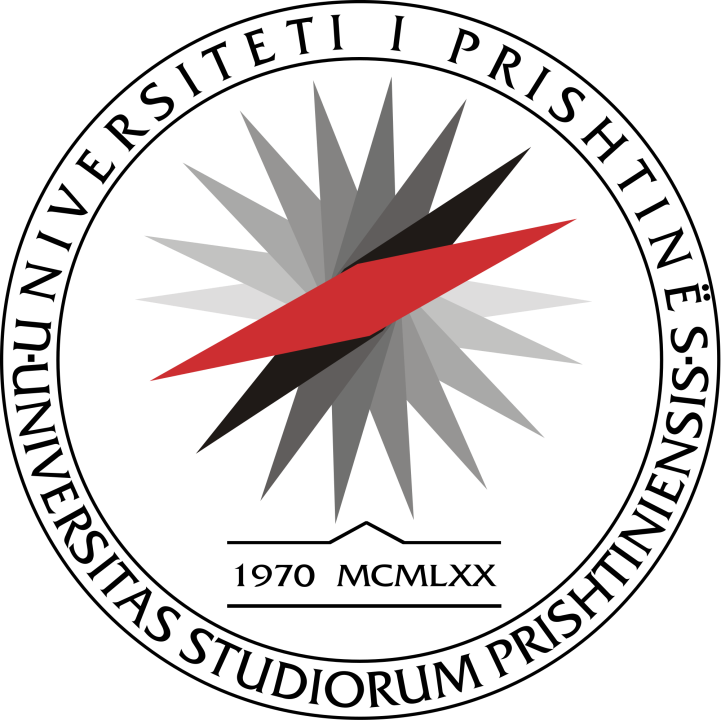
UNIVERSITETI I PRISHTINËS  
 FAKULTETI I INXHINERISË ELEKTRIKE DHE KOMPJUTERIKE



Detyra 2: Kali Linux- Sqlmap

Studentët:

Profesori: Prof. Blerim Rexha Ardit Abazi

Donart Pllashniku

Qlirim Hoti

Prishtinë, 2020

**Përmbajtja**

[**Abstrakt** 3](#_Toc59567372)

[**1.** **Përshkrimi i veglës Sqlmap** 4](#_Toc59567373)

[**1.1. Sqlmap** 4](#_Toc59567374)

[**1.2 Mbrojtja** 4](#_Toc59567375)

[**2. Teknikat e përdorimit të Sqlmap** 6](#_Toc59567376)

[**2.1 SQL Injektimi i bazuar në Boolean** 6](#_Toc59567377)

[**2.2 SQL Injektimi i bazuar në kohë** 6](#_Toc59567378)

[**2.3 SQL Injektimi i bazuar në gabime** 6](#_Toc59567379)

[**2.4 SQL Injektimi i bazuar në Union – Query** 7](#_Toc59567380)

[**2.5 SQL Injektimi - Stacked queries and out-of-band** 7](#_Toc59567381)

[**3. Karakteristikat** 7](#_Toc59567382)

[**3.1. Karakteristikat gjenerike** 7](#_Toc59567383)

[**3.2. Karakteristikat e numërimit dhe fingerprint-it** 8](#_Toc59567384)

[**3.3. Takeover karakteristikat** 9](#_Toc59567385)

[**4. Injeksionet** 9](#_Toc59567386)

[**4.1 Komanda -p** 9](#_Toc59567387)

[**4.2. Komanda: --dbms** 9](#_Toc59567388)

[**4.3 Komanda --os** 10](#_Toc59567389)

[**5. Përdorimi i Sqlmap** 11](#_Toc59567390)

[**5.1 Komanda --dbs** 11](#_Toc59567391)

[**5.2 Komanda --D** 12](#_Toc59567392)

[**5.3 Komanda –T dhe --columns** 12](#_Toc59567393)

[**5.4 Komanda --dump** 13](#_Toc59567394)

[**5.5 Komanda -current-user** 13](#_Toc59567395)

[**5.6 Komanda –current-db** 14](#_Toc59567396)

[**5.7 Komandat –privileges, --roles, --is—dba dhe --hostname** 14](#_Toc59567397)

