Зборник радова Факултета техничких наука, Нови Сад

UDK: 4.056
DOI: https://doi.org/10.24867/32OI03Panic

НЕИНВАЗИВНА АНАЛИЗА БЕЗБЕДНОСТИ ВЕБ АПЛИКАЦИЈА ДРЖАВНИХ ОРГАНА РЕПУБЛИКЕ СРБИЈЕ

NON-INVASIVE SECURITY ANALYSIS OF GOVERNMENT WEB APPLICATIONS IN THE REPUBLIC OF SERBIA

Душан Панић, *Факултет техничких наука, Нови Сад*

Област – ИНФОРМАЦИОНА БЕЗБЕДНОСТ

Кратак садржај – *Овај рад истражује неинванзивну анализу веб апликација у Републици Србији користећи методу веб фаза. Уз инспирацију из војних техника извиђања, развијен је прилагођени веб фазер у Go програмском језику који омогућава ефикасну аутоматизацију идентификације рањивости. Систем укључује компоненте као што су координатор послова, DNS и VPN сервери, који омогућавају скалабилност и координацију задатака. Аутоматска обрада резултата користи статистичке методе за филтрирање и анализу података,са фокусом на анализу фреквенције и екстремних вредности у прикупљеним подацима..*

Кључне речи: *Информациона безбедност, извиђање, тестирање информационе безбедности*

Abstract – *This paper explores a non-invasive security analysis of web applications in Republic of Serbia using the web fuzzing method. A customized web fuzzer was developed in the Go programming language which was inspired by military reconnaissance techniques, and enabling efficient automation of vulnerability identification. The system consists of a Job Scheduler as well as DNS and VPN servers, which allow scalability and task coordination. Automatic processing of results utilizes statistical methods for filtering and analysing data, focusing on the analysis of frequency and extreme values in the collected data.*

Keywords: *Information Security, Reconnaissance, Information Security Testing*

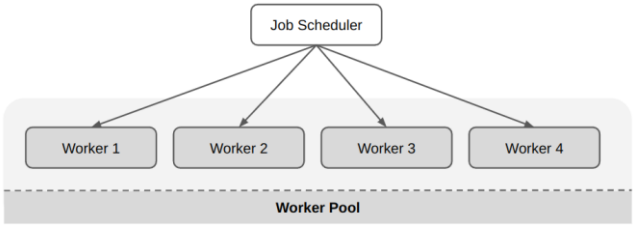
1. УВОД

Информациона безбедност се односи на заштиту информација и информационих система од неовлашћеног приступа, употребе, откривања, ометања, модификације или уништавања. Примарни фокус је ***балансирана*** заштита поверљивости, интегритета и доступности познатија као ***CIA*** тријада [2].
Постизање **информационе безбедности** се постиже ублажавањем ризика [1]. Информациона безбедност обухвата разне технике, полисе и процедуре које се користе за заштиту информација и ресурса од различитих врста претњи, укључујући крађу података, неовлашћен приступ заштићеним рачунарима и мрежама, злонамерне софтвере, социјални инжењеринг и друге. Приступ информационој безбедности често укључује комбинацију техничких, организационих и процедуралних мера за заштиту информација и система.

Циљеви истраживања укључују идентификацију и анализу кључних рањивости веб апликација, аутоматизовано прикупљање информација о безбедносном стању, евалуацију постојећих заштитних механизама, и на основу добијених резултата, предлагање мера за побољшање националне сајбер безбедности. Истраживање се ослања на развој дистрибуираног веб фазера у облаку, израђеног у *Go* програмском језику. Метод је имплементиран кроз активно извиђање, примењујући веб фаз технику која подразумева слање великог броја неочекиваних или неважећих улазних података на циљане веб апликације

2. ДИЗАЈН СИСТЕМА

Кључне компоненте система укључују *координатора*, *раднике*, *Docker* [3] контејнере, *прокси* [4] сервере и *VPN* [5] клијенте.



