

MINISTRY OF EDUCATION AND SCIENTIFIC RESEARCH



**TECHNICAL UNIVERSITY**  
OF CLUJ-NAPOCA

---

FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE  
DEPARTMENTUL CALCULATOARE

# Reverse proxy pentru prevenirea utilizatorilor de Tor si a atacurilor SQLI

License Thesis

Absolvent

**Alexandru Daniel Vid**

Conducător

**Your Supervisor's scientific title and name**

Iulie 2018





**TECHNICAL UNIVERSITY**

OF CLUJ-NAPOCA

FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE  
DEPARTMENTUL CALCULATOARE

DECANUL FACULTĂȚII  
Prof. dr. ing. Liviu MICLEA

DIRECTOR DEPARTAMENT  
Prof. dr. ing. Rodica POTOLEA

# Reverse proxy pentru prevenirea utilizatorilor de Tor si a atacurilor SQLI

## Lucrare de licență

1. Absolvent: Alexandru Daniel Vid
2. Conducător: Your Supervisor's scientific title and name
3. Conținutul lucrării: Pagina de prezentare, aprecierile coordonatorului, titlul capitolului 1, titlul capitolului 2, ..., titlul capitolului n, bibliografie, anexe, CD.
4. Locul documentării: UTCN, Cluj-Napoca
5. Consultanți: Donald Knuth, Leslie Lamport, others ...
6. Data emiterii temei: .....
7. Data predării: .....

Semnătură Conducător  
Your Supervisor's scientific title and name

Semnătură Absolvent  
Alexandru Daniel Vid

Iulie 2018





## Declarație pe proprie răspundere privind autenticitatea lucrării de licență

Subsemnatul *Alexandru Daniel Vid*, legitimat cu *CI* seria *XH* numărul *866549*, *CNP 1950417055056*, autorul lucrării *Reverse proxy pentru prevenirea utilizatorilor de Tor și a atacurilor SQLI* elaborată în vederea susținerii examenului de finalizare a studiilor de masterat la Facultatea de Automatică și Calculatoare, Departamentul Calculatoare, Specializarea *Calculatoare* din cadrul Universității Tehnice din Cluj-Napoca, sesiunea *Iulie* a anului universitar *20XX/20XX*, declar pe proprie răspundere, că această lucrare este rezultatul propriei mele activități intelectuale, pe baza cercetărilor mele și pe baza informațiilor obținute din surse care au fost citate în textul lucrării și în bibliografie.

Declar că această lucrare nu conține porțiuni plagiate, iar sursele bibliografice au fost folosite cu respectarea legislației române și a convențiilor internaționale privind drepturile de autor.

Declar, de asemenea, că această lucrare nu a mai fost prezentată în fața unei alte comisii de examen de licență sau disertație.

În cazul constatării ulterioare a unor declarații false, voi suporta sancțiunile administrative, respectiv, *anularea examenului de licență*.

Cluj-Napoca  
data

Semnătură  
Absolvent



## Rezumat

*Reverse proxy pentru prevenirea utilizatorilor de Tor si a atacurilor SQLI* incapsuleaza trasaturile normale ale unui reverse proxy oferind in plus protectie impotriva posibilelor atacuri de tipul SQL injection si blocarea ip-urilor utilizate de Tor, catre un server HTTP/HTTPS. Detectia atacurilor de tipul SQL injection se realizeaza prin analiza URI-urilor trimise de catre clienti, in relatie cu un model antrenat anterior folosind machine learning, astfel aceste request-uri nici nu vor ajunge la server, iar blocarea utilizatorilor de Tor se realizeaza prin folosirea unei liste de ip-uri(blacklist). Ambele implementari asigura adaugarea cu usurinta de noi detectii, lista de ip-uri blocate fiind accesibila utilizatorului atat pentru vizualizare cat si pentru editare.





# Cuprins

<b>Listă de tabele</b>	<b>iii</b>
<b>Listă de figuri</b>	<b>v</b>
<b>1 Introducere</b>	<b>1</b>
1.1 Context . . . . .	1
1.2 Motivație . . . . .	2
1.3 Structura lucrării . . . . .	3
<b>2 Obiective și specificații</b>	<b>5</b>
2.1 Obiective . . . . .	5
2.2 Specificații . . . . .	6
2.2.1 Specificații funcționale . . . . .	7
2.2.2 Specificații non-funcționale . . . . .	8
<b>3 Studiu bibliografic</b>	<b>9</b>
3.1 Abordări similare . . . . .	9
3.2 Tehnici/Tehnologii/Surse folosite . . . . .	14
<b>4 Fundamente teoretice</b>	<b>17</b>
4.1 Reverse proxy . . . . .	17
4.2 Support vector machine . . . . .	18
4.3 SQL injection . . . . .	19
4.4 Obținerea adreselor rețelei Tor . . . . .	20
4.5 Sistem de prevenire a intruziunilor . . . . .	21
<b>5 Analiză și proiectare</b>	<b>25</b>
5.1 Cerintele sistemului . . . . .	25
5.2 Specificatiile cazurilor de utilizare . . . . .	27
5.2.1 Actori, stakeholders si interese . . . . .	27
5.2.2 Basic flow . . . . .	27
5.2.3 Alternative flow . . . . .	28
5.3 Arhitectura sistemului si modulele principale . . . . .	29

<b>6</b>	<b>Implementation Details</b>	<b>31</b>
<b>7</b>	<b>Tests and Results</b>	<b>33</b>
7.1	Functional Tests . . . . .	33
7.2	Performance Tests . . . . .	33
<b>8</b>	<b>User Manual</b>	<b>35</b>
<b>9</b>	<b>Conclusions</b>	<b>37</b>
	<b>Bibliografie</b>	<b>39</b>
<b>A</b>	<b>Diverse anexe</b>	<b>43</b>
<b>B</b>	<b>Demonstrații matematice detaliate (dacă există)</b>	<b>45</b>
<b>C</b>	<b>Pseudo-cod sau cod (dacă există)</b>	<b>47</b>
<b>D</b>	<b>Articole publicate</b>	<b>49</b>

# Listă de tabele



# Listă de figuri

1.1	Incapsula diagrama reverse proxy . . . . .	1
2.1	Cutia neagra a sistemului . . . . .	7
3.1	Administrarea securitatii unei aplicatii . . . . .	10
3.2	Tipuri de trasaturi ale limbajului SQL . . . . .	13
3.3	Arhitectura unui sistem de clasificare a request-urilor HTTP . . . . .	14
4.1	Folosirea unui reverse proxy in arhitectura unei aplicatii. . . . .	17
4.2	Influentele aduse algoritmului de modifiacrea parametrului sigma in algoritmul de antrenare. . . . .	19
4.3	Exemplu de atac realizat prin SQL injection. . . . .	20
4.4	Exemplu de trafic realizat prin reseaua Tor. . . . .	21
4.5	Integrarea unui sistem de prevenire a intruziunilor intr-o retea. . . . .	22
5.1	Basic flow pentru evenimente . . . . .	27
5.2	Principalele module ale sitemului propus . . . . .	29



# Capitolul 1

## Introducere

### 1.1 Context

În general, tentativele de exploatarea vulnerabilităților unei aplicații vin sub formă de input către aceasta, generate de către un atacator care intenționează să întrerupă activitatea sau să preia controlul aplicației. Un sistem de prevenire a intruziunilor (IPS) are rolul de a sta între aplicație și clienții acesteia, și de a preveni exploatarea unor astfel de vulnerabilități.

Prin folosirea unui reverse proxy pentru accesarea resurselor unui server de către clienți, poate să aducă numeroase beneficii procesului de administrare a serverului [1]. Spre deosebire de un forward proxy care este un intermediar pentru o serie de clienți prestabiliți, pentru a accesa orice server, un reverse proxy este un intermediar pentru o serie de servere prestabilite pentru a fi accesate de orice client. Unul dintre avantajele folosirii unui reverse proxy este centralizarea întregului trafic al serverului/serverelor într-un singur punct de acces, aceasta fiind și principala caracteristică exploatată de acest proiect pentru filtrarea IP-urilor nedorite (în cazul nostru cele utilizate de rețeaua Tor) și verificarea URI-urilor pentru posibile atacuri de SQL injection.

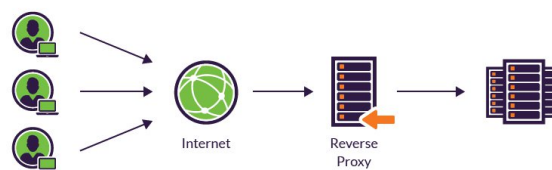


Figura 1.1: Incapsula diagrama reverse proxy

Figura 1.1 ilustrează modul de funcționare al unui reverse proxy în relație cu serverele aferente și posibili clienți.