[**6. Konkluzion** 16](#_Toc59567398)

[**7. Biblografia** 17](#_Toc59567399)

# **Abstrakt**

Organizatat anë e mban botës marrin sulme të ndryshme kibernetike çdo ditë. Fusha e sigurisë kibernetike vazhdon të bëhet jetike për botën digjitale për arsye se shumë e shumë të dhëna vazhdojnë të futën në botën digjitale e që tashmë të mos themi se shumica e të dhënave gjendën në botën digjitale. Të dhënat në përgjithësi e sidomos të dhëna personale janë cak dhe qëllim i pothuajse shumicës së sulmeve kibernetike që bëhen gjithandej botës. Andaj është punuar mjaftë shumë dhe vazhdohet të punohet në mbrojtje nga këto sulme. Megjithatë me rritjen e sulmeve janë zhvilluar edhe shumë vegla dhe softuere të ndryshëme për parandalim të tyre. Ka shumë mjete softuerësh që mund të përdoren për të testuar dobësitë e njëra prej llojeve të sulmeve që do të diskutohen në këtë punim është injektimet SQL në një aplikacion në internet. Vegla që do të analizohet në këtë punim është Sqlmap që është vegla më e preferuar për përdoruesit e Linux për detektim, testim për parandalim të këtyre sumleve.

# **Përshkrimi i veglës Sqlmap**

## **Sqlmap**

Sqlmap është njëra nga tools-at(mjetet) open source të njohura për testime automatike në procesin e zbulimit dhe shfrytëzimit të dobësive apo defekteve që mund të ketë apo shkaktohen nga të ashtuquajtuara injektime SQL tek ndonjë Databazë apo një sistemi relativ të menaxhimit të bazës së të dhënave.

Sqlmap është një vegël e fuqishme zbulimi e testimi që vie me një gamë të gjerë çelsash për çasje në marrjen e të dhënave nga baza të ndryshme të të dhënave e deri te hyrja në sisteme dhe ekzekutim të komandave në një sistem operativ. Automatizimi është një nga cilësit më të mira të kësaj vegle.

Sqlmap është zhvilluar nga 2 zhvillues Bernardo Damele A. G. dhe Miroslav Stampar në gjuhën programuese Python. Kjo e bënë këtë vegël të pavarur nga sistemi operativ. Që nga maji i vitit 2019 kjo vegël përkrahet edhe në versionin 3 të python pasi deri në këtë vitë ka qenë i përkrahur vetëm në versionin 2 më specifikisht në një kohë vetëm në versionin 2.6 e më lartë.

Disa nga veçoritë kryesore të Sqlmap mund të thuhet se është mbështetja e gjashtë teknikave të SQL injektimeve (Sulmi SQL injection bëhet me insertimin e SQL query përmes futjes së të dhënave prej klientit në aplikacion. Injektimi i suksesshëm i SQL mund të bëj të mundur leximin e të dhënave prej databazës, modifikimin e të dhënave apo edhe fshirjen e tyre nga databaza): boolean-based blind, time-based blind, error-based, UNION query-based, stacked queries dhe out-of-band. Veçori tjetër e rëndësishme është që lejon apo jep mbështetje përdoruesit në shkarkimin dhe ngarkimin e skedarëve të sistemit në llojet e bazave të të dhëna si: Microsoft SQL Server , MySQL ose PostgreSQL. [1] [5]

## **1.2 Mbrojtja**

Sulmi SQL injection bëhet me insertimin e SQL query përmes futjes së të dhënave prej klientit në aplikacion. Shembuj të ndryshëm të tillë kemi parë mjaft shumë dhe sulme të tilla ndodhin çdo ditë. Kjo rritje vije edhe si pasoj e rritjes së interaksionit me përdoruesit gjegjësisht vizitorët. Llojet e dëmeve që mund të shkaktohen janë:

* Manipulimi i të dhënave në databazë
* Vjedhja e informacioneve
* Humbje financiare( Në biznese si të llojit të bankave)
* Dëmtim i markës së biznesit (firmës)
* Shitja e informatave dhe të dhënave tjera të konsumatorëve etj.

