Зборник радова Факултета техничких наука, Нови Сад

UDK: 007:004.056
DOI: https://doi.org/10.24867/34BE19Popovic

Имплементација мултифакторске аутентификације за Open SSH сервер

Implementation of Multifactor Authentication for the Open SSH Server

Петар Поповић, *Факултет техничких наука, Нови Сад*

Студијски програм– ЕЛЕКТРОТЕХНИКА И РАЧУНАРСТВО

Кратак садржај – *Рад реализује имплементацију мултифакторске аутентикације за OpenSSH сервер, уз коришћење TOTP кодова као другог фактора. Приказани су кораци конфигурације, тестирања и анализа безбедносних аспеката током пријаве корисника на сервер.*

Кључне речи: *мултифакторска аутентификација, Open SSH сервер, MS аутентификатор, TOTP код*

Abstract – *In this paper, the implementation of multi-factor authentication for the OpenSSH server is presented, using TOTP codes as the second authentication factor. The configuration steps and security analysis during user login to the server are described.*

Keywords: *Multifactor authentication, Open SSH server, MS authenticator, TOTP code*

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

1. УВОД

Дигитална инфраструктура постала је критична за готово све сфере савременог друштва, од комуникација и образовања до финансијских трансакција и управљања индустријским процесима [1]. Са растом комплексности система и међусобне повезаности сервиса, површина напада се континуирано шири, а последице инцидената све су скупље и видљивије [2]. Зато превенција, рано откривање рањивости и увођење вишеслојних механизама заштите представљају темељ одрживе сајбер безбедности. Традиционална аутентификација подразумева аутентификацију лозинкама. Она се у данашњем времену показала недовољном услед поновне појаве *phishing*-а , крађе база и *Brute-force* напада. Чак и уз политичке сложености и ротације лозинки, компромитација једног фактора често доводи до неовлашћеног приступа подацима [3,4]. Додавање додатног другог фактора значајно умањује ризик од преузимања корисничког налога. Мултифакторска аутентификација заснована на временски ограниченим једнократним кодовима *TOTP* (енг. *Time-based One-Time Password)* уравнотежује безбедност и употребљивост [5].
Код се генерише локално на уређају корисника, важи кратко и безбедно се упоређује на серверу [6]. Овај модел не захтева сталну повезаност ка спољним сервисима и добро се уклапа у оперативне услове серверских окружења [7].
Циљ рада је да се пројектује, имплементира и оцени прилагођени модул за *TOTP*, те интегрише у *OpenSSH* сервер тако да се обезбеди ток „Лозинка + *TOTP*“ без нарушавања постојећих оперативних навика администратора. Интеграцијом *TOTP* у *SSH* (енг. Secure shell) очекује се смањена вероватноћа успешног неовлашћеног приступа у сценаријима компромитације лозинке.

2. *OPENSSH* СЕРВЕР И АУТЕНТИФИКАЦИЈА

*SSH (*енг. *Secure Shell)* представља криптографски мрежни протокол намењен безбедној комуникацији између клијента и удаљеног сервера. Развијен је као замена за небезбедне протоколе као што су *Telnet* и *rlogin*, који су слали податке у отвореном тексту и били изложени пресретању и злоупотреби [7]. *SSH* обезбеђује три кључна својства комуникације: поверљивост, интегритет и аутентичност учесника у комуникацији [8]. Архитектура OpenSSH се временом развијала да укључује модуларни *PAM* слој, подршку за аутентификацију базирану на сертификатима, доношење одлуке о приступу на основу више фактора и могућност интеграције са системима за централизовану контролу идентитета. Ова еволуција резултирала је тиме да данашњи *OpenSSH* омогућава сегментацију корисничких сесија, анализу логова и динамичко управљање приступом.