Conform topului alcatuit de fundatia OWASP cu cele mai mari 10 riscuri ale aplicatiilor in 2017 [2], SQL injection e considerat a fi cea mai mare vulnerabilitate a aplicatiilor web. Acest lucru se datoreaza fatului ca mare parte din aceste aplicatii se bazeaza pe procesearea continutului furnizat de catre utilizatori. Atacurile de tipul SQL injection costau in faptul ca datele furnizate de catre utilizator sunt introduse in interogari SQL, unde acestea sunt tratate ca si cod executabil [3]. Aplicatiile web vulnerabile la sql injection pot permite unor utilizatori neautorizati sa faca interogari intr-o baza de date asupra unor date la care nu ar avea acces in mod normal. Folosind acest tip de comportament neautorizat, un astfel de utilizator poate sa obtina accesul la informatii sensibile ale clientilor, dar si a administratorilor aplicatiei, precum credentiale sau date personale. Aceasta vulnerabilitate poate sa duca la furt de identitate sau fraudă.

In cazul retelei Tor, aceasta le permite utilizatorilor sa navigheze pe internet anonim. Anonimitatea online este importanta in multe cazuri aplicatiile web trebuie sa stie cine se conecteaza la aceasta pentru a le putea determina intentiile. Numele de Tor vine de la "the onion router" care sugereaza modul de operare al retelei. Fiecare participant la retea devine un nod de transfer, iar traficul retelei traverseaza o serie de astfel noduri pana sa ajunga la nodul de iesire ce creaza conexiunea cu destinatia dorita. Pachetele sunt criptate in mai multe "straturi", fiecare nod decriptand un singur strat de unde poate afla doar destinatia nodului urmator. Cand pachetul ajunge la ultimul nod, acesta trimite continutul la destinatie fara sa dezvalui identitatea sursei. Aceasta anonimitate useaza desfasurarea atacurilor online. Conform datelor din reseaua organizatiei CloudFlare 94% din traficul provenit din reseaua Tor este alcatuit din request-uri automate de origine malitioasa [4].

## 1.2 Motivație

In piata actuala exista multe sisteme de prevenire a intruziunilor ce ofera atat caracteristicile unui reverse proxy, cat si cele de securitate. Aceste caracteristici sunt oferite fie ca si produse individuale, fie ca si produse ce le incorporeaza pe ambele. Cu toate acestea, produsele de acest gen sunt in general scumpe, au o logica mascata de detectarea a posibilelor probleme de securitate si sunt greu de inteles si de configurat de catre utilizator dupa propriile nevoi.

Prin oferirea utilizatorilor posibilitatea de a intelege si modifica modul de functionare a unui astfel de sistem poate rezulta in sisteme mult mai eficiente si rapide, dedicate pentru preferintele si nevoile aplicatiei fiecarui utilizator in parte. Spre exemplu, un utilizator poate sa decida ca nu are nevoie de functionalitaile de detectie impotriva atacurilor de tip SQL injection pentru o anumita aplicatie, intrucat aceasta nu prezinta vulnerabilitati din acest punct de vedere, nefolosind o arhitectura bazata pe baze de date. In cazul acesta prin eliminarea unui astfel de modul, se elimina si verificarile aferente asupra request-urilor clientilor, imbunatatind astfel performantele sistemului.

*Reverse proxy pentru prevenirea utilizatorilor de Tor si a atacurilor SQLI ofera un*



sistem configurabil după preferințele utilizatorilor. Utilizatorul poate configura detectia bazata pe analiza request-ului primit de la client, acesta poate sa aleaga care module sa fie folosite pentru detectie, permitand si eventuala adaugare de noi module(cat timp acestea respecta anumite conditii de structura), meniurile din interfata utilizator fiind generate in mod dinamic in functie de modelele prezente. Utilizator poate, de asemenea sa configureze si lista ip-urilor blocate, permitandu-i-se sastearga din cele existente, respectiva sa aduge unele noi, după bunul plac.

## **1.3 Structura lucrării**

In aceasta sectiune se prezinta structura lucrarii pe capitole si o scurta descriere a continutului acestora:

Primul capitol 1, prezinta o scurta introducere despre proiect, contextul acestuia si motivatia pentru implementarea sistemului propus.

Capitolul 2 prezinta obiectivele lucrarii, specificatiile sistemului, motivand deciziile luate in implementarea sistemului, cerintele functionale si nonfunctionale necesare implementarii sistemului.

Capitolul 3 descrie alte abordari similare ale problemelor tratate de proiectul propus, prin evidentierea asemanarilor si diferentelor dintre acestea si se explica tehnologiile si metodele folosite de proiect.

In capitolul 4 sunt evidentiate si explicate pe scurt aspectele teoretice pe care se bazeaza proiectul.

Capitolul 5 descrie design-ul proiectului si cuprinde: cerintele sistemului, specificatiile cazurilor de utilizare, arhitectura sistemului, comportamentul sistemului, datele utilizate de sistem, dependintele sistemului si algoritmi esentiali si metodele folosite. Descrierea acestora se realizeaza prin asocierea cu diagramelor aferente.



# Capitolul 2

## Obiective și specificații

Acest capitol prezintă obiectivele lucrării, motivând deciziile luate în implementarea sistemului, specificațiile sistemului, cerințele functionale și nonfunctionale necesare implementării sistemului.

### 2.1 Obiective

Elaborarea unor măsuri de securitate împotriva anumitor tipuri de atacuri sau dezvoltarea unei logici de filtrare a clienților serviti de către aplicație este posibilă și în partea de implementare a serverului, ceea ce ar putea oferi și o eventuală creștere în performanțele aplicației, eliminând astfel posibile interceptări suplimentare și validări ale traficului. Într-o astfel de implementare presupune o arhitectură mult mai complexă pentru partea de server și individual cunoscute avansate despre securitate, posibilele vulnerabilități la care acesta poate să fie predispus, precum și modalități de combatere ale acestora.

O soluție mult mai simplă la această problemă este folosirea unui modul care să implementeze aceste funcționalități separat. Mare parte din produsele de pe piață, care îndeplinesc aceste caracteristici sunt destul de scumpe și nu oferă foarte multă libertate din punctul de vedere al configurării securității dorite de către utilizator asupra produsului său. În cazul IP-urilor blocate, multe aplicații nu oferă libertatea utilizatorilor de a edita listele de referință ale detectiilor, acestea fiind actualizate automat conform unor date interne, iar eventualele abateri de la aceste reguli se realizează prin adăugarea de excepții. În ceea ce privește validarea request-urilor primite de la clienți, mare parte din aceste sisteme nu oferă o protecție configurabilă. Cum s-a precizat mai sus, acest tip de sisteme pot să introducă mici întârzieri datorate validărilor suplimentare asupra request-urilor primite de la clienții produsului, însă în unele cazuri anumite validări sunt făcute inutile întrucât produsele respective nuputând să prezinte vulnerabilități de acel fel.

*Reverse proxy pentru prevenirea utilizatorilor de Tor și a atacurilor SQLI* are obiectivul de a oferi o componentă gratuită cât mai ușor de integrat și de configurat de către utilizator după preferințele produsului său, care să faciliteze o protecție cât mai eficientă

cu suport pentru îmbunătățiri. Sistemul trebuie să fie ușor de instalat și de configurat, oferindu-i utilizatorului o interfață clară, sugestivă și ușor de folosit prin care să interacționeze cu acesta. Detectiile sistemului trebuie să fie activabile, utilizator putând alege în momentul pornirii sistemului ce vulnerabilități să fie tratate de acesta. Lista ip-urilor blocate trebuie să fie ușor de vizualizat și editabilă, permitând utilizatorului să își impună cu ușurință propriile reguli asupra modului de funcționare a sistemului. Pentru realizarea detecției împotriva atacurilor de tipul SQL injection se impune prelucrarea unor date reale pentru antrenarea modelului de machine learning. Prin folosirea unor date provenite din atacuri reușite sau tentative de atacuri reale, se poate crea o precizie mult mai bună pentru o clasificare cât mai precisă a posibilelor atacuri. Sistemul trebuie de asemenea să fie construit modular pentru a permite realizarea de modificări cu ușurință, iar încărcarea detecțiilor realizată dinamic, permitând astfel ca adăugarea de noi detecții să fie cât mai simplă.

## 2.2 Specificații

*Reverse proxy pentru prevenirea utilizatorilor de Tor și a atacurilor SQLI* trebuie să fie capabil să servească ca și un reverse proxy pentru un server, să blocheze atacurile de tip SQL injection asupra lui și să nu permită conectarea clienților cu ip-uri utilizate de rețeaua tor la acesta.

*Reverse proxy pentru prevenirea utilizatorilor de Tor și a atacurilor SQLI* va procura utilizatorului o interfață grafică prietenoasă, ușor de folosit, prin intermediul căreia, acesta va putea să seteze mediul de rulare al sistemului. Interfața va permite setarea specificațiilor server-ului, adresa și portul pe care acesta accepta conexiuni, dar și a interfețelor prin intermediul cărora se pot realiza conexiuni la server.

În interfața grafică se vor afișa și eventualele detecții realizate de produs. Într-o fereastră separată utilizatorul trebuie să aibă posibilitatea să vizualizeze toate deciziile sistemului și motivele din spatele deciziilor, permitând astfel acestuia să înțeleagă modul de funcționare, respectiv să raporteze sau să modifice sistemul (în cazurile în care i se oferă această posibilitate) când comportamentul acestuia nu se află în conformitate cu nevoile sau cerințele sale.

