Alati za penetracijsko testiranje web aplikacija na operacijskom sustavu Kali Linux

Debogović, Artur

Master's thesis / Specijalistički diplomski stručni

2019

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Algebra University College / Visoko učilište Algebra**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:225:592841

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-03-29**

ALGEBRA VISOKO UČILIŠTE

Repository / Repozitorij:

Algebra Univerity College - Repository of Algebra Univerity College

zir.nsk.hr

dabar
DIGITALNI AKADEMSKI ARHIVI I REPOZITORIJI

VISOKO UČILIŠTE ALGEBRA

DIPLOMSKI RAD

Alati za penetracijsko testiranje web aplikacija na operacijskom sustavu Kali Linux

Artur Debogović

Zagreb, rujan 2018.

Predgovor

Ovim putem zahvaljujem se svom mentoru dr. sc. Damiru Deliji na strpljenu, pruženim savjetima i izvrsnom vodstvu pri izradi ovoga diplomskoga rada. Također se zahvaljujem svojim roditeljima koji su mi bili podrška tijekom cijelog studija i bez kojih sve ovo ne bi bilo moguće.

Sažetak

Naslov: Alati za penetracijsko testiranje web aplikacija na operacijskom sustavu Kali linux.

Cilj ovoga diplomskoga rada je pružiti uvid u postupak tj. osnove penetracijskog testiranja web aplikacija kroz primjenu raznih alata Kali Linux operativnog sustava. Rad se sastoji od dva dijela: teorijskog i praktičnog. U teorijskom dijelu govori se općenito o Kali Linux-u i penetracijskom testiranju dok se u praktičnom dijelu rada provodi postupak penetracijskog testiranja nad raznim web aplikacijama u svrhu prikupljanja informacija o serveru, identifikacije ranjivosti u web aplikacijama i na kraju iskorištavanje ranjivosti pronađene u web aplikacijama. Za potrebe demonstracije praktičnog dijela rada koristi se OWASP Broken Web Application Project v1.2 virtualno računalo koje je namijenjeno za učenje o sigurnosti web aplikacija, isprobavanju raznih alata za penetracijsko testiranje itd. , a da se pritom ne krši zakon.

Ključne riječi: Sigurnost web aplikacija, penetracijsko testiranje, Kali Linux, ranjivosti

Summary

Title: Penetration testing tools for web applications in the Kali Linux operating system

The main goal of this masters thesis is to provide insight into the process of penetration testing of web applications through the application of various Kali Linux operating system tools. Thesis consists of two parts: theoretical and practical. Theoretical part covers basic theory about Kali Linux and penetration testing, while practical part consists of performing the penetration testing process over various web applications for the purpose of collecting information about server, identifying vulnerabilities and finally exploiting vulnerabilities found in various web applications. For the purpose of demonstrating the practical part of the thesis, the OWASP Broken Web Application Project v1.2 virtual machine is being used because it is intended for learning about web application security, testing various penetration testing tools etc. without violating the law.

Key words: Web application security, penetration testing, Kali Linux, vulnerability

Sadržaj

1. Uvod 4
2. Općenito o Kali Linuxu 5
2.1. Kategorije alata u Kali Linux-u 5
3. Općenito o penetracijskom testiranju 7
3.1. Tipovi penetracijskog testiranja 8
3.1.1. Black box pristup 8
3.1.2. White box pristup 9
3.2. Procjena ranjivosti naspram penetracijskom testiranju 9
3.3. Svrha penetracijskog testiranja 10
3.4. Metodologije ispitivanja sigurnosti 11
3.4.1. Open Source Security Testing Methodology Manual (OSSTMM) 12
3.4.1.1 Ključne značajke i prednosti 14
3.4.2. Information Systems Security Assessment Framework (ISSAF) 15
3.4.2.1 Ključne značajke i prednosti 16
3.4.3. Open Web Application Security Project (OWASP) 17
3.4.3.1 Ključne značajke i prednosti 17
3.4.4. Web Application Security Consortium Threat Classification (WASC-TC) 18
3.4.4.1 Ključne značajke i prednosti 20
3.4.5. Penetration Testing Execution Standard (PTES) 20
3.4.5.1 Ključne značajke i prednosti 21
3.5. Opći model penetracijskog testiranja 21
3.5.1. Definiranje područje primjene 22
3.5.2. Prikupljanje informacija 23
3.5.3. Otkrivanje ciljnog područja 23
3.5.4. Enumeracija 24