Слика 1. *Визуелизација нодова у мрежи*

На слици 1 приказане су компоненте система:

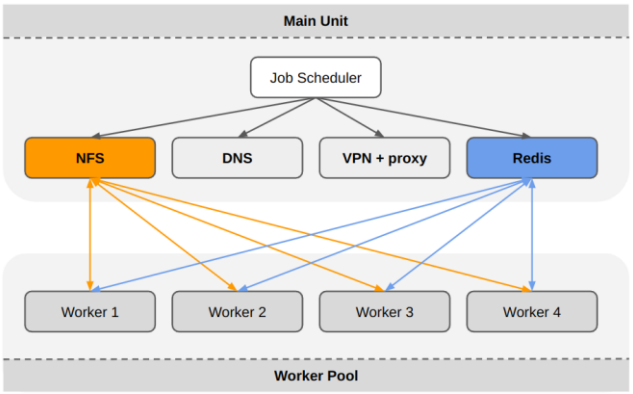
- ***Координатор*** срце система, задужено за оркестрацију и расподелу задатака према радницима.
- ***Радници*** **-** служе као извршиоци задатака који су им додељени од стране *координатора*.
- ***Docker контејнери*** **-** користе се за изолацију и покретање специфичних сервиса, укључујући прокси сервере и *PiHole DNS* сервер.
- ***Прокси сервер*** **-** сви *HTTP* захтеви се прослеђују на жељену веб адресу кроз *VPN*

НАПОМЕНА:
Овај рад проистекао је из мастер рада чији ментор је био др Имре Лендак, ванр. проф.

конекцију, што доприноси смањењу ризика од блокирања.

Координатор представља кључни чвор система, играјући централну улогу у оркестрацији и дистрибуцији задатака према радницима. Инсталирани су и подешени следећи елементи: **Redis сервер** [6] се користи као централни систем за управљање редовима задатака, омогућавајући синхронизацију и ефикасну дистрибуцију послова између радника. Redis служи и као глобални механизам за закључавање, осигуравајући да се исти задатак не извршава симултано на више машина. ***NFS* сервер** [7] служи за дељење фајлова између машина у мрежи, омогућавајући им приступ заједничким ресурсима потребним за извршавање задатака. ***Docker*** на *координатор* виртуелној машини омогућава контејнеризацију апликација и сервиса, укључујући *прокси* сервере и *VPN* клијенте.

Радници су неопходне компоненте дистрибуираног система, са основним извршења задатака додељених од стране *координатора*. Њихова примарна функција је обрада и анализа. Кључна карактеристике и улоге радника је **извршавање задатака,** тиме што активно раде на обради и анализи података примљених од *координатора*. Оне су способне да паралелно процесуирају више задатака, чиме се оптимизује време потребно за комплетан процес анализе.



Слика 2. *Визуелизација комуникације између нодова у мрежи*

Радници остварују комуникацију са *координатором* коришћењем Redis сервера што је приказано на слици 2. Информације о новим задацима се смештају у Redis базу од стране *координатора*, омогућавајући радницима да преузму и реализују задатке. ***NFS* клијент** је конфигурисан на сваком раднику омогућавајући јој приступ дељеним фајловима који се налазе на *NFS* серверу хостованом на *координаторовој* виртуелној машини. Ово омогућава способност система да одржи конзистентност података кроз више компоненти и да усклади операције између различитих машина. ***Веб фазер*** је покренут на радницима, а написан је у Go програмском језику. Фазер служи за идентификацију потенцијално рањивих путања унутар веб апликација. Фазер представља *кључни алат у процесу анализе и тестирања*, пружајући детаљне информације о безбедносном стању анализираних система. Оптимизација перформанси добијена је употребом **HTTP "Keep Alive"** [8] механизма за поновно коришћење TCP/HTTP конекција, чиме се значајно смањује кашњење при скенирању, јер не постоје поновна TCP односно TLS [9] руковања (енг. handshake). Користећи Go рутине за ефикасно паралелно процесирање задатака, овај приступ осигурава високу пропусност уз минималну потрошњу ресурса. Фазер је дизајниран да буде скалабилан и лак за интеграцију у дистрибуиране системе.