În momentul configurării modului de rulare al sistemului, utilizatorul trebuie să aibă și posibilitatea de a impune ce module de securitate să fie folosite de acesta în timpul rularii. Pentru a eficientiza cât mai mult sistemul, utilizatorul poate să aleagă care sunt modulele de interes pentru propria aplicație, evitând astfel validarea unor evenimente ce nu prezintă interes pentru acesta.

În timpul rularii sistemul va asculta pe interfețele setate de către utilizator pentru posibile cereri de conexiuni la server-ul setat. În funcție de modulele alese în momentul pornirii, acesta va verifica sau nu adresa clientului validând astfel conexiunea. În cazul în care adresa clientului se află pe lista neagră de adrese, conexiunea acestuia este refuzată, iar aplicația înregistrează această decizie în fereastra de evenimente vizibilă utilizatorului.

Dupa realizarea conexiunii la server, fiecare request trimis de clienti catre acesta va fi evaluat conform modulelor configurate. Daca requesturile sunt considerate ca fiind "curate" acestea sunt trimise mai departe la server. In caz contrar, clientului i se intoarce un cod de eroare, iar requestul nu va mai fi trimis mai departe catre server, de asemenea inregistrand evenimentul in fereastra de evenimente vizibila utilizatorului.

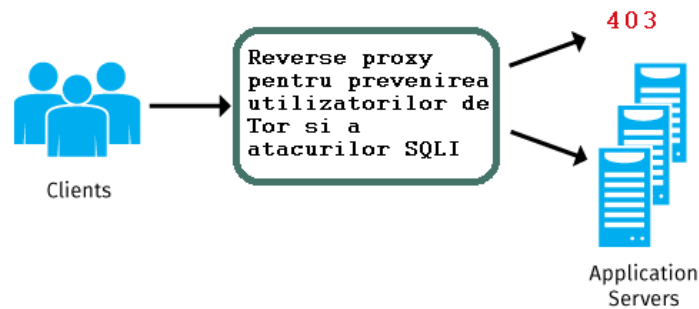


Figura 2.1: Cutia neagra a sistemului

Figura 2.1 prezinta cutia neagra a sistemului propus.

### 2.2.1 Specificații funcționale

Sistemul trebui sa prezinte o interfata grafica usor de folosit de catre utilizator si sa fie capabil sa redirectioneze traficul interceptat catre un anumit server, clasificand si filtrand traficul malitios. Pentru a atinge obiectivele proiectului, urmatoarele cerinte functionale trebuie indeplinite:

- Sa realizeze conexiunea la un server HTTP/HTTPS si sa redirectioneze traficul primit catre acesta.
- Sa intercepteze traficul venit pe o anumita interfata si port prestabilit.
- Sa prelucreze request-urile primite de la clineti intr-un format specific clasificatorului de SQL injection.
- Sa nu redirectioneze requesturile clasificate ca si SQL injection.
- Sa blocheze conectarea clientilor ce folosesc ip-uri clasificate ca ip-uri de Tor.
- Sa permita utilizatorului sa editez si sa vizualizeze lista ip-urilor de Tor.
- Sa prezinte in interfata grafica toate interventiile rezlitate asupra traficului(blocari de conexiuni sau de request-uri).
- Sa permita utilizatorului sa configureze modul de operare al sistemului.

### 2.2.2 Specificații non-funcționale

Sistemul trebuie, de asemenea, să aibă următoarele caracteristici non-funcționale pentru a realiza obiectivele specificate:

- Sa fie usor de instalat si de folosit pentru orice utilizator, oricat de neexperimentat.
- Sa poata intercepta traficul de pe orice/oricate interfete disponibile.
- Sa poata rula pe orice sistem de operare Windows cu Python2 instalat.
- Sa aiba o rata de blocare de 100% a ip-urilor de pe lista neagra, iar in cazul detectiei de SQL injection sa nu aiba detectii false pozitive mai mari 2-3% si o acuratete generala de peste 90%

# Capitolul 3

## Studiu bibliografic

În acest capitol sunt prezentate alte abordări similare ale problemelor tratate de proiectul propus, prin evidențierea asemănarilor și diferențelor dintre acestea și se explică tehnologiile și metodele folosite de proiect.

### 3.1 Abordări similare

Precum Richard Bassett, Cesar Urrutia și Nick Ierace susțin în articolul **Intrusion prevention systems** [5] ”sistemele de prevenire a intruziunilor sunt o componentă importantă a sistemelor de protecție IT, iar fără această tehnologie, datele noastre și rețelele ar fi mult mai predispuse activităților malicioase”.

În general tentativele de exploatare a vulnerabilităților unei aplicații vin sub formă de input către o aplicația țintă. Acest input fiind generat de către un atacator ce intenționează să controleze sau să îi întrerupă activitatea. În cazul unui atac reușit, un astfel de atacator poate să dezactiveze temporar aplicația (atacuri de tipul denial of service) sau poate accesa, altera sau executa date/cod în interiorul aplicației. Un sistem de prevenire a intruziunilor are rolul de a examina traficul destinat unei aplicații și de a intercepta și bloca astfel de tentative [6].

Un sistem de prevenire a intruziunilor este, de regulă, folosit alături de un sistem firewall respectiv alături de un sistem de detectare a intruziunilor. Deși au scopuri asemănătoare, aceste sisteme au funcționalități diferite și rezolvă diferite probleme de securitate. Un sistem de prevenire a intruziunilor este cel mai bine comparat cu sistemele de tip firewall. Un sistem firewall tipic este constituit dintr-o serie de reguli ce permit traficului să treacă. Aceste reguli sunt sub formă ”dacă traficul îndeplinește anumite condiții poate trece”, însă dacă nu există nici o regulă care să potrivească anumite pachete, acestea sunt blocate. Asemeni sistemelor firewall, sistemele de prevenire a intruziunilor prezintă un set de reguli de filtrare a pachetelor, cererilor sau a clienților, însă aceste reguli sunt de regulă reguli de blocare. Astfel, dacă un anumit pachet nu potrivește nici o regulă a sistemului de prevenire a intruziunilor îl lasă să treacă [7].

Spre deosebire de sistemele de tip firewall sau cele de prevenire a intruziunilor, care ofera control utilizatorului asupra traficului ce trece prin retea, sistemele de detectie a intruziunilor permite acestuia sa vizualizeze evenimentele din retea. Precum si Joel Snyder sustine in articolul **Do you need an IDS or IPS, or both?** [7] sistemele de detectie a intruziunilor ofera unui utilizator facilitati asemanatoare unui analizator de pachete [8], insa din perspectiva de securitate a retelei. Aceste informatii furnizate de catre sistem ii permit utilizatorului sa decopere:

- Incalcari ale politicilor de securitate, precum utilizatori sau sisteme ce desfasoara activitati ce incalca politicile prestabilite.
- Posibile sisteme infectate ce folosesc reteaua pentru a se raspandi sau sa atace alte sisteme.
- Scurgeri de informatie cauzate de infectarea unor sisteme cu malware sau de utilizatori rau intentionati.
- Erori de configurare in sisteme sau aplicatii cu setari de securitate incorecte sau configurari proaste ce consuma prea multa banda de retea.
- Detectarea unor clienti sau servere ce acceseaza sau sunt accesate in mod neautorizat, respectiv aplicatii malitioase ce fac asta.

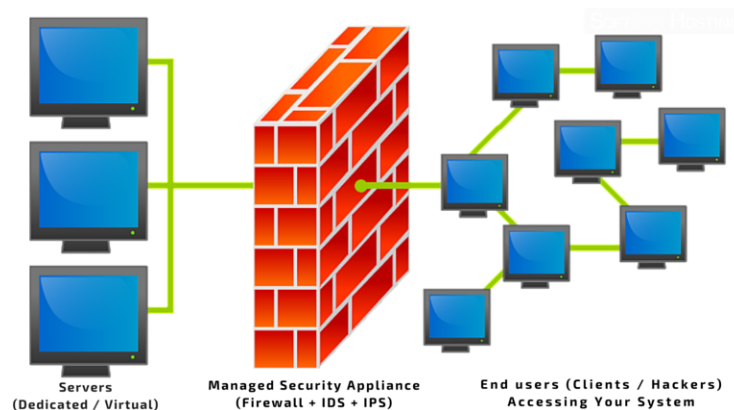


Figura 3.1: Administrarea securitatii unei aplicatii

Figura 3.1 prezinta administrarea securitatii unei aplicatii folosind combinatia dintre cele trei sisteme.

In comparatie cu sistemele de detectie a intruziunilor care sunt sisteme pasive si scaneaza reteaua fara sa interfereze cu traficul, sistemele de prevenire a intruziunilor sunt plasate intre server si clienti, alterand in mod automat traficul in cazul in care acesta



declanseaza una din regulile prezente in sistem. Precum sunt prezentate si in articolul **What is an intrusion prevention system?** [6], printre functionalitatile unui sistem de prevenire a intruziunilor se numara:

- Notificarea unui administrator de retea in cazul in care un sau mai multe reguli sunt declansate.
- Oprirea pachetelor considerate malitioase pentru retea.
- Blocarea unor utilizatori prin excluderea adreselor ip ale acestora.
- Resetarea unor conexiuni.