3.5.5. Mapiranje ranjivosti ... 24
3.5.6. Socijalni inženjering ... 24
3.5.7. Iskorištavanje ranjivosti ... 25
3.5.8. Eskalacija privilegija ... 26
3.5.9. Održavanje pristupa ... 26
3.5.10. Dokumentacija i izvješćivanje ... 26
4. Izviđanje ... 27
4.1. Nmap ... 27
4.2. Identifikacija Vatrozida za zaštitu web aplikacija ... 29
4.3. Analiza izvornog koda web stranice ... 31
4.4. Firebug alat za analizu i modifikaciju elemenata web stranice ... 32
4.5. Dohvaćanje i modifikacija cookie-a ... 34
4.6. Otkrivanje skrivenih web stranica i direktorija uz pomoć robots.txt ... 35
4.7. DirBuster-alat za pronalaženje datoteka i direktorija na web serveru ... 37
4.8. CeWL-alat za profiliranje lozinki ... 38
4.9. JohnTheRipper ... 40
5. Indeksiranje web stranica i direktorija ... 41
5.1. Ponavljanje zahtjeva sa Burp repeater-om ... 43
5.2. Identificiranje relevantnih datoteka i direktorija iz rezultata indeksiranja ... 46
6. Identifikacija ranjivosti ... 46
6.1. Hackbar ... 47
6.2. Presretanje i modificiranje zahtjeva sa Tamper Data dodatkom za firefox web preglednik ... 49
6.3. Presretanje i modifikacija zahtjeva sa Burp Suite-om ... 50
6.4. Identificiranje cross-site scripting (XSS) ranjivosti ... 53
6.5. Identifikacija SQL injection ranjivosti na temelju poruka o greškama ... 55

6.6. Identificiranje blind SQL injection ranjivosti 57
6.7. Identificiranje ranjivosti u web kolačićima(Cookies) 59
6.8. SSLScan alat za dohvaćanje informacija o SSL i TLS protokolu 60
6.9. Identificiranje POODLE ranjivosti 62
7. Alati za automatizirano skeniranje ranjivosti 63
7.1. Nessus 63
7.2. Wapiti 67
7.3. Nikto 71
8. Iskorištavanje ranjivosti 73
8.1. Brute force napad na login stranice s Burp-om 73
9. Zaključak 78
Popis slika 79
Literatura 83

1. Uvod

U današnje vrijeme gotovo svaki dan je moguće putem medija čitati o rušenje web stranica tj. DDoS napad, curenju podataka o milijunima korisničkih računa, krađi brojeva kreditnih kartica i krađi identiteta na web stranicama… . Iz tog razloga kako bi sačuvali svoj ugled i povjerenje korisnika od izuzetne je važnosti za organizacije da shvate rizike koji su vezani za informacijsku sigurnost i potrebu za zaštitom osjetljivih podataka tj. da znaju koje su im slabosti u infrastrukturi, kako mogu biti napadnuti i kakve će biti posljedice u smislu izgubljenih/ukradenih informacija ili kompromisa sustava ukoliko napad bude uspješan i što je najvažnije kako otkloniti ranjivosti i minimizirati rizik. To posebno vrijedi za organizacije koje su dostupne javnosti putem interneta.

Upravo je to zadatak penetracijskog testera koji koristi iste tehnike i alate kao i pravi maliciozni korisnici kako bi otkrio ranjivosti i njihov utjecaj na sustav organizacije.

U ovom radu testira se sigurnost pojedinih web aplikacija na virtualnom računalu OWASP Broken Web Application Project v1.2 koje ima ulogu servera koristeći alate Kali Linux OS-a.

Rad se sastoji od dva dijela. U prvom dijelu tj. teoretskom govori se općenito o tome što je penetracijsko testiranje, koja je razlika između pojedinih tipova penetracijskog testiranja, raznim metodologijama po kojima je moguće testirati sigurnost i osnovnim koracima penetracijskog testiranja. U praktičnom dijelu rada demonstrira se rad alata koji se koriste u penetracijskom testiranju u svrhe kao što su npr. prikupljane informacija o ciljanom sustavu, identifikacija vatrozida i IDS/IPS sustava ,popisivanje relevantnih web stranica i direktorija, automatizirano otkrivanje ranjivosti(usporedbe radi demonstrira se i ručno otkrivanje ranjivosti) i na kraju primjer iskorištavanja ranjivosti na temelju ranije prikupljenih informacija.