Docker контејнери су коришћени за изолацију и покретање *прокси* и *PiHole* DNS сервера. Контејнеризација је омогућила ефикасно управљање овим сервисима, пружајући изоловану средину за сваку компоненту која олакшава деплојмент, скалабилност и управљање верзијама. ***Прокси сервер*** имплементиран је сервер *HTTP посредника за прослеђивање* који је постављен унутар *Docker* контејнера. Овај приступ је омогућио сигурно и ефикасно усмеравање *HTTPS* [10] захтева преко *VPN*-а. Коришћен је *NordVPN* сервис. Употреба *Dockera* омогућила је изолацију прокси сервера од осталих делова система. ***PiHole DNS сервер*** [11] такође постављен унутар *Docker* контејнера. *PiHole* служи као DNS кеш сервер, оптимизујући DNS упите и побољшавајући перформансе система тако то смањује потребу за поновљеним екстерним DNS захтевима.

Прокси сервер функционише као посредник између система и циљаних веб апликација, усмеравајући све *HTTPS* захтеве преко *VPN* тунела. Коришћењем *NordVPN*-а, ова конфигурација значајно доприноси сигурнијој и анонимнијој конекцији, умањујући ризик од блокирања захтева фазера од стране циљаних сервера. Улога и значај *прокси* сервера су **анонимизација захтева** употребом *VPN-а* у комбинацији са прокси сервером, захтеви постају теже препознатљиви за циљане веб апликације. Ово је посебно важно у сценаријима где је потребно извршити велики број захтева без ризика од блокирања по IP адреси. **Избегавање блокирања** коришћењем динамичке промене IP адреса преко *VPN-а*, смањује се вероватноћа да ће захтеви бити блокирани од стране циљаних сервера, чиме се повећава успешност *фаз* процеса.

Оптимизација *HTTP* конекција

- **Перзистентне конекције**: употребом перзистентних *HTTP* конекција значајно је смањен оверхеад повезан са поновним успостављањем *TCP* веза. Ово директно доприноси бржем одзиву система, омогућавајући бржу обраду и прослеђивање захтева кроз *прокси* сервер.
- **Повећани лимити конекција**: постављањем високих лимита за максималан број истовремених конекција (до 500 конекција по хосту), обезбеђујемо да систем може ефикасно руковати великим бројем захтева без ризика од преоптерећења или загушења мреже.
- **Конфигурација *TLS***: кроз прилагођавање ***TLS*** конфигурације са ***InsecureSkipVerify***: true, олакшавамо процес тестирања сервера са

самопотписаним сертификатима или онима који нису издати од стране поузданог ауторитета.

Да би се подржао велики број истовремених конекција и захтева, машине су конфигурисане са прилагођеним системским лимитима за број отворених фајлова (***nofile***) и процеса (***nproc***), као и са оптимизованим кернел параметрима (***sysctl.conf***) за боље перформансе мреже и системских ресурса

3. ПРОВЕРА ВАЛИДНОСТИ ДОМЕНА ЗА ТЕСТ

За потребе тестирања одабран је РНИДС-ов документ **"Списак домена резервисаних за потребе државних органа и организација"** коришћен као почетна база за израду тест скупа Преузет је са **РНИДС-а.** [12] Документ је у тренутку писања рада важећи, а публикован је 24. 10. 2016. Обрадом фајла, елиминацијом дупликата дошли смо до цифре од **1937** домена за скенирање. За потребе овог рада, ако је тестиран abs.rs систем је унео следеће URL-ове за тестирање: http://test.rs, https://test.rs, http://www.test.rs, https://www.test.rs .

Тип	Фреквенција	Проценат
Добри хостови	226	2.92%
Лоши хостови	7522	97.08%
Укупно	7748	100.00%

Табела 1. *Приказ обраде валидности домена из РНИДС-овог фајла*

У циљу чишћења и детерминације који су то исправни домени за скен, за један домен прављена су 4 уноса, 1937 * 4 је 7748. Све то је пропуштено кроз систем ради валидације DNS-ом да ли хост постоји, али да ли хост одговара на HTTP односно HTTPS захтеве. Након обраде улазног фајла, У даљем истраживању као финални почетни скуп користићемо све добре хостове, односно 226 уноса.