In cea ce priveste functionalitatile oferite de un sistem de prevnire a intruziunilor, acestea sunt specifice tipului sistemului. Conform autorului articolului **Intrusion Prevention System (IPS): Definition & Types** [9], Beth Hendricks, exista patru tipuri primare de astfel de sisteme:

- Network-based: Protejeaza intraga retea.
- Wireless: Protejeaza doar reteaua wireless.
- Network behavior: Examineaza traficul din retea.
- Host-based: Software cu scopul de a proteja un singur calculator.

Sistemele de tipul Network-based(reprezentand si categoria in care se incadreaza *Reverse proxy pentru prevenirea utilizatorilor de Tor si a atacurilor SQLI*) presupun implementarea unor senzori in retea carea captureaza si analizeaza traficul ce trece prin acesta. Acesti senzori sunt plasati in puncte cheie a retelei pentru a putea captura in timp reala traficul, iar in cazul interceptarii unor activitati malitioase sa poata interveni imediat, fara sa scada performanta retelei. Aceste sisteme ofera protectie retelei indiferent de dimensiunile sau cresterea acesteia, adaugarea de noi device-uri fiind posibila fara sa necesite adaugarea de noi senzori. Adaugarea de noi senzori fiind nevoita doar in cazul in care traficul retelei depaseste capacitatea de procesare a senzorilor curenti, infulentand astfel performantele retelei [10].

In functie de nevoi, un sistem de prevenire a intruziunilor poate sa ofere diferite optiuni de protectie pentru diferite parti ale retelei. Unele sunt capabile sa opreasca traficul malitios sau sa limiteze latimea de banda catre anumite parti ale retelei. Conform [11] aceste sisteme pot oferi protecti impotriva urmatoarelor tipuri de atacuri:

- **ICMP Storms:** un volum mare de ecouri ICMP pot sa indice activitati malitioase precum cineva ce scaneaza reteaua.
- **Ping to Death:** un utilizator poate sa modifice comanda de ping, astfel incat sa trimita un numar mare de pachete de dimensiune mare catre o destinatie tinta pentru a o scoate din uz.

- **SSL Evasion:** unele atacuri se pot folosi de criptarea SSL pentru a evita dispozitivele de securitate, intrucat in general acestea nu sunt decriptate.
- **IP Fragmentation:** consta in exploatarea faptului ca pachetele sunt descompuse in fragmente pentru a staisface cerintele de dimensiune a retelelor traversate, inundand o destinatie tinta cu fragmente false pentru a ii consuma resursele.
- **SMTP mass mailing attacks:** un sistem infectat poate sa se foloseasca de de email-ul utilizatorului pentru a se raspandi, rezultand intr-un trafic mare destinat serverului de mail.
- **DoS/DDoS attacks:** cu scopul de a face o resursa indisponibila utilizatorilor, este realizata prin inundarea sistemului tinta cu un numar mare request-uri de la unul sau mai multe(in cazul DoS distribuit - DDoS) sisteme malitioase.
- **SYN Flood attacks:** atacatorul trimite un numar mare de pachete de initiere a unei conexiuni fara sa mai raspunda ulterior, epuizand astfel resursele de memorie.
- **Http obfuscation:** pentru a evita sa fie detectate de anumite sisteme de protectie, unele atacuri folosesc tehnici de ofuscarea a request-urilor HTTP.
- **Port Scanning:** este folosit pentru descoperi ce porturi sunt deschise pe un sistem, ulterior permitandu-i atacatorului sa stie ce vulnerabilitati ar putea prezenta sistemul.
- **ARP Spoofing:** un atacator trimite in retea pachete false de ARP legandu-si propria adresa MAC de adresa IP a unui alt sistem. Ca urmare, atacatorul va primi pachete destinate sistemului cu adresa IP folosita in pachetul de ARP.
- **CGI Attacks:** un atacator poate sa trimita request-uri malitioase, determinand destinatia sa trateze request-ul primit ca si cod executabil, oferindu-i atacatorului acces pe sistem.
- **Buffer Overflow attacks:** presupune ca atacatorul sa depaseasca limitele unui buffer de lungime fixa, excesul de date ajungand sa suprascrie zone adiacente de memorie rezultand in caderea sistemului sau dandu-i atacatorului oportunitatea sa ruleze cod propriu.
- **OS Fingerprinting attacks:** presupune ca atacatorul sa descopere ce sistem de operare ruleaza pe un sistem si folosindu-se de aceasta informatie sa exploateze vulnerabilitati specifice acelui sistem de operare.

Sistemul propus, *Reverse proxy pentru prevenirea utilizatorilor de Tor si a atacurilor SQLI* implementeaza un sistem de prevenire a intruziunilor folosindu-se de un reverse proxy pentru a intercepta tot traficul care intra si iese din retea( reprezentand senzorii ce au rol



## 3.2 Tehnici/Tehnologii/Surse folosite

Pentru realizarea sistemului propus s-au folosit doua limbaje de programare: Python2/3(pentru partea de back end) si C#(pentru partea de front end). In partea de back end a proiectului se realizeaza implementarea unui reverse proxy pentru a intercepta traficul uneia sau mai multor interfețe, un modul pentru clasificarea request-urilor impotriva atacurilor SQL injection si un modul pentru generarea listei negre de ip-uri ce utilizeaza frecvent rețeaua Tor. Toate aceste componente sunt realizate prin utilizarea de librării open-source pentru a usura si eficientiza munca precum: twisted [16], beautiful soup [17], libsvm [18].

Motivul utilizării atât limbajului Python3 cât și Python2 este datorat diferențelor de module și librării open-source disponibile pentru cele două limbaje dar și a faptului că suportul pentru Python2 se încheie în anul 2020. Conform documentațiilor oficiale [19] și [20], dar și articolului *Python 2 to python 3: why, and how hard can it be?* de Tim Grey [21], între cele două versiuni nu sunt modificări majore, însă în anumite cazuri pot exista librării care să ofere doar suport pentru una dintre acestea.

În realizarea modulului pentru clasificarea request-urilor impotriva atacurilor SQL injection s-a folosit o colecție de date reale atât de atacuri cât și de trafic curat. Pentru uniformizarea acestor date și pentru a trata tentativele de păcălire a clasificatorului prin codarea unor caractere în valoarea lor în cod hexadecimal (exemplu 'https://www.google.ro/search?q=a' echivalent cu 'https://www.google.ro/search?q=%61') datele au fost preprocesate și transformate în întregime în coduri hexadecimale [22]. În procesarea datelor, pentru indentificarea trasaturilor relevante, s-au indentificat caracterele specifice limbajului [23] și cuvintele cheie rezervare [24]. Ulterior, pentru antrenarea modelului de support vector machine și pentru clasificarea noilor request-uri s-a folosit software-ul open-source libsvm [25].

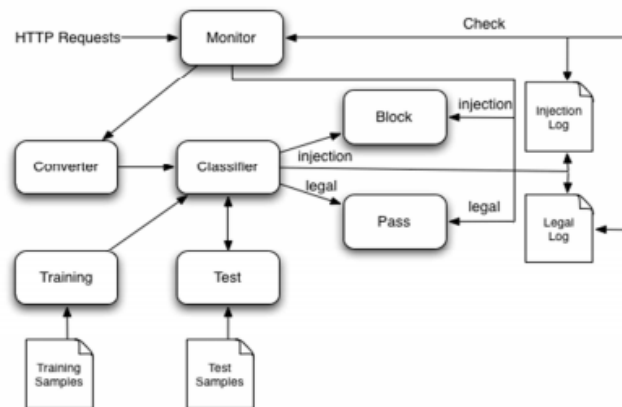


Figura 3.3: Arhitectura unui sistem de clasificare a request-urilor HTTP

Figura 3.3 arhitectura unui sistem de clasificare a request-urilor HTTP de un sistem bazat pe machine learning. Structura este prezentată în lucrarea prezentată și anterior

**Preventing SQL Injection Attack Based on Machine Learning** [12]. Acesta structura a reprezentat un model de pornire in realizarea modului de prevenire a atacurilor SQL injection, implementarea modului incercand sa aduca imbunatatiri de performanta prin modificarea algoritmului folosit pentru antrenarea modelului de support vector machine dar si prin filtrarea trasaturilor propus in lucrare in conformitate cu raportul dintre obijnuinta de aparitie a acestora atat in request-urile ce intentioneaza sa execute un atac cat si in cele curate.

Pentru blocarea ip-urilor utilizate de retea Tor s-a folosit un script scris in Python3. Programul interogheaza periodic(din 6 in 6 ore) informatiile oferite de *Tor Network Status* [26] indentificand astfel nodurile cu un "Uptime" mai mare de 7 zile in parcursul unei luni. Blocarea ip-urilor se realizeaza prin compoarea cu o astfel de lista generata lunar.

