаглашености. Ово омогућава корисницима да брзо идентификују и реше проблеме.

6. ЗАКЉУЧАК

У раду је приказано решење за сигурност ланца испоруке софтвера, које пружа свеобухватну заштиту од широког спектра претњи које могу настати током развоја и испоруке софтверских производа.

Примарна предност *in-toto* у односу на друге постојеће алате за безбедност јесте у транспарентности коју пружа у процесу развоја софтвера, где на овај начин сви актери имају јасно дефинисане улоге. Може да се користи у различитим окружењима, чиме се повећава флексибилност и скалабилност алата. На практичном примеру је приказано да је безбедност софтверског ланца испоруке могућа, и да може аутоматски да се изврши уз коришћење *in-toto* платформе.

Рад на развоју стандардизованих интерфејса који омогућавају лако повезивање *in-toto* са другим безбедносним алатима и платформама може побољшати његову употребљивост и ефикасност. Укључивање вештачке интелигенције може помоћи у препознавању образаца и потенцијалних претњи, чиме би се побољшала брзина и тачност анализе безбедности. Увођење напреднијих метода потписивања и интеграције, као што су квантно-отпорни алгоритми може додатно осигурати отпорност система на будуће претње.

7. ЛИТЕРАТУРА

[1] Beatriz M. Reichert, Rafael R. Obelheiro, "*Software supply chain security: a systematic literature review*". International Journal of Computers and Applications 1-15, 2024

[2] Antigoni Kruti, Usman Butt, Rejwan Bin Sulaiman, "*A review of the SolarWinds Attack on Orion Platform using Persistent Threat Agent and Techniques for gaining Unauthorized Access*", arXiv preprint arXiv:2308.10294, 2023

[3] Riku Kestila, "*Acknowledging the risks of open source dependencies to software supply chain security*", (Master's thesis), 2022

[4] https://in-toto.io/ (приступ 5.11.2024.)

[5] Santiago Torres-Arias, Thrishank Karthik Kuppusamy, Reza Curtmola, Justin Cappos, "*in-toto: Providing farm-to-table guarantees for bits and bytes*", In *28th USENIX Security Symposium (USENIX Security 19)* (pp. 1393-1410), 2019

[6] Michael Lieberman, Brandon Lum, "*Securing the Software Supply Chain*", Maning, 2023

[7] Leah Roberts, "*Countermeasures for preventing malicious infiltration on the information technology supply chain*", (Doctoral dissertation, Purdue University) 2023

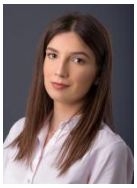
[8] Phong Q. Nguyen, "*Can We Trust Cryptographic Software? Cryptographic Flaws in GNU Privacy Guard v1.2.3*", In *Advances in Cryptology-EUROCRYPT 2004: International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, May 2-6, 2004. Proceedings 23* (pp. 555-570). Springer Berlin Heidelberg, 2004

[9] Santiago Torres-Arias, "*in-toto: Practical Software Supply Chain Security*", New York University Tandon School of Engineering, 2020

[10] https://github.com/secure-systems-lab/securesystemslib (приступ 5.11.2024.)

[11] https://in-toto.readthedocs.io/en/latest/layout-creation-example.html (приступ 5.11.2024.)

Кратка биографија:



Николина Тошић рођена је 1999. године у Новом Саду. Мастер рад на Факултету техничких наука из области Електротехнике и рачунарства – електронско пословање одбранила је 2024.године.

Контакт: nikolinatosic999@gmail.com