4. АУТОМАТСКА ОБРАДА РЕЗУЛТАТА

Циљ овог експеримента је да се аутоматски обраде резултати добијени фаз тестирањем веб апликације, са фокусом на елиминацију шума у подацима и идентификовање релевантних информација.

Методом мноштва упита ка циљаној веб апликацији направљено је укупно 4388 захтева. Резултати су бележени у следећем формату:

- URL
- Величина одговора (у бајтовима)
- Број линија у телу одговора
- HTTP статус код
- Број речи у телу одговора

Пример:


```
[ {
    "url": "htts://www.[ УКЛОЊЕНО].rs/sitemap",
    "size": 54332,
    "lines": 859,
    "statusCode": 200,
    "words": 2700
  },
  {
    "url": "https://www.[
УКЛОЊЕНО].rs/newsletter",
    "size": 124281,
    "lines": 2461,
    "statusCode": 200,
    "words": 7099
  },
  {
    "url": "htts://www.[
УКЛОЊЕНО].rs/phpinfo.php",
    "size": 117454,
    "lines": 1484,
    "statusCode": 200,
    "words": 5963
  },
  … остали резултати …
]
```


Слика 3. *Подаци са шумом*

Употребом статистичке анализе фаз резултата омогућена је елиминација шума у подацима. Након завршеног теста неког хоста, анализира се фреквенција резултата добијених фаз упитом. У оквиру овог истраживања, спроведен је фаз процес у коме је упућен велики број захтева према циљаној веб апликацији. У сврху анализе, снимљени су резултати свих упућених захтева, који су се потом подвргли даљој обради. Међутим, у почетним скуповима података било је присутно доста шума, односно информација које нису значајне за даљу анализу. Како би се извукле релевантне информације, примењене су следеће методе статистичке анализе:

Анализа фреквенције појављивања вредности метрика: анализиране су фреквенције појављивања величине одговора, броја линија, HTTP статус кодова и броја речи у телу одговора на свакој појединачној страници. На основу добијених података, израђено је статистичко звоно које приказује дистрибуцију вредности за сваку метрику.

Идентификација екстремних вредности: након анализе дистрибуције, идентификоване су екстремне вредности, оне које значајно одступају од медијане, за сваку појединачну метрику. За детекцију екстрема коришћена је стандардна девијација. Критеријум за одређивање екстремних вредности био је $X<\mu-2\sigma$ за сваку метрику.

Након иницијалне анализе, екстремне вредности су третиране као значајне информације, док су остале вредности третиране као шум. Ако је неки резултат добио вредност екстрема за било коју од анализираних метрика (величина одговора, број линија, HTTP статус код, број речи), тај резултат је означен као шум и уклоњен из даље анализе.

Пример резултата након филтрирања:

```
"results": [
   { "url": "https://www.[ REDACTED ].rs/newsletter" },
   { "url": "htts://www.[ REDACTED ].rs/phpinfo.php" },
   { "url": "htts://www.[ REDACTED ].rs/sitemap" },
   ...
]
```


Слика 4. *Подаци очишћени од шума*

На слици 4. виде се аутоматски обрађени подаци, очишћени од шума. На овај начин, ефективно је елиминисан шум из података, што је омогућило фокусирање на релевантне резултате за даљу анализу.

Резултат после филтрирања, је да је укупан број релевантних резултата смањен са 4388 на 14, од чега су преостали резултати имали знатно већи број понављања у односу на медијану. Овај процес је значајно смањио количину података који захтевају даљу пажњу и анализу, док је истовремено повећана вероватноћа релевантности преосталих података.