Componenta ce incorporeaza toate modulele de protectie, este cea de reverse proxy. Aici este monitorizat tot traficul ce vine de pe o anumita interfata(una sau mai multe, in functie de configuratia utilizatorului) si este trecut prin toate modulele disponibile pentru a verifica conditiile de securitate. Pentru testarea daca o adresa ip este utilizata frecvent de retea Tor, in momentul in care un client doreste sa realizeze o conexiune la serverul protejat de sistem, adresa ip a acestuia este verificata sa nu se afle pe lista ip-urilor blocate. Pentru actualitate, lista adreselor ip blocate este actualizata periodic cu adresele ip utilizate frecvent de retea Tor in ultima luna. Modulul de prevenire a atacurilor SQL injection este integrat tot in componenta de reverse proxy, insa evaluarea request-urilor este facuta dupa realizarea conexiunii intre client si server. Request-urile primite de catre server sunt tratate asemanator celor folosite pentru antrenarea modului de support vector machine, insa pentru clasificarea acestora este folosit modulul antrenat in faza initiala si software-ul de prezicere oferit tot de libsvm [18].



# Capitolul 4

## Fundamente teoretice

În acest capitol sunt evidențiate și explicate pe scurt aspectele teoretice pe care se bazează proiectul.

### 4.1 Reverse proxy

Un reverse proxy este un server intermediar care trimite mai departe request-urile pentru conținut de la mai mulți clienți nedefiniți către unul sau mai multe servere. Un reverse proxy este un tip de proxy care în mod normal este situat în spatele unui firewall într-o rețea internă și redirecționează traficul clienților către serverele asociate. Acesta introduce un nivel în plus de abstractizare și control, asigurând controlul fluxului de trafic [27].

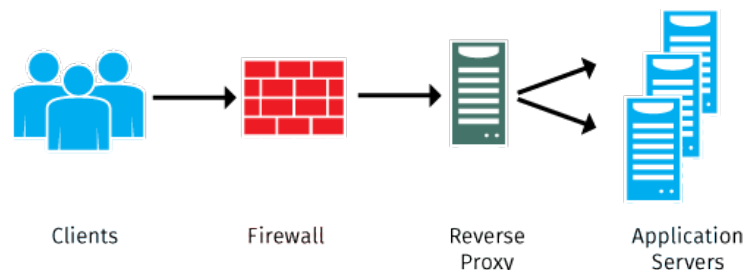


Figura 4.1: Folosirea unui reverse proxy în arhitectura unei aplicații.

Figura 4.2 prezintă modalitatea de integrare a unui reverse proxy în implementarea arhitecturii de back end a unei aplicații.

Cele mai obișnuite caracteristici ce pot fi oferite de utilizarea unui reverse proxy sunt:

- Load balancing - un reverse proxy poate să distribuie request-urile primite de la clienți, astfel încât nici un server să nu fie copleșit de requesturi. În cazul în care un

server este supraincarcat cu request-uri sau este cazut, acesta poate sa redirectioneze traficul catre alte servere functionale.

- Web acceleration - un reverse proxy poate sa realizeze compresia datelor sau sa memoreze in memoria cache continut ce este frecvent accesat sau poate sa realizeze operatiile de criptare SSL executate in mod normal de server, imbunatatind astfel in mod considerabil viteza de comunicare dintra client si serverul destinatie.
- Securitate si anonimitate - prin interceptarea request-urilor primite de catre server, acesta asigura anonimitatea serverului actionand ca un nivel extra de securitate. De asemenea se asigura ca mai multe servere pot fi accesate prin intermediul unui punct comun, indiferent de structura retelei interne.

## 4.2 Support vector machine

Algoritmul de machine learning, support vector machine reprezinta un model obtinut prin folosire a diversi algoritmi pentru antrenarea acestuia, folosit pentru a clasifica date. Acest model intra in categoria de invatare supravegheata('supervised learning'), intrucat pentru obtinerea lui se foloseste un set de date ca si exemplu, date pe care modelul le va folosi ca referinta pentru clasificarea noilor evenimente.

Realizarea unui astfel de model se obtine urma executarii unui proces elaborat ce implica mai multi pasi:

- Primul pas reprezinta indentificarea datelor relevante in cea ce priveste problema tratata(setul de antrenare). In conformitate cu scopul clasificarii unor evenimente/-date in doua(sau mai multe) categorii, initial trebuie indentificate o serie de astfel de evenimente si categorizate de catre utilizator in evenimente ce sigur apartin fiecarei dintre categoriile tinta.
- Dupa obtinerea datelor de antrenare, trebuie indentificate toate trasaturile relevante din aceste date, trasaturi care sa fie cat se poate de specifice fiecarei categorii in parte. Fiind recomandata evitarea trasaturilor ce sunt prezente in mare parte din date sub aceasi forma (ex:caracterul '=' sau '?' intr-un URI folosit pentru clasificarea atacurilor SQL injection), indiferent de categoria din care acestea fac parte.
- Dupa obtinerea trasaturilor specifice datelor de antrenare, se realizeaza antrenarea modelului folosind un algoritm specific. In cazul proiectului propus s-a folosit algoritmul gata implementat, furnizat de biblioteca open source LIBSVM [18]. Pentru obtinerea modelului datele de antrenare au fost procesate folosind un kernel gaussian. Un kernel gaussian reprezinta modul in care modelul proceseaza datele de antrenare astfel incat clasificarea noilor date sa fie realizata prin calcularea similaritatilor dintre acestea si cele de antrenare. In calcularea similaritatii dintre aceste doua tipuri de date, un parametru foarte important este sigma. Acest parametru este ales pentru



intrg setul de date, iar valoarea lui este direct proportionala cu gradul de similaritate pe care algoritmul il va asocia la doua evenimente/date diferite.

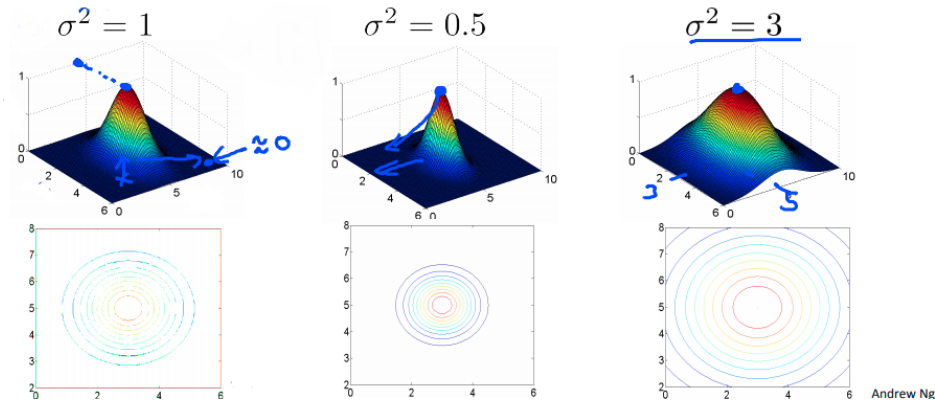


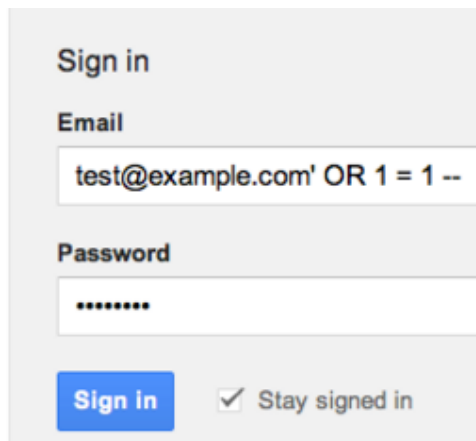
Figura 4.2: Influențele aduse algoritmului de modificare a parametrului sigma in algoritmul de antrenare.

Figura 4.2 prezinta cum inflenteaza clasificarea unui nou eveniment valoarea parametrului sigma din formula algoritmului de support vector machine. Figura 4.2 a fost preluata din slide-urile cursului de machine learning sustinut de Andrew Ng [28]

## 4.3 SQL injection

Atacurile de tipul SQL injection sunt realizate prin injectarea de cod executabil intr-o baza de date.

Procesul de interactionare cu o baza de date presupune realizarea de interogari asupra acesteia. In formularea acestor interogari, utilizatorul trebuie sa prezinte interpretorului, sub forma de siruri de caractere, numele tabelelor interogate sau valorile unor capuri specifice din acestea. Aceste siruri de caractere sunt delimitate folosind caracterul " sau '. Atacurile de tipul SQL injection exploateaza folosirea acestor delimitatori de siruri de caractere, trimitand siruri de caractere eronate intentionat catre baza de date. Un utilizator rau intentionat poate sa furnizeze astfel de siruri de caractere catre o baza de date prin intermediul oricarui procesator de continut disponibil unui client al unei aplicatii ce comunica cu o baza de date. Aceste siruri de caractere delimiteaza prematur valoarea care este folosita in interogare, introducand dupa aceasta o serie de caractere pe care interpretorul le va trata ca si cod executabil, oferindui astfel utilizatorului sa execute operatiuni asupra bazei de date la care nu ar avea accesul in mod normal. Aceste operatiuni pot sa reprezinte alterarea bazei de date sau obtinerea de date confidentiale.



The image shows a 'Sign in' form with two input fields: 'Email' and 'Password'. The 'Email' field contains the text 'test@example.com' OR 1 = 1 --'. The 'Password' field contains several dots, indicating it is masked. Below the fields are two buttons: a blue 'Sign in' button and a checkbox labeled 'Stay signed in'.