2.1 Архитектура *OpenSSH* и слојеви безбедности

SSH протокол се састоји од три логичка слоја. Први слој је транспортни који обезбеђује енкрипцију и интегритет саме комуникације. Други слој протокола је слој аутентификације и он омогућава серверу да потврди идентитет клијента који успоставља везу са сервером. Последњи је слој канала и он служи за управљање више логичких сесија унутар једне SSH конекције. Током успостављања везе, клијент и сервер размењују јавне кључеве и договарају се о симетричним сесијским кључевима који ће бити коришћени за енкрипцију података [9]. Додатни PAM (енг. Pluggable Authentication Modules) слој, представља флексибилан и проширив механизам за управљање процесом аутентификације. Основна предност овог слоја је могућност да раздвоји апликациону логику од механизама за проверу идентитета [10].

3. *MFA* И ИМПЛЕМЕНТАЦИЈА У OPENSSH

Мултифакторска аутентификација (*MFA*) подразумева употребу два или више независних фактора приликом провере идентитета корисника. Основна идеја је да компромитација једног фактора не доведе аутоматски до компромитације налога корисника. Фактори додатне провере се обично сврставају у три категорије. Прва категорија су нешто што корисник зна, као лозинка, безбедносно питање или пин код. Други могући фактор јесте нешто што корисник поседује као мобилни уређај, паметни токен, одвојени дигитални кључ или нешто друго. Као последњи могући фактор додатне провере је нешто што сам корисник јесте. Ту спадају сви биометријски подаци корисника, отисак прста, лице или зеница ока.
Оваква комбинација значајно смањује ризик од неовлашћеног приступа и усклађује систем са безбедносним стандардима.

3.1 Интеграција *MFA* у *OpenSSH*

OpenSSH омогућава интеграцију *MFA* преко *PAM* слоја који се налази између механизма за пријаву и система за проверу идентитета. Конкретна имплементација може се постићи додавањем *TOTP* модула у конфигурацију [11]. Цео процес пријаве корисника на систем започиње уносом корисничког имена и лозинке у конзолу. Уколико је први фактор аутентификације био успешан, сервер од корисника тражи унос *TOTP* кода. У следећем кораку након уноса кода, код се верификује путем библиотеке која генерише исти алгоритам. Тек успешна верификација овог кода омогућава кориснику потпуни приступ систему.
Овај приступ не захтева екстерне сервисе и задржава све предности *OpenSSH* укључујући безбедност, отворени код и једноставну конфигурацију.

4. СПЕЦИФИКАЦИЈА СИСТЕМА

Ово поглавље описује захтеве, архитектуру и дизајн система за имплементацију мултифакторске аутентификације у OpenSSH серверу. Систем је развијен са циљем да обезбеди додатни ниво безбедности у SSH окружењу применом другог фактора аутентификације заснованог на *TOTP* алгоритму.

4.1 Архитектура система

У овом поглављу приказана је архитектура и основна логика решења за интеграцију мултифакторске аутентификације у SSH окружењу.
Систем се састоји од три главне компоненте:

- Клијентска апликација - представља интерфејс преко ког корисник иницира конекцију са сервером;
- Серверска инфраструктура - прихвата захтев, управља фазама пријаве и преусмерава процес ка *PAM* систему;
- Безбедносни слој - имплементира додатни фактор аутентификације и валидацију једнократних кодова;

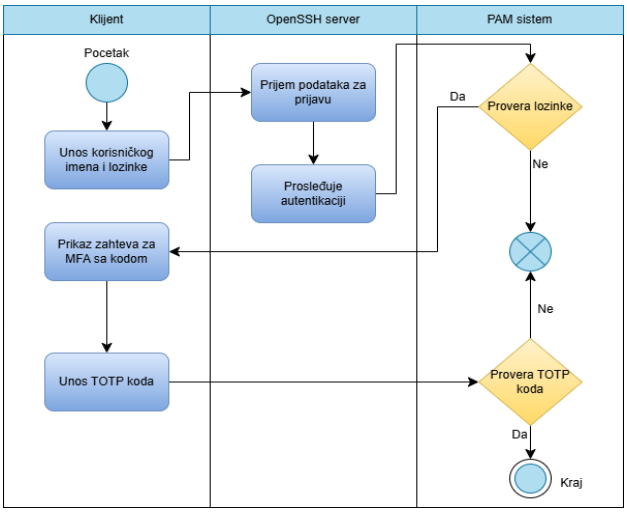
Клијент систему приступа преко терминала који шаље корисничко име и лозинку серверу. Сервер, раније конфигурисан да користи *PAM* систем као посредни слој, прослеђује податке *PAM*-у. Даље се иницира мултифакторска аутентификација и позива се *TOTP* модул ради верификације временски ограниченог кода генерисаног на уређају.
PAM механизам је од централног значаја у овој архитектури јер омогућава додавање нових метода аутентификације без измене кода *OpenSSH* сервера. На овај начин обезбеђује се проширивост и одрживост система у различитим инфраструктурама. Овај модел значајно унапређује безбедност јер минимизује ризик од компромитације налога и онемогућава приступ са само једним фактором аутентификације [11].