2. Općenito o Kali Linuxu

Kali Linux (Kali) je distribucija Linux operativnog sustava koja je razvijena s naglaskom na zadatke penetracijskog testiranja . Prethodno inačica Kali Linux-a poznata je kao [1]BackTrack, koji je nastao spajanjem tri različite distribucije Linux-a namijenjenih penetracijskom testiranju: IWHAX, WHOPPIX i Auditor.

Prva verzija Kali Linux-a (verzija 1.0) objavljena je 12. ožujka 2013.

[2]Glavne značajke Kali Linux-a su:

- Bazira se na Debian Linux distribuciji.
- Potopno je besplatan i dostupan na korištenje svima koji žele.
- Posjeduje više od 600 aplikacija za penetracijsko testiranje.
- Ima podršku za široki raspon bežičnih mrežnih kartica.
- Posebno prilagođeni kernel za „modifikaciju“ mrežnih paketa.
- Svaki softverski paket sadrži GPG([3]GNU Privacy Guard) potpis Developer-a.
- Korisnici imaju mogućnost prilagođavanja Kali Linux-a svojim potrebama.
- Podržava sustave bazirane na ARM arhitekturi

2.1. Kategorije alata u Kali Linux-u

[4]Kali Linux sadrži niz alata koji se mogu koristiti tijekom procesa penetracijskog testiranja. Alati za penetracijsko testiranje u Kali Linux-u mogu se svrstati u sljedeće kategorije:

- Prikupljanje informacija: Ova kategorija sadrži nekoliko alata koji se mogu koristiti za prikupljanje informacija o DNS-u,mrežnim rutama, IDS / IPS-u ,mreži, operativnim sustavima,SSL-u, SMB-u,VPN-u, VoIP-u, SNMP-u, e-mail adresama i VPN-u.
- Procjena ranjivosti: U ovoj kategoriji nalaze se alati koji služe za skeniranje/pronalaženje ranjivosti općenito. Također obuhvaća alate za procjenu

[1] https://www.backtrack-linux.org/ , 11.6.2018. 11:37:47
[2] http://docs.kali.org/introduction/what-is-kali-linux, 11.6.2018. 11:47:39
[3] https://www.gnupg.org/ , 11.6.2018. 11:53:26
[4] Web Penetration Testing with Kali Linux, (Joseph Muniz, 2013: 30)

Cisco-ve mreže i alate za procjenu ranjivosti na raznim poslužiteljima baza podataka. Ova kategorija također uključuje i nekoliko alata za [5]„fuzzing“.

- Web aplikacije: U ovu kategoriju spadaju alati vezani uz web aplikacije kao što je skener sustava za upravljanje sadržajem, alati za iskorištavanje ranjivosti pojedinih baza podataka, fuzzer-i za web aplikacije, proxy-i za web aplikacije, botovi za web indeksiranje i skeneri web ranjivosti.
- Alati za probijanje lozinke: U ovoj kategoriji nalaze se alati koji se mogu koristiti za napad na lozinke(online ili offline).
- Alati za eksploataciju ranjivosti: Ova kategorija sadrži alate koji se mogu koristiti za iskorištavanje ranjivosti koje se nalaze u ciljanom okruženju. Uz ove alate se često u kombinaciji koriste i alati za provođenje napada socijalnog inženjeringa kako bi se došlo do informacija o potencijalnim ranjivostima.
- Sniffing i spoofing: Alati u ovoj kategoriji mogu se koristiti za špijuniranje mreže i web prometa što može dovesti do presretanja i krađe podataka. Ova kategorija također uključuje alate za maskiranje(spoofing) mreže kao što su Ettercap i Yersinia.
- Održavanje pristupa: U ovoj kategoriji nalaze se alati koji služe za održavanju pristupa na ciljanom računalu. U nekim slučajevima biti će potrebno prvo dobiti najvišu razinu ovlasti na ciljanom uređaju prije nego što će biti moguće instalirati alate u ovoj kategoriji. Kada su svi preduvjeti ispunjeni mogu se koristiti alati za backdooring operacijskog sustava i web aplikacije, ali i alati za tuneliranje.
- Alati za izradu izvještaja: U ovu kategoriju spadaju alati koji služe kao pomoć pri dokumentiranju postupka i rezultata penetracijskog testiranja.
- Sistemski servisi: Obuhvaćaju nekoliko servisa koji mogu biti korisni tijekom penetracijskog testiranja, kao što su Apache servis, MySQL servis, SSH servis i Metasploit servis.
- Osim što sadrži alate koji se koriste za penetracijsko testiranje, Kali Linux također dolazi s nekoliko alata koji se mogu koristiti za sljedeće:
- Napadi na bežične mreže: Ova kategorija uključuje alate za napad na Bluetooth, RFID / NFC i bežične uređaje.
- Reverzni inženjering: Ova kategorija sadrži alate koji se mogu koristiti za debug programa ili rastavljanje izvršne datoteke. Svrha ovog postupka je analizirati proces