Figura 4.3: Exemplu de atac realizat prin SQL injection.

Figura 4.3 prezinta o tentativa de atac prin SQL injection in care in campul de validare a email-ului se incearca injectarea de cod ce va fi executat in interogarea de validare a credentialelor. Prin prezenta caracterului ' se escapeaza tot textul urmat dupa acesta ca fiind cod si nu un string ce face parte din campul de email. Operatia logica "OR 1=1" va determina interpretorul sa returneze adevarat(valid) pentru orice adresa de email introdusa inaintea caractrului '.

## 4.4 Obtinerea adreselor retelei Tor

Reteaua Tor reprezinta un softwrae gratis de anonimizare a traficului pe internet. Numele este de fapt un actonim pentru "The Onion Router"(router-ul ceapa) care sugereaza modul de functionarea al acestuia, ficare nod din retea adugand un strat extra de securitate celor precedente. Modul de functionare al retelei se bazeaza pe rutarea traficului prin cat mai multe noduri pentru a anonimiza si a face cat mai greu de urmarit traficul unei anumite persoane. Aceste noduri prin care traficul este directionat sunt sustinute gratis de catre voluntari/utilizatori de Tor din intreaga lume.

Pentru criptarea traficului retea Tor foloseste criptarea la nivelul aplicatiei in cea ce priveste structura retelelor de calculatoare(modelul OSI sau TCP/IP). Datele tras-mise sunt criptate, incluzand destinatia, cu exceptia nodului urmator, astfel creandu-se structura de "ceapa" asupra unui pachet. Selectia nodurilor prin care se face rutarea pachetelor este aleasa random. Fiecare pachet decripteaza un "strat", afand nodul urmator pentru pachetul respectiv, nodul final decriptand datele initiale si realizand transmisia catre destinatie, fara sa ii comunice sursa pachetului. Intrucat in comunicarea pachetelor, pe parcursul rutelor parcurse, se cunoaste in permanenta doar nodul urmator, acest lucru impiedica monitorizarea traficului intre sursa si destinatie.

Cu toate ca retea tor ofera anonimitate de partea clientului, acest lucru nu se

realizeaza si faca de ea. Reteaua nu se ascunde fata de serviciile accesate prin intermediul ei. Astfel un site anume poate sa detecteze daca un anumit client il acceseaza folosind reteaua Tor sau nu.

Intrucat reteaua Tor nu isi ascunde adresele ip folosite de catre aceasta, indentificarea lor si procurarea de date despre acestea este destul de usoara. In proiectul propus s-a folosit un serviciu care furnizeaza gratuit astfel de date [26](adresa ip, uptime etc.) si s-au folosit algoritmi proprii pentru procesarea acestor date, eliminand astfel necorectitudinea dintre utilizatorii retelei.

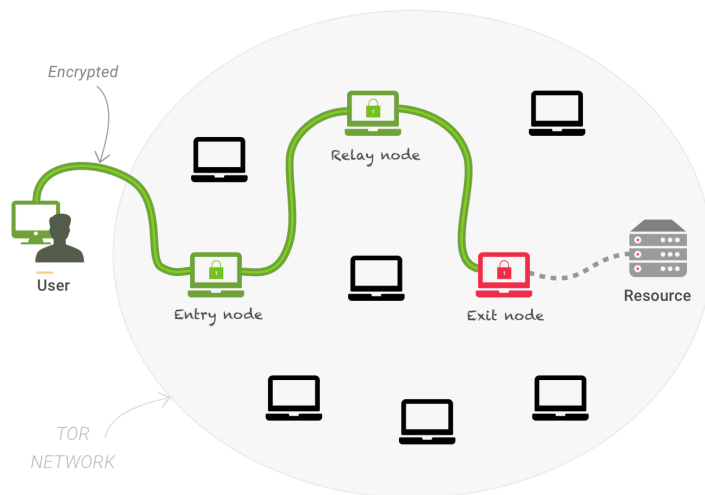


Figura 4.4: Exemplu de trafic realizat prin reteaua Tor.

Figura 4.4 prezinta principalele elemente folosite la rutarea treficului de la client la destinatie prin intermediul retelei Tor.

## 4.5 Sistem de prevenire a intruziunilor

Conform scurtei descrieri prezentate in capitolul anterior, un sistem de prevenire a intruziunilor are rolul de a filtra traficul dintre clientii unui server si serverul propiu zis.

Acest sistem functioneaza liniar, adica este plasat direct intre server si clienti acestuia. In cazul proiectului propus, componenta de baza pentru interceptarea traficului este realizata prin implementarea unui reverse proxy, oferind astfel caracteristica de interceptare si decriptare a traficului, ce permite analiza acestuia, dar si avantajele specifice utilizarii unui reverse proxy.

Pentru filtrarea traficului, un sistem de prevenire a intruziunilor implementeaza anumiti senzori care au rolul sa inspecteze tot traficul ce trece prin sistem, realizand aceasta inspectie in timp real. Datorita acestei verificari, orice pachet considerat malitios este oprit

din a ajunge la serverul destinatie. In proiectul propus, implementarea acestor senzori este realizata in doua moduri. In cazul validarii adreselor ip impotriva utilizatorilor de Tor se foloseste o lista de ip-uri ce contine adrese frecvent utilizate de retea Tor. In interiorul reverse proxy-ului, in momentul crearii unei noi conexiuni, acesta verifica ca adresa ip ce solicita conectarea la server sa nu fie continuta de lista mentionata. In cazul detectiei impotriva atacurilor de SQL injection, senzorul este implementat prin utilizeaza un model de support vector machine. In interiorul reverse proxy-ului in momentul interceptarii unui request venit din exterior catre retea internă, acesta verifica daca requestul poate fi clasificat ca si tentativa de atac, in caz afirmativ blocand trecerea acestuia mai departe catre server.



Figura 4.5: Integrarea unui sistem de prevenire a intruziunilor intr-o retea.

Figura 4.5 prezinta arhitectura unei retele interne ce integreaza un sistem de prevenire a intruziunilor pentru protejarea acesteia.

Un sistem de prevenire a intruziunilor poate sa efectueze oricare din urmatoarele actiuni in momentul detectarii unui eveniment malitios [29]:

- Sa intrerupa sesiunea dintre client si server, in cazul in care clientul desfasoara sau incearca sa desfasoare activitati malitioase in retea protejata de sistem. Acest lucru se poate realiza prin blocarea anumitor credentiale asociate cu utilizatorul respectiv sau prin blocarea adresei ip a acestuia.
- In conditiile in care un sistem de prevenire a intruziunilor detecteaza clasifica o activitate ca fiind malitioasa, acesta poate sa ia masuri automat pentru a preveni un astfel de atac pe viitor(ex: in momentul detectarii unei tentative de atac prin SQL injection, sistemul de prevenire a intruziunilor poate sa blocheze in mod automat adresa ip a utilizatorului ce incerca sa faca atacul, nepermitandu-i acestuia sa se mai conecteze la serverul destinatie pentru un anumit interval de timp sau pana la interventia unui administrator).
- O alata abordare posibila in momentul declansarii unui eveniment malitios este alterarea traficului astfel incat sa elimine continutul malitios din acesta. Pentru realizarea acestui lucru, un sistem de prevenire a intruziunilor poate sa stearga atasamente

infectate din interiorul unui mail, sa altereze continutul uni pachet sau sa omita transmiterea mai departe a unor pachete.



# Capitolul 5

## Analiză și proiectare

În acest capitol este descris design-ul proiectului și cuprinde: cerințele sistemului, specificațiile cazurilor de utilizare, arhitectura sistemului, comportamentul sistemului, datele utilizate de sistem, dependențele sistemului și algoritmi esențiali și metodele folosite. Descrierea acestora se realizează prin asocierea cu diagramelor aferente.

### 5.1 Cerințele sistemului

Sistemul propus *Reverse proxy pentru prevenirea utilizatorilor de Tor și a atacurilor SQLI* reprezintă un software gratuit care oferă clientului atât caracteristici specifice unui reverse proxy, cât și modalități de protecție asemănătoare unui sistem de prevenire a intruziunilor. Sistemul este ușor de instalat și de utilizat, chiar și de către utilizatorii neexperimentați, oferindu-le acestora o interfață clară și sugestivă, ce maschează logica complicată din spate. În spate, sistemul oferă un reverse proxy ce interceptează tot traficul destinat către un anumit server, de pe una sau mai multe interfețe. În procesul de interceptare, acesta implementează și câteva modalități de prevenire a unor tentative de intruziune. Sistemul blochează toate ip-urile utilizate frecvent de rețeaua Tor în ultima lună și atacurile de SQL injection cu o acuratețe de 90%.

Sistemul trebuie să îndeplinească următoarele cerințe **functionale**:

1. Să realizeze conexiunea la un server HTTP/HTTPS și să redirectioneze traficul primit către acesta.
2. Să intercepteze traficul venit pe o anumită interfață și port prestabilit.
3. Să prelucereze request-urile primite de la clienți într-un format specific clasificatorului de SQL injection.
4. Să nu redirectioneze requesturile clasificate ca și SQL injection.
5. Să blocheze conectarea clientilor ce folosesc ip-uri clasificate ca ip-uri de Tor.