Parandalimet dhe kontrollimet mund të organizohen duke analizuar respektivisht kontrolluar çdo parametër hyrës. Forma të ndryshme të teknikave të kontrollave me AI duke bërë ndarjen e parametrav hyrës normal dhe ato që mund të jenë të dëmshme. Parametra hyërs - input-at të mos lejojn hyrje të simboleve që përdoren në query dhe të kenë gjatësi të caktuar. Pra forma të ndryshme mirëpo të cilat pa bërë filtrimin e e të gjitha inputeve nuk mund të parandalohen.

Poashtu lajmërimet për gabime të ndryshme të kenë njoftime të përgatitura dhe të mos jenë njoftime të zakonshme të SQL.

Vegla Sqlmap është vegla që mund të na ndihmoj të testojm dhe me këtë rast edhe të parandalojm sulme të mundshme duke na treguar dobsitë e web aplikacionit. Mirëpo përdorimi i Sqlmap për sulmimin e objektivave pa pëlqimin paraprak të ndërsjellë është i paligjshëm. Përgjegjësia është e përdoruesit dhe duhet t'i bindet të gjitha ligjeve të zbatueshme lokale, shtetërore dhe federale. Zhvilluesit nuk marrin asnjë përgjegjësi dhe nuk janë përgjegjës për çdo keqpërdorim ose dëmtim të shkaktuar nga ky program. Pasi që kjo vegël mund të përdoret edhe nga sulmuesit prandaj duhet pasur kujdes.

# **Teknikat e përdorimit të Sqlmap**

Sqlmap është në gjendje të zbulojë dhe shfrytëzojë dhe testoj disa lloje të ndryshme të SQL injektimeve e disa prej tyre do të shtjellohen në vazhdim:

## **2.1 SQL Injektimi i bazuar në Boolean**

SQL Injektimi me bazë Boolean është një teknikë e cila mbështetet në dërgimin e një SQL query në bazën e të dhënave. Ky lloj i injektimit detyron aplikacionin të kthejë përgjigje (rezultat) tjetër në varësi të pyetjes. Në varësi të rezultatit boolean (TRUE ose FALSE), përmbajtja brenda përgjigjes HTTP do të ndryshojë ose do të mbetet e njëjtë. Rezultati mund të lejoj një sulmues të gjykojë nëse ngarkesa e përdorur kthehet True ose False edhe pse nuk janë marrur të dhëna nga baza e të dhënave. Sapo të gjendet një dobësi apo backdoor, Sqlmap në mënyrë automatike zëvendëson ose shton SQL query dhe pret përgjigjiet. Këto përgjigjie i krahason dhe i ndan varësisht nga madhësia. Ky sulm është më i ngadalshëm dhe mund të sheh edhe më shumë gjëra në Databazë që mund të jenë të dëmshme për përdoruesit. [3]

## **2.2 SQL Injektimi i bazuar në kohë**

Time-based SQL injektimi është injektim i llojit inferencial ose sulm me injektim të verbër. Është një lloj sulmi në të cilin nuk transferohen të dhëna midis sulmuesit dhe bazës së të dhënave dhe sulmuesi nuk do të jetë në gjendje të marrë rezultate aq lehtë dhe kjo është edhe njëra nga arsyejet pse quhet edhe një sulm me injeksion të verbër. Një sulmues mund të rindërtojë dhe të krijojë një strukturë të re të bazës së të dhënave brenda bazës së të dhënave. Në një sulm të bazuar në kohë, një sulmues dërgon një komandë SQL te serveri me kod për të detyruar një vonesë në ekzekutimin e query-it në mënyrë që nga vonesa e përgjigjes të dihet nëse rezultati është True apo False. Ky lloj i sulmit është nga lloji i sulmeve shumë të ngadalshëm që web aplikacionin mund ta lë në një rrezik të lartë. [8]

## **2.3 SQL Injektimi i bazuar në gabime**

Error-Based Injektimi mbështetet në mesazhet e gabimit të hedhura nga serveri i bazës së të dhënave për të marrë informacion në lidhje me strukturën e bazës së të dhënave. Në shumë raste gabimi i gjeneruar nga baza e të dhënave është e mjaftueshme që sulmuesi të kuptojë tërësisht bazën e të dhënave. Prandaj çaktivizimi i lajmërimeve të drejtpërdrejta për përdorues ose vizitorë që nuk kanë çasje në të dhëna në web aplikacion.