4.2 Процес аутентификације

Процес корисничке аутентификације у OpenSSH систему са интегрисаним мултифакторским механизмом може се описати кроз следеће кораке:

- Иницијализација конекције - клијент покреће SSH сесију и шаље серверу своје корисничко име.
- Унос лозинке - корисник уноси лозинку која се прослеђује серверу ради провере.
- Валидација лозинке - сервер позива *PAM* механизам да провери лозинку.
- Генерисање и унос *TOTP* кода - корисник користи апликацију која генерише код на основу тајног кључа и времена.
- Провера кода преко *TOTP* модула - *PAM* верификује унети код.
- Доношење одлуке - ако су оба фактора исправна, приступ је дозвољен; у супротном, сесија се прекида.

Овако дефинисан процес повећава безбедност и обезбеђује виши степен поузданости и контроле приступа у мрежним окружењима (слика 1).



Слика 1. *Дијаграм активности*

5. ИМПЛЕМЕНТАЦИЈА СИСТЕМА

За интеграцију мултифакторске аутентификације у *OpenSSH* сервер имплементиран је нови *TOTP* модул. Његова функција је да прихвати, обрађује и исправно

одговори на захтев клијента за аутентификацију. Представља проширење постојећег *PAM* модула.

5.1 Структура и интерфејси модула

У оквиру имплементације мултифакторске аутентификације, развијен је посебан *PAM* модул који омогућава интеграцију *TOTP* другог фактора у OpenSSH аутентификациони процес. Овај модул је кључни слој који омогућава проверу корисничког *TOTP* кода, који се генерише синхронизовано у односу на тајни кључ корисника и системско време.
Основни PAM модул развијен је у језику C и ослања се на стандардне интерфејсе. Најважнија тачка комуникације са стеком јесте функција за аутентификацију у којој почиње ток *MFA*.
Развој модула био је вођен циљем минималне интервенције у постојећи *OpenSSH* код тако што се користе механизми који су део *PAM* архитектуре и њених интерфејса, а која постоји као слој за аутентификацију у Linux систему. *PAM* модул неприметно интегрише нову *MFA* логику, а корисници настављају да користе већ познате клијентске алате као што је мобилна апликација за генерисање *TOTP* кодова.

5.2 Конфигурација система

За интеграцију развијеног *PAM* модула у системску аутентификацију *OpenSSH* сервера потребно је извршити суштинске измене у два конфигурациона фајла. Један је конфигурациони фајл *OpenSSH* а други *PAM*.
У конфигурационом фајлу *OpenSSH*, за активирање мултифакторске аутентификације, мора да се омогући *ChallengeResponseAuthentication*, што омогућава интеракцију корисника путем *keyboard-interactive* механизма. Поред тога, *AuthenticationMethods* се конфигурише тако да захтева почетни унос лозинке и потом *TOTP* кроз *keyboard-interactive*. Ово осигурава да корисник мора успешно проћи оба корака да би се пријавио. Ове измене представљају основни предуслов за успешан рад тока мултифакторске аутентификације засноване на привременим токенима у *OpenSSH* окружењу. Измењен конфигурациони фајл је приказан на листингу 1.

```
Port 2222
UsePAM yes
KbdInteractiveAuthentication yes
PasswordAuthentication no
AuthenticationMethods keyboard-interactive
```