[5] https://www.owasp.org/index.php/Fuzzing, 12.6.2018. 11:27:27

razvoja programa/aplikacije. Reverzni inženjering se također koristi u analizi zlonamjernog softvera tj. malware-a kako bi se utvrdilo kako malware funkcionira ili u istraživanju u kojem se pokušavaju identificirati ranjivosti u softverskim aplikacijama.

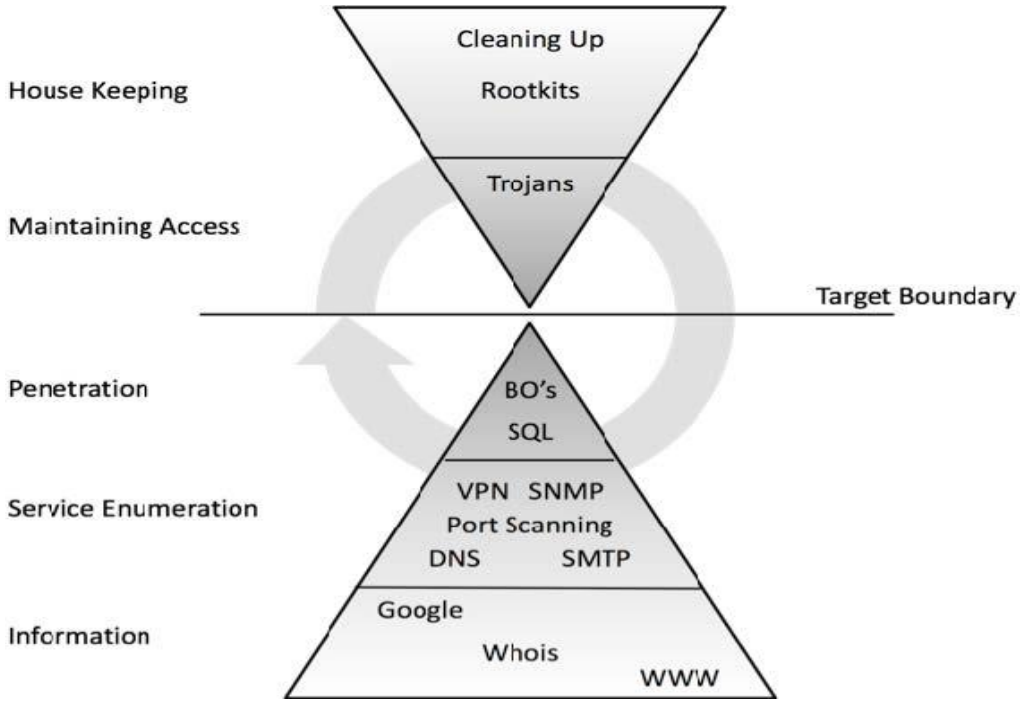
- Ispitivanje stresa: U ovoj kategoriji sadržani su alati koji mogu pomoći u testiranju opterećenja na mreži, ali i u bežičnom, web i VoIP okruženju.
- Hakiranje hardware-a: Alati u ovoj kategoriji mogu se koristiti kod rada s Android i Arduino aplikacijama.
- Forenzika: U ovoj kategoriji nalazi se nekoliko alata koji se koristite u digitalnoj forenzici, a neke od primjena su izrada forenzičke slike diska tj. kloniranje tvrdog diska, rekonstrukcija datoteka i analiza kopije tvrdog diska. Važno je napomenuti da je prilikom provođenja postupka digitalne forenzike potrebno slijediti [6]Kali Linux Forensics smjernice kako se ne bi narušio integritet podataka na uređaju na kojem se vrši analiza.