## **2.4 SQL Injektimi i bazuar në Union – Query**

SQL Injektimi i bazuar në union lejon një sulmues të nxjerrë informacion nga baza e të dhënave duke zgjeruar rezultatet e kthyera nga pyetja origjinale. Operatori i Unionit mund të përdoret vetëm nëse query-të origjinale / të reja kanë të njëjtën strukturë (numri dhe lloji i të dhënave të kolonave). Sulmuesi në këtë rast mund të nxjerrë informatat shumë shpejtë për arsye të mundsisë të kombinimit të shumë komandave në një. [10]

## **2.5 SQL Injektimi - Stacked queries and out-of-band**

**Stacked queries -** Një pikëpresje (;) përdoret për të përfunduar një statement në SQL. Duke fshirë radhën origjinale dhe duke shtuar një të ri sulmuesi mund të ndryshoj të dhënat në bazën e të dhënave. Një pikëpresje lejon sulmuesin të ekzekutojë statements të shumta në bazën e të dhënave që ka ngjashmëri në sulmin e bazuar në Union. Injeksioni i bazuar në Union mund të bëhet vetëm duke përdorur statments SELECT, kurse te Stacked queries mund të bëhet me çfarëdo statements. Pra ky lloj i sulmit gjeneron një cenueshmëri të madhe të bazës së të dhënave.

**Out-of-band-** Nuk është shumë i përdorur e një ndër arsyet kryesore pse nuk përdoret është sepse varet nga tiparet që aktivizohen në serverin e bazës së të dhënave. Rastet kur përdoret ky lloj i sulmit është kur një sulmues nuk është në gjendje të përdorë të njëjtin kanal për të filluar sulmin dhe për të mbledhur rezultate. Kjo teknik i ofron një sulmuesi një alternativë ndaj teknikave të bazuara në kohë, veçanërisht nëse përgjigjet e serverit nuk janë shumë të qëndrueshme (duke e bërë një sulm përfundimtar të bazuar në kohë të pasigurt). [11]

# **Karakteristikat**

## **Karakteristikat gjenerike**

Sqlmap i jep mbështetje te plotë MySQL, Oracle, PostgreSQL, Microsoft SQL Server, Microsoft Access, IBM DB2, SQLite, Firebird, Sybase, SAP MaxDB, Informix, MariaDB, MemSQL, TiDB, CockroachDB, HSQLDB, H2, MonetDB, Apache Derby, Amazon Redsh Vertica, Mckoi, Presto, Altibase, MimerSQL, CrateDB, Greenplum, Drizzle, Apache Ignite, Cubrid, InterSystems Cache, IRIS, eXtremeDB dhe FrontBase sistemeve të menaxhimit të bazës së të dhënave, si dhe perkrahë pesë teknika të SQL injektimit, si: boolean-based blind, time-based blind, error-based, UNION query-based, stacked queries and out-of-band.

Sqlmap mund të lidhet direkt me bazën e të dhënave pa kaluar përmes një SQL injektimi, duke siguruar kredencialet, adresën IP, portin dhe emrin e bazës së të dhënave, ku si përfundim mund të nxjerr numrin e përdoruesit, hash fjalëkalimin, privilegjet, rolet, tabelat dhe kolonat. Poashtu mund të bëjë njohjen automatike e formateve të hash-it të fjalëkalimit dhe mbështetje për thyerjen e tyre duke përdorur një sulm të bazuar në fjalor. Mbështet hedhjen e të gjitha tabelave të bazës së të dhënave, një sërë shënimesh ose kolona specifike sipas zgjedhjes së përdoruesit. Përdoruesi mund të zgjedhë të hedhë vetëm një varg të karaktereve nga secila kolonë. Mbështet kërkimin për emra specifik të bazës së të dhënave, tabela specifike në të gjitha bazat e të dhënave ose kolona specifike në të gjitha tabelat e bazave të të dhënave. Mbështetje për të shkarkuar dhe ngarkuar ndonjë file nga sistemi i file-it themelor të serverit të bazës së të dhënave kur softueri i bazës së të dhënave është MySQL, PostgreSQL ose Microsoft SQL Server. Gjithashtu mund të ekzekutoj komanda arbitrare për të rimarrë prodhimin e tyre standard në serverin e bazës së të dhënave që qëndron në sistemin operativ kur softueri i bazës së të dhënave është MySQL, PostgreSQL ose Microsoft SQL Server. Mbështetje për të përsëritur strukturën e tabelave të bazës së të dhënave back-end dhe shënimet në një bazë të të dhënave lokale SQLite 3 dhe mbështetje për të vendosur një lidhje të paqëndrueshme TCP jashtë bandës midis makinës sulmuese dhe serverit të bazës së të dhënave që qëndron në sistemin operativ. [2]