6. Sa permita utilizatorului sa editez si sa vizualizeze lista ip-urilor de Tor.
7. Sa prezinte in interfata grafica toate interventiile rezlizate asupra traficului(blocari de conexiuni sau de request-uri).
8. Sa permita utilizatorului sa configureze modul de operare al sistemului.

Sistemul trebuie, de asemenea, să aibă următoarele caracteristici **non-funcționale**:

1. Sa fie usor de instalat si de folosit pentru orice utilizator, oricat de neexperimentat.
2. Sa poata intercepta traficul de pe orice/oricate interfete disponibile.
3. Sa poata rula pe orice sistem de operare Windows cu Python2 instalat.
4. Sa aiba o rata de blocare de 100% a ip-urilor de pe lista neagra, iar in cazul detectiei de SQL injection sa nu aiba detectii false pozitive mai mari 2-3% si o acuratete generala de peste 90%



## 5.2 Specificatiile cazurilor de utilizare

### 5.2.1 Actori, stakeholders si interese

Principalii actori si stakeholder-i sunt administratorii de servere, respectiv de baze de date. In implementarea sistemului propus se urmareste satisfacerea nevoilor acestor persoane, oferindule un plus de securitate asupra datelor ce sunt accesate de catre clienti, respectiv impotriva clientilor rau intentionati. Interesele acestor comunitati de utilizatori sunt urmarite pentru a livra un produs care sa ofere aceste protectii intr-o maniera cat de prietenoasa pentru utilizator si cat mai eficienta.

### 5.2.2 Basic flow

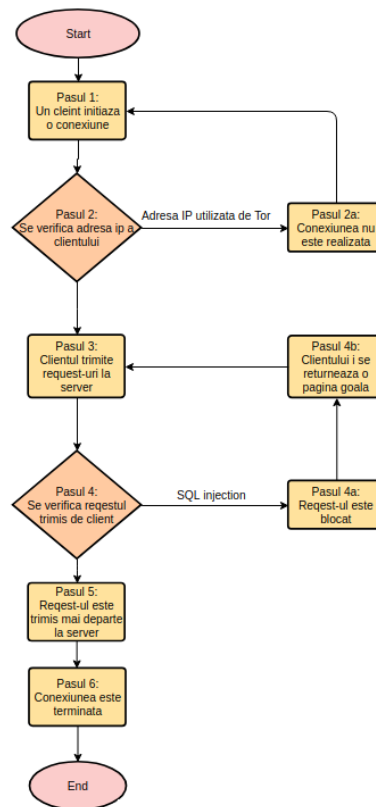


Figura 5.1: Basic flow pentru evenimente

Figura 5.1 prezinta cum arata basic flow-ul pentru evenimentele din sistemul propus.

Evenimentele sistemului:

1. **Start:** aceste cazuri de utilizare sunt initiate in momentul in care sistemul este plasat intre un server si utilizatorii acestuia.
2. **Pasul 1:** un client doreste si initializeze o conexiune la server.
3. **Pasul 2:** adresa ip a clientului este verificata ca acesta sa nu fie un utilizator de Tor.
4. **Pasul 3:** clientul comunica cu serverul prin request-uri individuale.
5. **Pasul 4:** request-urile sunt verificate ca acestea sa nu fie atacuri de SQL injection.
6. **Pasul 5:** request-urile sunt trimise mai departe catre server.
7. **Pasul 6:** conexiunea este terminata(de catre client sau server).
8. **End:** la sfarsitul unui sir de evenimente, utilizatorul sistemului poate sa vada daca clientul ce a initializat evenimentele a realizat actiuni considerate ca malitioase.

### 5.2.3 Alternative flow

Evenimentele alternative in cazurile de utilizare sunt urmatoarele:

1. **Pasul 2a:** in cazul in care un utilizator cu adresa ip de Tor incearca sa realizeze conexiunea la server aceasta este refuzata.
2. **Pasul 4a:** in cazul in care un utilizator incearca sa trimita la server un request ce reprezinta o tentativa de atac SQL injection, acesta este blocat.
3. **Pasul 4b:** pentru request-urile blocate ca si SQL injection clientului i se returneaza o pagina goala.

## 5.3 Arhitectura sistemului si modulele principale

În conformitate cu cerințele sistemului și specificațiile cazurilor de utilizare, sistemul propus este alcătuit din module specifice care să trateze în mod individual problemele majore ridicate de sistem, printr-o implementare modulară, oferind astfel ușurința implementării de noi funcționalități.

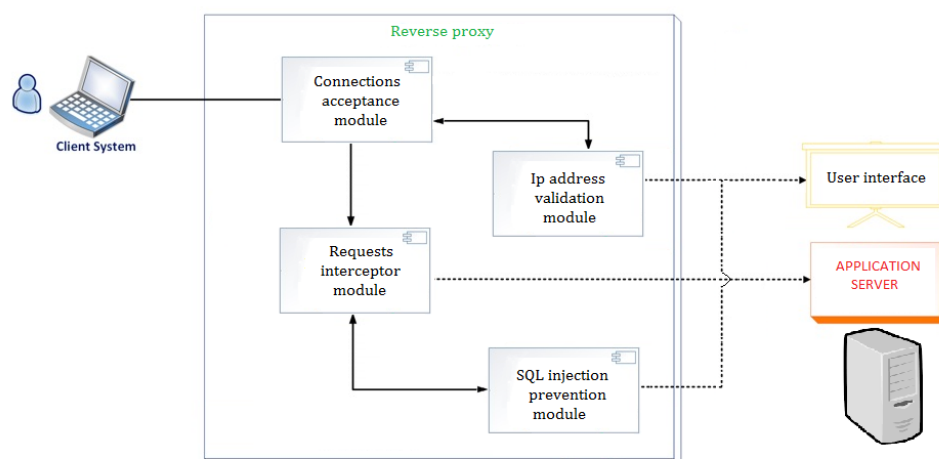


Figura 5.2: Principalele module ale sistemului propus

Figura 5.2 prezintă care sunt principalele module ale sistemului propus, precum și interacțiunea dintre acestea.

Principală și teoretic singura componentă a sistemului este reprezentată de cea de **reverse proxy**, însă pentru a obține rezultatele dorite și pentru a încorpora funcționalitățile de protecție în această, ea a fost împărțită în cele 4 submodule: **connections acceptance module** și **request interceptor module**, ce reprezintă componentele ce suportă logica unui reverse proxy și **ip address validation module** și **SQL injection prevention module**, ce reprezintă modulele cu care interacționează cele ce implementează structura de reverse proxy pentru a oferi funcționalitățile de securitate.

### Client system și application server.

Aceste două componente reprezintă componentele clasice între care vin plasat sistemul propus. **Client system** este reprezentat de orice client dorește să acceseze baza de date/parte de server a unei aplicații. **Application server** reprezintă serverul aplicație la care se pot conecta clienții pentru a avea acces la un anumit conținut. Sistemul propus are rolul de intermediere între cele două tipuri de componente, prevenind astfel eventuale tentative de exploatare a unor vulnerabilități din partea clientului către server.

### **Connections acceptance module.**

Acest modul este constituit din componentele oferite de biblioteca open source **twisted**, fiind modificate ulterior pentru a permite integrarea modulului **Ip address validation module**. Modulul are rolul de a crea un socket pe sistemul pe care rulează acesta, care să asculte pentru posibile conexiuni. În cazul în care un client încearcă să inițieze o nouă conexiune acesta transmite adresa IP a clientului către modulul **Ip address validation module**, iar în funcție de răspunsul acestuia conexiunea acestuia va fi realizată sau refuzată.

### **Request interceptor module.**

Asemeni modulului anterior, **Connections acceptance module**, acest modul este construit peste codul oferit de biblioteca open source **twisted**. Biblioteca oferă suport pentru interceptarea request-urilor dintre un client și server, implementându-se o logică suplimentară ce încorporează modulul de detectie a atacurilor SQLI( **SQL injection prevention module**). În acest modul se interpretează natura URL-urilor trimise de către client către server. Interpretarea constă în analiza URI-ului unui request de către modulul de prevenire a atacurilor SQLI, iar în funcție de rezultatul acestuia returnând răspunsuri corespunzătoare atât clientului cât și serverului(în caz de tentativă de atac, la server nu ajunge nimic).

### **Ip address validation module.**

### **SQL injection prevention module.**

# Capitolul 6

## Implementation Details

Ponderea acestui capitol relativ la întreaga lucrare este de 20-30%.  
Conține detalii de implementare:

- organizarea codului sursă, organizarea logică a codului (module, ierarhii de clase)
- descrierea claselor, funcțiilor, API-urilor importante ale aplicației
- descrierea la nivel de implementare a algoritmilor principali
- descrierea părților mai dificile
- alte detalii de implementare relevante, specifice fiecărei aplicații

Descrierea implementării trebuie să reflecte modul în care ea corespunde (se mapează) design-ului.

Nu se vor da detalii irelevante. Descrierea codului trebuie gândită ca un ghid de parcurgere a codului sursă de către cineva care vrea să continue proiectul vostru.