Листинг 1. Конфигурациони *OpenSSH* фајл

5.3 Функције аутентификације *TOTP* модула

Главна функција сваког PAM аутентификационог модула је функција за обраду аутентификације. Она интегрише све алгоритамске и безбедносне кораке који воде од покретања сесије, преко прикупљања и провере података корисника, до коначне одлуке о дозвољавању или ускраћивању приступа систему.
На самом почетку функција преузима параметре и корисничке податке преко PAM модула. Користећи помоћну функцију, она идентификује корисника који покушава пријаву. Потом, приступа се фајлу који садржи тајни бинарни кључ у главном директоријуму тог корисника.
Функција за декодирање служи да тај кључ безбедно претвори у облик погодан за криптографску обраду. После успешног декодирања кључа, ток се наставља ка делу за интеракцију са самим корисником. Ту се шаље прилагођени текст за испис преко prompt_code и чека се унос кода од стране корисника. Овде систем подржава напредно конфигурисање, где се преко листе аргумената може мењати име фајла, број дозвољених покушаја, временски корак, број потребних цифара и стил корисничког prompt-a.

5.4 Генерисање и валидација токена

У модулу је додатно имплементирана функција за генерисање кода (*totp*). Користи се за генерисање и валидацију токена на серверској страни. У развијеном модулу користи *HMAC-SHA1* алгоритам.
Као улаз добија тајни кључ, временско стање, временски корак и број цифара за излаз. Прво се израчунава вредност броја (*counter*) који је целобројни део тренутног времена подељеног временским интервалом.
Овај број се представља као низ од 8 бајтова и шаље заједно са кључем у *HMAC* функцију. Добијени *MAC* (*Message Authentication Code*) се затим динамички сече, а добијена вредност се модуларно скраћује на жељени број цифара — најчешће шест. Функција је приказана на слици 2.

```
static unsigned int totp(const unsigned
char *key, size_t keylen,time_t t,
int step,int digits) {
       long long counter = (long long)
              (t / step);
       unsigned char msg[8];
       for (int i = 7; i >= 0; i--) {
         msg[i] = (unsigned char)
         (counter & 0xFF);
         counter >>= 8;
       }
       unsigned char mac[EVP_MAX_MD_SIZE];
       unsigned int maclen = 0;
       HMAC(EVP_sha1(), key, (int)keylen,
       msg, sizeof(msg), mac, &maclen);
       int off = mac[maclen - 1] & 0x0F;
       unsigned int bin =
         ((mac[off] & 0x7Fu) << 24) |
         ((mac[off + 1] & 0xFFu) << 16) |
         ((mac[off + 2] & 0xFFu) << 8) |
         (mac[off + 3] & 0xFFu);
       unsigned int mod = 1;
       for (int i = 0; i < digits; i++)
         mod *= 10u;
       return bin % mod;
}
```

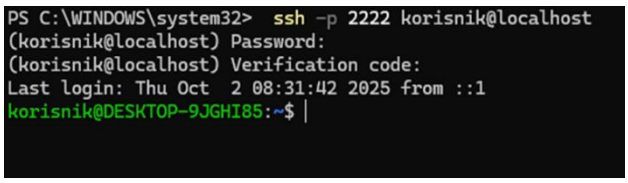
Слика 2. *Функција за генерисање* TOTP *кода*

При валидацији списак важећих кодова генерише се за неколико временских интервала чиме се умањује утицај малих десинхронизација корисника и сервера.

Ако унети код одговара неком референтном коду у датом прозору, *MFA* провера је успешно завршена.