## **Karakteristikat e numërimit dhe fingerprint-it**

Versionin e zgjeruar të softuerit të bazës së të dhënave back-end dhe sistemit themelor operativ fingerprint të bazuar në gabimet e mesazheve dhe karakteristikat e vecanta sic është injeksioni i komentit MySQL. Gjithashtu është e mundur edhe forcimi i emrit të DBMS nëse tashmë dihet. Softuer të thjeshtë të ueb serverit dhe teknologji fingerprint të ueb aplikacioneve. Mbështetje për të numëruar përdoruesit e databazës, hash të fjalekalimeve, rolet dhe privilegjet e përdoruesve, databazat, tabelat dhe kolonat. Formatin për njohjen automatike të hash të fjalekalimeve dhe mbështetjen për thyerjen e tyre me një sulm të bazuar në ndonjë fjalor. Mbështetje për të brute-force emrin e tabelave dhe kolonave.Kjo është e nevojshme kur sesioni i përdoruesit nuk ka qasje leximi në sistemin e tabelave që përmbajnë informacionin e skemës ose kur DBMS nuk e ka të rujatur këtë informacion(p.sh MySQL < 5.0). Mbështetje për kërkimin e emrave të vecantë të databazës, tabelave të vecanta në databazë apo kolonave të vecanta në tabela.Kjo ështeë e nevojshme për të identifikuar tabelat që përmbajnë të dhënat kredenciale të aplikacionit, ku emrat e kolonave përkatese përmbajnë string si emrin dhe fjalëkalimin. Mbështetje për ekzekutimin e custom SQL statements si në rastin kur një klient interaktiv SQL lidhet me databazën back-end, sqlmap automatikisht zbërthen statement e dhënë dhe përcakton se cila metodë përshtatet më së miri për injektim.

## **Takeover karakteristikat**

Mbështet shkarkimin dhe ngarkimin të çfarëdo file-i nga serveri i bazës së të dhënave që qëndron në themel të sistemit kur programi i bazës së të dhënave është MySQL, PostgreSQL ose Microsoft SQL Server, përkrahe poashtu injektimin e funksioneve të personalizuara të përdoruesit: një përdorues mund të përpilojë një bibliotekë të përbashkët, pastaj të përdorë sqlmap për të realizuar ndryshime brenda funksioneve të përdoruesit të sistemit të menaxhimit të databazës. Këto funksione mund të ekzekutohen, dhe të hiqen opsionalisht, përmes sqlmap. Kjo suportohet kur softueri i bazës së të dhënave është MySQL ose PostgreSQL.

# **Injeksionet**

**4.1 Komanda -p**

Sqlmap automatikisht teston të gjitha GET dhe POST parametrat. Kur vlera e --level është >=2 teston gjithashtu edhe vlerat e HTTP Cookie header. Kur kjo vlere eshte >=3 teston HTTP User-Agent dhe vlerat e HTTP Referer header për SQL injeksionet.