Exemplu de cod:

```
# include <stdio.h>

int main (int argc, char **argv)
{
    int i;

    for (i=0; i<argc; i++)
        printf("argv[%d] = %s\n", i, argv[i]);

    return 0;
}
```



# Capitolul 7

## Tests and Results

Ponderele acestui capitol relativ la întreaga lucrare este de 5-10%.

Aici sunt prezentate metodele de validare a soluțiilor/sistemului descris în capitolele anterioare, scenariile de testare a corectitudinii funcționale, a utilizabilității, performanței etc.

Rezultatele testelor experimentale necesită, în general interpretări (dacă rezultatele obținute corespund așteptărilor, intuițiilor cititorului, de ce apar variații/exceptii etc.) și comparații cu rezultatele altor metode similare.

Sistemele de testare și testele propriu-zise trebuie descrise detaliat astfel încât să poată fi reproduse și de alții care poate vor să-și compare soluțiile lor cu a voastră (eventual, codul testelor poate fi pus în anexe). Dacă se poate alegeți pentru evaluarea sistemului vostru benchmark-uri (pachete de testare) dedicate, astfel încât comparația cu alte sisteme să poată fi făcută mai ușor. În plus, astfel de teste sunt mult mai complete și mai realiste decât cele dezvoltate de voi. Oricum, încercați ca testele efectuate să nu fie triviale, ci să acopere scenarii cât mai reale, mai complexe și mai relevante ale funcționării sistemului vostru.

### 7.1 Functional Tests

### 7.2 Performance Tests





# Capitolul 8

## User Manual

Describe pașii de instalare și rulare a aplicației. Dacă dezvoltarea aplicației s-a bazat sau a presupus instalarea și configurarea unei infrastructuri (complexe), descrieți detaliat pașii pe care i-ați urmat (referințele utilizate) și mai ales abaterile voite sau necesare de la documentațiile referite. Încercați ca cineva care vă continuă tema să nu mai fie nevoit să mai piardă timp inutil cu pregătirea mediului de lucru și să poată trece cât mai repede la abordarea temei propriu-zise a proiectului.

Indicați, de asemenea, explicit versiunile aplicațiilor, bibliotecilor folosite și salvați o copie a acestora pe CD-ul atașat lucrării. E posibil ca aplicația voastră să nu mai funcționeze la fel pe alte versiuni și e bine de știut acest lucru și, în același timp, e bine ca mediul descris de voi să poată fi reprodus ulterior.

Se întinde pe aproximativ 2-3 pagini.



# Capitolul 9

## Conclusions

Cuprinde:

- un rezumat al contribuțiilor aduse: ce s-a realizat, relativ la ce s-a propus, în ce constă experiența acumulată, care au fost punctele dificile atinse și rezolvată, recomandări pentru alții care abordează tema, la ce este bun ce s-a obținut etc.
- a analiză critică a rezultatelor obținute: avantaje, dezavantaje, limitări
- o descriere a posibilelor dezvoltări și îmbunătățiri ulterioare

Poate fi organizat pe secțiuni, dacă se dorește.  
Se întinde pe aproximativ 1-2 pagini.



# Bibliografie

- [1] J. C. Villanueva, “Top 8 Benefits of a Reverse Proxy,” <https://www.jscape.com/blog/bid/87841/Top-8-Benefits-of-a-Reverse-Proxy>, Aug 2012.
- [2] “OWASP Top 10 Application Security Risks - 2017,” [https://www.owasp.org/index.php/Top\\_10-2017\\_Top\\_10](https://www.owasp.org/index.php/Top_10-2017_Top_10), Last accessed: September 30st, 2018.
- [3] J. V. William G.J. Halfond and A. Orso, “A Classification of SQL Injection Attacks and Countermeasures,” 2006, College of Computing Georgia Institute of Technology, Proceedings of the IEEE International Symposium on Secure Software Engineering.
- [4] M. Prince, “The Trouble with Tor,” <https://blog.cloudflare.com/the-trouble-with-tor/>, Mar 2016.
- [5] R. Bassett, C. Urrutia, and N. Ierace, “Intrusion prevention systems,” <https://dl.acm.org/citation.cfm?id=1071927>, June 2005, ACM New York, NY, USA .
- [6] “What is an intrusion prevention system?” <https://www.paloaltonetworks.com/cyberpedia/what-is-an-intrusion-prevention-system-ips>.
- [7] J. Snyder, “Information Security,” <https://searchsecurity.techtarget.com/Do-you-need-an-IDS-or-IPS-or-both>.
- [8] M. Rouse, “Network management and monitoring: The evolution of network control,” <https://searchnetworking.techtarget.com/definition/network-analyzer>.
- [9] B. Hendricks, “Intrusion Prevention System (IPS): Definition & Types,” <https://study.com/academy/lesson/intrusion-prevention-system-ips-definition-types.html>.
- [10] C. Paquet, “Implementing cisco ios network security (iins): (ccna security exam 640-553) (authorized self-study guide),” Apr 2009, by Cisco Press.
- [11] K. Rajesh, “An overview of ips – intrusion prevention system and types of network threats,” <https://www.excitingip.com/626/an-overview-of-ips-intrusion-prevention-system-and-types-of-network-threats/>, October 2009.

- [12] E. H. Cheon, Z. Huang, and Y. S. Lee, "Preventing SQL Injection Attack Based on Machine Learning," 2013, Department of Computer Engineering, Woosuk University and Department of Computer Information Engineering, Kunsan National University .
- [13] F. Valeur, D. Mutz, and G. Vigna, "A Learning-Based Approach to the Detection of SQL Attacks," 2005, Reliable Software Group, Department of Computer ScienceUniversity of CaliforniaSanta Barbara .
- [14] P. Winter and S. Lindskog, "How the Great Firewall of China is Blocking Tor," 2012, Karlstad University .
- [15] R. Singh, R. Nithyanand<sup>2</sup>, S. Afroz, P. Pearce, M. C. Tschantz, P. Gill<sup>1</sup>, and V. Paxson, "Characterizing the Nature and Dynamics of Tor Exit Blocking," 2017, University of Massachusetts – Amherst, Stony Brook University, University of California – Berkeley, International Computer Science Institute .
- [16] "Framework pentru programarea retelelor scris in python." [Online]. Available: <https://twistedmatrix.com/documents/8.1.0/api/twisted.html>
- [17] "Librarie pentru procesarea de cod html/xml scris in python." [Online]. Available: <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>
- [18] "Software integrat cu suport pentru svm." [Online]. Available: <https://www.csie.ntu.edu.tw/~cjlin/libsvm/>
- [19] "Documentation for python 3.7.1rc1." [Online]. Available: <https://docs.python.org/3/>
- [20] "Documentation for python 2.7.15." [Online]. Available: <https://docs.python.org/2.7/>
- [21] Tim Gray, "Python 2 to python 3: why, and how hard can it be?" [Online]. Available: <https://optimalbi.com/blog/2018/01/19/python-2-to-python-3-why-and-how-hard-can-it-be/>
- [22] "Tabela cu toate caracterele ascii si codurile lor specifice in diferite baze." [Online]. Available: <https://www.asciitable.com/>
- [23] "Caracterele speciale ce pot fi folosite intr-o interogare sql." [Online]. Available: [https://docs.oracle.com/cd/A97630\\_01/text.920/a96518/cqspcl.htm](https://docs.oracle.com/cd/A97630_01/text.920/a96518/cqspcl.htm)
- [24] "Cuvinte rezervate specifice limbajului sql." [Online]. Available: <https://docs.microsoft.com/en-us/sql/t-sql/language-elements/reserved-keywords-transact-sql?view=sql-server-2017>
- [25] "A practical guide to support vector classification." [Online]. Available: <https://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf>

- [26] “Statusul nodurilor utilizate de tor.” [Online]. Available: <https://torstatus.blutmagie.de/index.php>
- [27] by NGINX Page, “What is a reverse proxy server?” [Online]. Available: <https://www.nginx.com/resources/glossary/reverse-proxy-server/>
- [28] Andrew Yan-Tak Ng, “Machine learning by stanford university.” [Online]. Available: <https://www.coursera.org/learn/machine-learning/home/welcome>
- [29] by Firewalls.com Page, “How intrusion prevention systems (ips) work in firewall.” [Online]. Available: <https://community.spiceworks.com/topic/362007-how-intrusion-prevention-systems-ips-work-in-firewall>





**Anexa A**

**Diverse anexe**



## Anexa B

### Demonstrații matematice detaliate (dacă există)



## Anexa C

### Pseudo-cod sau cod (dacă există)

```
/** Maps are easy to use in Scala. */
object Maps {
  val colors = Map("red" -> 0xFF0000,
                   "turquoise" -> 0x00FFFF,
                   "black" -> 0x000000,
                   "orange" -> 0xFF8040,
                   "brown" -> 0x804000)

  def main(args: Array[String]) {
    for (name <- args) println(
      colors.get(name) match {
        case Some(code) =>
          name + " has code: " + code
        case None =>
          "Unknown color: " + name
      }
    )
  }
}
```



Anexa D

Articole publicate