5.5 Пријава на систем

Током процеса мултифакторске аутентификације неопходно је кориснику у сваком тренутку обезбедити јасне и прецизне поруке. *PAM* модул је по том питању потпуно модуларан и омогућава једноставно дефинисање свих исписа на терминалу. Како би се избегла непотребна сложеност, кориснику се приказују само основне поруке за унос лозинке и за унос верификационог кода. У случају исправног уноса оба параметра, систем приказује поруку о успешној пријави, док се у супротном исписује обавештење о неуспешној аутентификацији. Интеракција корисника са *OpenSSH* сервером након увођења мултифакторске аутентификације остаје једноставна и интуитивна, уз минималне измене у односу на класичан ток пријаве. Главна разлика је у додатном кораку провере идентитета. Након иницијализације *SSH* конекције, корисник уноси корисничко име и лозинку као и у стандардној сесији. Уколико је лозинка исправна, сервер активира *PAM* механизам који прослеђује процес ка *TOTP* модулу. Тада се кориснику приказује упит за унос једнократног кода који се генерише на његовом мобилном уређају у апликацији за аутентификацију. Ако је код валидан, приступ систему се одобрава и корисник је успешно улогован. У супротном, пријава се прекида уз одговарајућу поруку о грешци. Цео ток процеса од иницијализације конекције до успешне верификације једнократног кода приказан је на слици 3.



Слика 3. *Терминал приликом успешне пријаве*

6. ЗАКЉУЧАК

У оквиру овог рада изложена је имплементација мултифакторске аутентификације за OpenSSH сервере, са посебним освртом на интеграцију Microsoft Authenticator апликације као другог фактора заснованог на ТОТП механизму. Представљено решење омогућава значајно повећање безбедности приступа серверу, што је потврђено кроз серију тестирања у локалном SSH окружењу. Резултати показују да је увођење мултифакторске аутентификације практично и ефективно - додатни слој аутентикације не захтева комплексну инфраструктуру, кориснички ток није значајно успорен, а процес пријаве остаје интуитиван и лак за све кориснике.

Примена ТОТП токена и подршка кроз Microsoft Authenticator доказале су се као једноставне за интеграцију и употребу, уз минималне захтеве за додатну конфигурацију и обуку. Узимајући у обзир све изведене тестове, имплементација мултифакторске аутентификације базиране на ТОТП стандардима потврдила је своју поузданост и скалабилност. Овакав модел је посебно прикладан за окружења где је потребно брзо обезбедити критичне ресурсе, као и за све сценарије где се жели подизање отпорности на напредне претње и злоупотребе корисничких акредитација.

Имплементација оваквог решења представља препоручени корак за све организације и појединце који желе да подигну ниво заштите SSH сервера.

7. ЛИТЕРАТУРА

[1] Stallings, W. (2017). Network Security Essentials: Applications and Standards. 6th ed., Pearson.

[2] Schneier, B. (2015). *Data and Goliath: The Hidden Battles to Collect Your Data and Control Your World.* W.W. Norton & Company.

[3] Das, A., Bonneau, J., Caesar, M., Borisov, N., & Wang, X. (2014). *The Tangled Web of Password Reuse.* NDSS Symposium.

[4] Florêncio, D., & Herley, C. (2011). *Where Do Security Policies Come From?* Proceedings of the Sixth Symposium on Usable Privacy and Security (SOUPS).

[5] O'Gorman, L. (2003). Comparing Passwords, Tokens, and Biometrics for User Authentication. Proceedings of the IEEE, 91(12), 2021–2040.

[6] M'Raihi, D., Machani, S., Pei, M., & Rydell, J. (2011). TOTP: Time-Based One-Time Password Algorithm. RFC 6238, IETF.

[7] Ylönen, T., & Lonvick, C. (2006). The Secure Shell (SSH) Protocol Architecture. RFC 4251, IETF.

[8] Kim, H. & Smith, M. (2019). "Security comparison of Secure Shell (SSH) and predecessor protocols." Journal of Communication and Computer, 16(6), 262–270

[9] Barret, D., Silverman, R., & Byrnes, R. (2012). SSH, The Secure Shell: The Definitive Guide. O'Reilly.

[10] Samar, V., & Lai, C. (1996). Pluggable Authentication Modules. Sun Microsystems White Paper.

[11] M'Raihi, D., et al. (2011). TOTP: Time-Based One-Time Password Algorithm. RFC 6238.

Кратка биографија:

Петар Поповић рођен је 2001. године у Лозници. Основне академске студије је завршио 2023. године на Факултету техничких наука у Новом Саду. Мастер рад на Факултету техничких наука из области Рачунарство и аутоматика – Електронско пословање одбранио је 2025. године.