U ovom radu fokus će biti na alate za penetracijsko testiranje Kali Linuxa.

3. Općenito o penetracijskom testiranju

[7]Penetracijsko testiranje predstavlja kontinuirani ciklus istraživanja, a potom i napada na određenu metu. Napad treba biti strukturiran i proračunat te ako je ikako moguće testiran/verificiran u laboratorijskom okruženju prije nego se provede uživo. Na Slici 1. nalazi se vizualni prikaz metodologije penetracijskog testiranja:

[6] https://docs.kali.org/general-use/kali-linux-forensics-mode, 13.6.2018. 15:37:40
[7] Penetration Testing with Kali Linux v1.0.1, Offensive Security, 2014: 13



[8]Slika 3.1 Dijagram metodologije penetracijskog testiranja

Iz slike 3.1 se može zaključiti da čim je opseg prikupljenih informacija veći, to je vjerojatnost uspješnog provaljivanja veća. Jednom kada se prođe početna ciljna granica, slijedi ponovno započinje ciklus istraživanja - na primjer, skupljanje informacija o internoj mreži kako bi se prodrlo dublje u sustav.

U konačnici , svaki profesionalac za računalnu sigurnost razvija vlastitu metodologiju, obično temeljenu na specifičnim tehničkim snagama

3.1. Tipovi penetracijskog testiranja

Iako postoje različite vrste penetracijskog testiranja, dva najčešća pristupa koji su široko prihvaćeni od strane industrije su black box i white box.

3.1.1. Black box pristup

[9]Kod ovog pristupa, sigurnosni revizor procjenjuje mrežnu infrastrukturu te nije upoznat sa internim tehnologijama koje implementira ciljana organizacija. Kroz primjenu brojnih

[8] Penetration Testing with Kali Linux v1.0.1, Offensive Security, 2014: 14
[9] Kali Linux 2: Assuring Security by Penetration Testing Third Edition, Lee Allen, 2016: 48

tehnika hakiranja i prolazeći kroz organizirane testne faze, ranjivosti se mogu otkriti i potencijalno iskoristiti. Važno je da revizor razumije, klasificira i određuje prioritet tih ranjivosti prema njihovoj razini rizika (niska, srednja ili visoka). Rizik se može mjeriti s obzirom na prijetnju koju predstavlja pojedina ranjivost. U idealnom slučaju pen-tester bi identificirao sve moguće vektore napada koji bi mogli ugroziti ciljanu organizaciju. Kada proces testiranja završi, generira se izvješće koje sadrži sve potrebne informacije o stvarnom sigurnosnom stanju , kategorizaciji i prevođenju identificiranih rizika u poslovni kontekst. Black box testiranje je općenito skuplja usluga od White box varijante.

3.1.2. White box pristup

[10]Za razliku od black box pristupa kod White box penetracijskog testiranja revizor je upoznat s svim internim i temeljnim tehnologijama koje se koriste u ciljanom okruženju. Takav pristup penetracijskom testeru otvara široka vrata u smislu pregleda i kritičkoga vrednovanja sigurnosnih propusta uz minimalne moguće napore i najveću točnost. Klijent tj. organizacija ovim pristupom dobiva veću vrijednost u odnosu na black box pristup u smislu da će se ukloniti svi problemi vezani za interno okruženje organizacije tj. sigurnosni propusti koji se nalaze u okruženju ciljne infrastrukture te tako otežati infiltriranje zlonamjernog korisnika izvana. Koraci koji su uključeni u white box testiranje slični su koracima koji se primjenjuju kod black box testiranja. Štoviše, dobra praksa je da se white box pristup integrira u redovni životni ciklus razvoja organizacije kako bi se uklonili eventualni sigurnosni problemi u ranoj fazi prije nego što ih otkriju i iskoriste uljezi. Vrijeme, trošak i razina znanja koji su potrebni za pronalaženje i rješavanje sigurnosnih propusta su usporedivo manji nego kod black box pristupa

3.2. Procjena ranjivosti naspram penetracijskom testiranju

[11]Iako djeluju i zvuče kao ista stvar procjena ranjivosti i penetracijsko testiranje su u stručnoj terminologiji dva različita pojma. Organizacije bilo komercijalne ili nekomercijalne mogu pogrešno protumačiti pojam penetracijskog testiranja pri odabiru tipa procjene sigurnosti. Iz