Gjithsesi, është e mundur që të specifikohet manualisht një listë prerëse-ndarëse (,) e parametrave që dëshrirojmë t’i testojmë në sqlmap. Për shembull, për të testuar vetëm GET parametrin id dhe HTTP User-Agent, shkruajmë provide –p id, user-agent

**4.2. Komanda: --dbms**

Sqlmap në mënyrë automatike detekton sistemin e menaxhimit të bazës së të dhënave të ueb aplikacioneve back-end. Versioni 9.0 i Sqlmap, mbështet plotësisht këto sisteme menaxhimi të databazës:

· MySQL

· Oracle

· PostgreSQL

· Microsoft SQL Server

· Microsoft Access

· SQLite

· Firebird

· Sybase

· SAP MaxDB

Nëse për cfarëdo lloj arsye Sqlmap dështon në detektimin e back-end DBMS pas identifikimit të ndonjë SQL injektimit, mundësohet emërtimi i back-end DBMS sipas vetë preferancave të përdoruesit (p.sh postgresql).

Per MySQL dhe Microsoft SQL Server mundësohet forma MySQL <version> apo Microsoft SQL Server <version>, ku <version> është një version valid për DBMS, sic është 5.0 për MySQL dhe 2005 për Microsoft SQL Server.

Vërejtje: Ky opsion nuk është i detyrueshëm dhe rekomandohet të përdoret vetëm në rastin kur jeni plotësisht të sigurtë për sistemin e menaxhimit të bazës së të dhënave back-end. Nëse nuk jeni, lejoni sqlmap të kryejë atë per ju.

**4.3 Komanda --os**

Sqlmap në mënyrë automatike detekton sistemin operativ themelor të sistemit të menaxhimit së bazës së të dhënave te ueb aplikacioneve back-end, kur ky informacion është vartësi ndonjë komande tjetër të mundshme. Për momentin, sisteme operative plotësisht të suportuara janë dy:

· Linux

· Windows

Është e mundur të forcohet emri i sistemit operativ nëse përdoruesi paraprakisht e di emrin, kështu që Sqlmap mënjanon kryerjen e këtij veprimi.

Vërejtje: Ky opsion nuk eshtë i detyrueshëm dhe rekomandohet të përdoret vetëm në rastin kur jeni plotësisht të sigurtë për sistemin operativ themelor të sistemit të menaxhimit të bazës së të dhënave back-end. Nëse nuk e dini, lejoni Sqlmap automatikisht të identifikon atë për ju.

# **Përdorimi i Sqlmap**

Përmes testimeve do të mundohemi të tregojmë përdorimin e Sqlmap për të zbuluar dhe shfrytëzuar të metat e SQL injektimeve duke marrë ID-në e emailit, përdoruesin(username) dhe fjalëkalimin nga baza e të dhënave të çdo faqeje interneti të prekshme nga SQL injektimet në Kali Linux.

Si shembull për këto sulme do të mirret faqja: <http://testphp.vulnweb.com/listproducts.php?cat-1> ( Fig.1) të zbulojmë bazën e të dhënave e cila është një faqe që lejohet për testimin dhe zbulimin e të metave të SQL injektimit duke përdorur Sqlmap si dhe do te përdoren disa nga komandat për të numëruar bazën e të dhënave përmes një aplikacioni të prekshëm nga SQL injektimet.

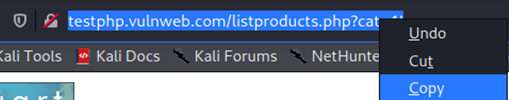


Fig. Linku

**5.1 Komanda --dbs**

Për të marrë një listë të bazave të të dhënave, përdorim opsionin –dbs, me ane te komandës sikur ne Fig. 2 :



Fig. Komanda -dbs

Pas ekzekutimit të komandes: **Sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat-1 –dbs, s**hfaqen emrat e dy baza të të dhënave për vulnweb.com (Fig. 3): **[\*] acuart dhe [\*]information\_schema**

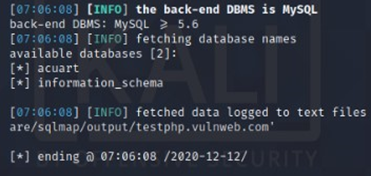


Fig. Rezultati i komandes --dbs

**5.2 Komanda --D**

Për ta gjetur listen e tabelave në bazën e të dhënave dhe më shumë detaje në burimet kryesore (acuart), përdorim opsionin --tables. Përveç kësaj, ne duhet të specifikojmë tabelën përmes opsionit –D, komanda është paraqitur ne Fig.4 :