4. ЗАКЉУЧАК

Ова студија наглашава значај и ефикасност веб фазинга као методе за откривање постојећих рањивости, али исто тако истиче потребу за проширењем опсега анализе. Континуирано праћење и ажурирање безбедносних конфигурација, заједно са стриктним контролама приступа, кључни су за смањење ризика од неауторизованог приступа и излагања осетљивих информација. Имплементирани су аутоматизовани алати за скенирање и анализу *HTTP* одговора са циљем откривања ненамерно отворених путањи програмабилног интерфејса апликације које могу представљати сигурносни ризик**.** Формулисане су препоруке за имплементацију ефикасних механизама заштите и ублажавања ризика повезаних с ненамерно отвореним путањама програмабилног интерфејса апликације, укључујући побољшану контролу приступа и конфигурацију сигурности.

Будући рад би могао да укључи фокус на веб фаз усмерен на поддомене, односно истраживање дигиталних имовина једне организације. Након идентификације имовине, могло би се прећи на континуирани систем фаза у одређеном временском интервалу. Овај приступ би омогућио дубље разумевање структуре и потенцијалних слабости унутар дигиталног простора организација, чиме се повећава ефикасност у откривању и ублажавању рањивости. Истраживање дигиталних имовина пружа темељан увид у комплексност и ширину веб присуства организације, често откривајући заборављене сегменте који могу представљати безбедносне ризике. Заборављене дигиталне имовине (енг. **Shadow IT**) у многим случајевима представљају критичне сегменте система, јер су то често неодржавани сервери и веб апликације са застарелим софтвером. Континуирани систем фаза на откривеним поддоменима и дигиталној имовини омогућава систематично откривање рањивости, које могу произаћи из конфигурацијских грешака, застарелог софтвера, или чак непажње при управљању дигиталним ресурсима.

Овим истраживањем направљени су значајни кораци за унапређење информационе безбедности на националном нивоу, подизање свести о важности сајбер безбедности и креирање сигурнијег дигиталног окружења.

5. LITERATURA

[1] Jason Andress, “The Basics of Information Security: Understanding the Fundamentals of InfoSec in Theory and Practice”, ISBN 9780128008126.

[2] Martin Fletcher, „An introduction to information risk”. The National Archives, доступно: https://www.sciencedirect.com/science/article/abs/pii/S2214212616301806?via%3Dihub

[3] Jurgen Cito, Gerald Schermann, John Erik Wittern, Philipp Leitner, Sali Zumberi, Harald C. Gall, “An Empirical Analysis of the Doker Container Ecosystem on GitHub”, доступно: https://peerj.com/preprints/2905.pdf

[4] “National Intstitute Of Standards And Technology”, доступно: https://csrc.nist.gov/glossary/term/proxy

[5] Paul Ferguson, Geoff Huston, “What is a VPN?”, доступно: https://cpham.perso.univ-pau.fr/ENSEIGNEMENT/COMMUN/vpn_ferguson.pdf

[6] https://www.ibm.com/topics/redis

[7] S. Shepler, B. Callaghan, D. Robinson, R. Thurlow, C. Beame, M. Eisler, D. Noveck, “Network File System (NFS) version 4 Protocol”, доступно: https://www.rfc-editor.org/rfc/pdfrfc/rfc3530.txt.pdf

[8] Owen Garret, “HTTP Keepalive Connections and Web Performance”, доступно: https://www.f5.com/company/blog/nginx/http-keepalives-and-web-performance

[9] “Mozilla Developer Network Web Docs”, доступно: https://developer.mozilla.org/en-US/docs/Web/Security/Transport_Layer_Security

[10] “Mozilla Developer Network Web Docs”, доступно: https://developer.mozilla.org/en-US/docs/Glossary/HTTPS

[11] Leyla Bilge, Engin Kirda, Christopher Kruege, Marco Balduzzi, “EXPOSURE: Finding Malicious Domains Using Passive DNS Analysis”, доступно: https://sites.cs.ucsb.edu/~chris/research/doc/ndss11_exposure.pdf

[12] Списак домена резервисаних за потребе државних органа и организација, доступно: https://www.rnids.rs/registar_dokumenata/2016_10_24-spisak-rezervisani_drzavni.pdf

Кратка биографија:

Душан Панић рођен је 1987. год у Лозници. Душан је софтвер архитекта са искуством у инжењеринг поузданости и информационој безбедности. Контакт: dpanic@gmail.com