[10]Kali Linux 2: Assuring Security by Penetration Testing Third Edition, Lee Allen, 2016: 49
[11]Kali Linux 2: Assuring Security by Penetration Testing Third Edition, Lee Allen, 2016: 50

tog razloga važno je razumjeti razlike između ove dvije vrste testova. Procjena ranjivosti je proces procjene unutarnjih i vanjskih sigurnosnih kontrola identificirajući prijetnje koje mogu dovesti do nepoželjne izloženosti imovine organizacije . Ova procjena tehničke infrastrukture ne ukazuje samo na rizike postojećih obrambenih mehanizama, nego također preporučuje i prioritetizira strategije za sanaciju propusta. Interna procjena ranjivosti pruža jamstvo za osiguranje internih sustava, dok vanjska procjena ranjivosti ukazuje na propuste (ukoliko postoje) u mehanizmima koji predstavljaju granicu između interne mreže organizacije i interneta. Kod oba kriterija testiranja, svaka stavka na mreži se strogo testira protiv više napadačkih vektora kako bi se identificirale prisutne prijetnje i kvantificirale reaktivne mjere. Ovisno o vrsti procjene koja se provodi, slijedi jedinstveni skup postupaka, alata i tehnika testiranja za automatsko otkrivanje i prepoznavanje ranjivosti koje mogu rezultirati imovinskom štetom za organizaciju. To se može postići koristeći integrirane platforme za upravljanje ranjivostima koje sadrže bazu podataka sa aktualnim ranjivostima i sposobna je testirati različite vrste mrežnih uređaja, a da pritom ne narušava integritet konfiguracije.

Ključna razlika između procjene ranjivosti i penetracijskog testiranja je da penetracijsko testiranje ide korak dalje od prepoznavanja ranjivosti tj. obuhvaća cjelokupni proces eksploatacije, eskalacije privilegija i održavanja pristupa ciljanom sustavu. S druge strane, procjena ranjivosti pruža široki prikaz svih postojećih nedostataka u sustavu bez mjerenja utjecaja tih nedostataka na sustav koji se razmatra. Druga velika razlika između ova dva pojma je da je penetracijsko testiranje znatno više nametljiv proces od procjene ranjivosti i agresivno primjenjuje sve tehničke metode za iskorištavanje ranjivosti u produkcijskom okruženju organizacije dok se proces procjene ranjivosti temelji na pažljivom prepoznavanju i kvantificiranju svih poznatih ranjivosti na neinvazivni način.

3.3. Svrha penetracijskog testiranja

Savršeno vrijeme za provedbu penetracijskog testiranja je kada postoji sumnja da su sigurnosni mehanizmi poput vatrozida, IDS/IPS sustava (sustavi za otkrivanje upada), sustava za praćenje integriteta datoteka itd. učinkoviti. Dok procjena ranjivosti ima za cilj pronalazak pojedinačnih ranjivosti penetracijskim testiranjem će se pokušati potvrditi da su te ranjivosti iskoristive u ciljanom okruženju. Kvalificirani konzultant uvijek pokušava izraditi najbolju vrstu procjene temeljenu na poslovnom zahtjevu klijenta. S druge strane dužnost klijenta je da prije donošenja konačne odluke razmotri ključne detalje odabranog

programa procjene sigurnosti. Penetracijsko testiranje je skupa usluga u usporedbi s procjenom ranjivosti**.**

3.4. Metodologije ispitivanja sigurnosti

[12]Postoje razne metodologije otvorenog koda za potrebe procjene sigurnosti. Koristeći ove metodologije, strateški se mogu riješiti vremenski kritični i izazovni zadaci procjene sigurnosti sustava bez obzira na njegovu veličinu i složenost. Neke metodologije usredotočuju se na tehnički aspekt sigurnosnih testiranja, dok se druge usredotočuju na rukovodilačke kriterije, a vrlo malo ih se odnosi na oba aspekta. Osnovna ideja ovih metodologija je provođenje različitih vrsta testova korak po korak kako bi točno procijenilo sigurnosno stanje sustava.