Fig. Komanda --D

Pas ekzekutimit të komandës: **Sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat-1 --D acuart –tables,** shfaqen emrat e tabelave për vulnweb.com, Fig. 5:

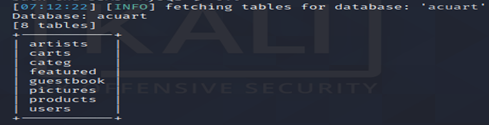


Fig. Rezultatet e komandes

**5.3 Komanda –T dhe --columns**

Tani ne do te gjejmë listën e kolonave nga tabela e përdoruesve dhe për ta bërë këtë përdorni opsionin - columns. Përveç kësaj, ne duhet të tregojmë bazën e të dhënave për të cilën ne jemi të interesuar ( -D acuart) dhe pas çelësit -T tabela ne shënojmë emrin për të cilën duam të shohim listën e kolonave në ketë rast për shfrytëzuesit (users), Fig. 6 :

Fig. Komanda -T dhe –columns

Pas ekzekutimit të komandës: **Sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat-1 --D acuart -T users –columns,** siç po shihet përmes opsionit column kemi gjetur emrin, email-in, fjalekalimin, adresen dhe disa të dhëna të tjera, Fig. 7.

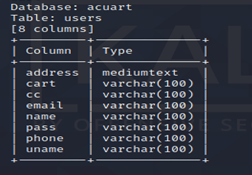


Fig. Rezultatet e komandes

**5.4 Komanda --dump**

Për të gjetur te dhëna për kolona te caktuar përdorim opsionin --dump. Në këtë opsion ne dëshirojmë të marrim si rezultat emailin, emrin dhe fjalëkalimin dhe për këtë përdorim komandën sikur ne Fig. 8.

Fig. Komanda --dump

Pas ekzekutimit të komandës: **Sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat-1 --D acuart -T users -C email,name,oass –dump**

Rezultati është shfaqur ne Fig 9.

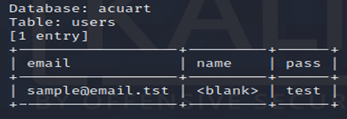


Fig. Rezultati i komandes

**5.5 Komanda -current-user**

Për të identifikuar përdoruesin aktual të bazës së të dhënave përdorim opsionin current user të paraqitur ne Fig. 10:



Fig. Komanda -curent-user

Pas ekzekutimit të komandës: **Sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat-1 --current-user,** shfaqet emri i përdoruesit momental, rezultati i te cilit është paraqitur ne Fig. 11

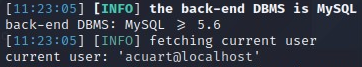


Fig. Rezultati i komandës

**5.6 Komanda –current-db**

Për të identifikuar emrin aktual të bazës së të dhënave përdorim opsionin current db, Fig. 12 :



Fig. Komanda --current-db

Pas ekzekutimit të komandës: **Sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat-1 --current-db,** rezultati do shfaq emrin e bazës së të dhënave, sikur ne Fig. 13:



Fig. Rezultatet e komandes

**5.7 Komandat –privileges, --roles, --is—dba dhe --hostname**

Për të identifikuar privilegjet, rolet dhe nëse përdoruesi aktual i DB është administratori i DB-së, përdorim komandat e paraqitura ne Fig. 14:



Fig. Komandat –privileges, --roles, --is-dba

Pas ekzekutimit të komandës: **Sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat-1 --privileges --roles --is-dba –hostname,** rezultati është paraqitur ne Fig. 15:

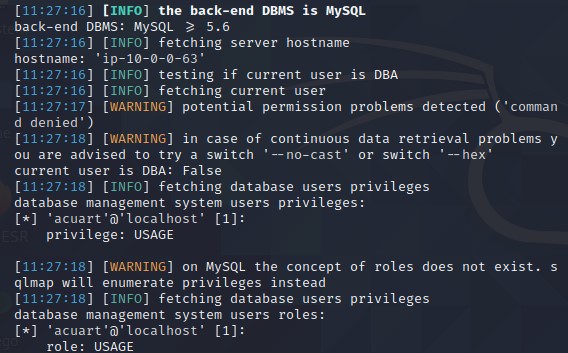


Fig. Rezultatet e komandës

Pra kështu mund te gjejmë të dhëna kredenciale rreth një shfletuesi te caktura përmes SQL errors duke përdorur mjetin Sqlmap.