Neke od poznatijih metodologija procjene sigurnosti koje pružaju prošireni pogled prilikom procjene sigurnosti mreže i aplikacija ističući njihove ključne značajke i prednosti su:

- Open Source Security Testing Methodology Manual
- Information Systems Security Assessment Framework
- Open Web Application Security Project Testing Guide
- Web Application Security Consortium Threat Classification
- Penetration Testing Execution Standard

Svi ovi testni modeli i metodologije pomoći će sigurnosnim stručnjacima prilikom odabira najbolje strategiju koja je u skladu sa zahtjevima klijenata. Prve dvije metodologije pružaju opće smjernice i metode ispitivanja sigurnosti za gotovo bilo koju vrstu informacija. Metodologije Open Web Application Security Project (OWASP) i Web Application Security Consortium (WASC) prvenstveno se bave procjenom sigurnosti web aplikacija. Penetration Testing Execution Standard (PTES) pruža smjernice za sve aspekte penetracijskog testiranja. Međutim, važno je napomenuti da je sigurnost po sebi neprekidan proces i da je penetracijski test „snimka“ koja određuje sigurnosno stanje sustava za vrijeme trajanja testiranja. Svaka manja promjena u ciljanom okruženju može utjecati na cijeli proces ispitivanja sigurnosti i može predstavljati pogreške u konačnim rezultatima. Osim toga, prilagodba bilo koje jedinstvene metodologije ne mora nužno pružiti potpunu sliku procesa procjene rizika.

[12] Kali Linux 2: Assuring Security by Penetration Testing Third Edition, Lee Allen, 2014: 51

Sigurnosnom revizoru prepušta se odabir najbolje strategije koja bi se trebala baviti ciljanim kriterijima testiranja.

Postoje mnoge metodologije sigurnosnog testiranja; odabir najbolje zahtijeva pažljiv selektivni proces kroz koji se može utvrditi trošak i učinkovitost procjene. Stoga, utvrđivanje pravilne strategije procjene ovisi o nekoliko čimbenika, uključujući tehničke pojedinosti ciljanog okruženja i raspoloživost resursa, znanje penetracijskog testera, poslovne ciljeve i regulatorna pitanja. S poslovnog stajališta, učinkovitost i kontrola troškova su od izuzetne važnosti.

3.4.1. Open Source Security Testing Methodology Manual (OSSTMM)

[13]OSSTMM međunarodno je prepoznati standard čiji je autor Pete Herzog, a razvijen je od strane ISECOM-a za ispitivanje i analizu sigurnosti. Koriste ga mnoge organizacije u svakodnevnom ciklusu procjene. Iz tehničke perspektive, njegova metodologija je podijeljena u četiri ključne skupine - opseg, kanal, indeks i vektor. Opseg definira proces prikupljanja podataka o svim sredstvima koja djeluju unutar ciljanog okruženja. Kanal određuje vrstu komunikacije i interakcije s tim sredstvima koja mogu biti fizička, spektralna i komunikacijska. Svaki kanal predstavlja jedinstveni skup sigurnosnih komponenti koje moraju biti testirane i provjerene tijekom razdoblja procjene. Te komponente sastoje se od fizičke sigurnosti, ljudske psihologije, podatkovnih mreža, bežičnog komunikacijskog medija i telekomunikacija. Indeks je metoda koja se koristi za klasifikaciju ciljne imovine koja je obilježena posebnim identifikacijama, kao što su MAC adresa i IP adresa. Na kraju, vektor definira smjer kroz koji auditor može procijeniti i analizirati imovinu koja je ključna za funkcionalnost organizacije. Cijeli ovaj proces predstavlja tehnički putokaz kroz koji se temeljito ocjenjuje ciljana okolina, a poznat je kao opseg revizije.

[14]Postoje različiti oblici sigurnosnih testova koji su klasificirani prema OSSTMM metodologiji, a njihova je organizacija predstavljena kroz šest standardnih sigurnosnih testnih tipova:

[13] http://www.isecom.org/research/osstmm.html, 21.6.2018. 13:10:41
[14] Kali Linux 2: Assuring Security by Penetration Testing Third Edition, Lee Allen, 2016: 53