# **Konkluzion**

Si përfundim mund të themi se Sqlmap është vegël që ka aftësin për të skanuar plotësisht një aplikacion në internet për të gjetur dobësitë e SQL injektimit dhe automatikisht të shfrytëzojë një dobësi të zbuluar për të marrë bazën e të dhënave. Kjo vegël ofrohet e parainstaluar në Kali Linux dhe është një mjet bazik për çdo kontrollues profesional të sigurisë. Çfarëdo URL(Linku) që i ipet kjo vegël ekzekuton automatikisht një skanim të plotë të SQL injektimit dhe nëse është e mundur nxjerr detaje të bazës së të dhënave dhe detajet e përdoruesit të DB. Këto informata të numëruara të databazës përfshijnë baza të të dhënave, role, privilegje, përdorues, tabela dhe kolonat e tyre dhe madje mund të marrin vlera hash të fjalëkalimeve. Ai madje ka aftësinë të anashkalojë firewalls (WAF) duke përdorur skripta manipulimi të paraprogramuara.

Mirëpo si në çdo vegël të automatizuar përqindja e saktësis nuk është 100% prandaj edhe Sqlmap na jep rezultate pozitive të rrejshme ku përdoruesi apo testuesi duhet të kontrolloj manualisht dhe të konfirmoj nëse një dobësi ekziston me të vërtet apo jo.

**Vërejtje edhe njëherë:** Përdorimi i Sqlmap për sulmimin e objektivave pa pëlqimin paraprak të ndërsjellë është i paligjshëm. Përgjegjësia është e përdoruesit dhe duhet t'i bindet të gjitha ligjeve të zbatueshme lokale, shtetërore dhe federale. Zhvilluesit nuk marrin asnjë përgjegjësi dhe nuk janë përgjegjës për çdo keqpërdorim ose dëmtim të shkaktuar nga ky program.

# **Biblografia**

[1] <https://tools.kali.org/vulnerability-analysis/sqlmap>

[2] <https://portswigger.net/web-security/sql-injection>

[3] [https://beaglesecurity.com/blog/vulnerability/boolean-based-blind-sql-injection.html#:~:text=Boolean%2Dbased%20SQL%20injection%20is,SQL%20query%20to%20the%20database.&text=The%20result%20allows%20an%20attacker,from%20the%20database%20are%20recovered](https://beaglesecurity.com/blog/vulnerability/boolean-based-blind-sql-injection.html" \l ":~:text=Boolean%2Dbased%20SQL%20injection%20is,SQL%20query%20to%20the%20database.&text=The%20result%20allows%20an%20attacker,from%20the%20database%20are%20recovered).

[4] <https://github.com/sqlmapproject/sqlmap/wiki/FAQ>

[5] <http://sqlmap.org/>

[6] <https://github.com/sqlmapproject/sqlmap/wiki/Screenshots>

[7] <https://github.com/sqlmapproject/sqlmap>

[8] [https://beaglesecurity.com/blog/vulnerability/time-based-blind-sql-injection.html#:~:text=Time%2Dbased%20SQL%20injection%20is,injection%20or%20blind%20injection%20attack.&text=In%20a%20time%2Dbased%20attack,query%20is%20true%20or%20false](https://beaglesecurity.com/blog/vulnerability/time-based-blind-sql-injection.html" \l ":~:text=Time%2Dbased%20SQL%20injection%20is,injection%20or%20blind%20injection%20attack.&text=In%20a%20time%2Dbased%20attack,query%20is%20true%20or%20false).

[9] <https://medium.com/@hninja049/example-of-a-error-based-sql-injection-dce72530271c>

[10] [https://sqlwiki.netspi.com/injectionTypes/unionBased/#mysql](https://sqlwiki.netspi.com/injectionTypes/unionBased/" \l "mysql)

[11] <https://beaglesecurity.com/blog/vulnerability/stacked-queries-sql-injection.html>