- Blind: slijepo testiranje ne zahtijeva prethodno poznavanje ciljnog sustava. Međutim, klijent se obavještava prije izvršenja testiranja. Etičko hakiranje je jedan od primjera ispitivanja ovog tipa. Ova vrsta testiranja je široko prihvaćena zbog svoje etičnosti tj. informiranja mete unaprijed.
- Double blind: Kod ovog tipa testiranja, revizor ne zahtijeva nikakvo znanje o ciljanom sustavu, niti je druga strana informirana prije izvršenja testa. Black box i penetracijsko testiranje su primjeri ovakvoga ispitivanja. Većina sigurnosnih procjena danas se provodi pomoću ove strategije što za revizore predstavlja pravi izazov u smislu odabira najboljih alata i tehnika kako bi postigli svoj cilj.
- Gray box: Kod gray box ispitivanja revizor ima ograničeno znanje o ciljanom sustavu, a druga strana se također obavještava prije nego što se test izvrši. Procjena ranjivosti jedan je od osnovnih primjera gray box ispitivanja.
- Double gray box: Double gray box Ispitivanje funkcionira na način sličan gray box testu, razlika je u tome što je definiran vremenski okvir za reviziju i nema testiranja kanala i vektora. White box testiranje je primjer ovog tipa ispitivanja.
- Tandem: U tandem testiranju, revizor ima minimalna znanja za procjenu ciljnog sustava, a druga strana se također obavještava unaprijed, prije nego što se test provede. Tandem testiranje se provodi izuzetno temeljito. Crystal box i interno ispitivanje primjeri su tandem testiranja.
- Reversal: Kod ovog testiranju revizor je u potpunosti upoznat s ciljnim sustavom, a druga strana nikada neće biti obaviještena o tome kako i kada će se test provesti.

Karakteristike koje pruža metodologija OSSTMM su fleksibilnost i sposobnost izvođenja određenih testova koji se u pravilu bave procjenom sigurnosti kontrole pristupa, sigurnosti procesa, kontrola podataka, fizičke lokacije, zaštite okoline, razine svijesti o sigurnosnim rizicima, razine povjerenja, zaštitom od prijevare itd. . Sveobuhvatni postupci testiranja usredotočeni su na ono što treba testirati, kako treba testirati, koje taktike treba primijeniti prije, tijekom i poslije testiranja i kako interpretirati i povezati konačne rezultate. Izrada „snimke“ trenutnog stanja sigurnosti i zaštite ciljnog sustava je od značajne važnosti za organizaciju. U tu svrhu metodologija OSSTMM uvodi pojam RAV (Risk Value Assessment). Osnovna funkcija RAV-a je analiza rezultata ispitivanja i izračun stvarne vrijednosti sigurnosti temeljene na tri čimbenika: operativna sigurnost, kontrola gubitaka podataka i ograničenja štete. Ova konačna sigurnosna vrijednost poznata je kao RAV

rezultat. Pomoću RAV bodova, revizor na temelju trenutnog stanja sigurnosti sustava lako može izdvojiti i definirati smjernice po kojima bi se postigla bolja zaštita.

Iz perspektive poslovanja, RAV može pomoći u optimiziranju potrebnih ulaganja za sigurnost kao i kod opravdanja ulaganja u učinkovitija sigurnosna rješenja.

3.4.1.1 Ključne značajke i prednosti

[15]Ključne značajke i prednosti OSSTMM-a:

- Korištenje OSSTMM metodologije značajno se smanjuje pojava lažnih negativa i lažnih pozitiva i osigurava ponovljivost sigurnosnih mjerenja.
- Prilagodljiva metodologija za mnoge vrste sigurnosnih testova, kao što su penetracijsko testiranje, White box testiranje, procjena ranjivosti itd. .
- Osigurava temeljitu provedbu procjene, a rezultati se prikupljaju na dosljedan, mjerljiv i pouzdan način.
- Sama metodologija slijedi proces od četiri pojedinačno povezane faze, a to su, faza definiranja, faza informiranja, regulacijska faza i faza ispitivanja kontrola. U svakoj od tih faza se prikupljaju, procjenjuju i provjeravaju informacije o ciljanom okruženju.
- RAV daje izračun stvarne vrijednosti sigurnosti na temelju operativne sigurnosti, kontrola gubitaka podataka i ograničenja štete. RAV rezultat predstavlja trenutačno stanje sigurnosti sustava.
- Formalizacija izvješća o procjeni pomoću [16]Security Test Audit Report (STAR) obrasca može biti korisna za menadžment i tehnički tim prilikom pregleda ciljeva testiranja, procjeni rizika i rezultata svake testne faze.
- Metodologija se redovito ažurira s novim trendovima ispitivanja sigurnosti, propisima i etičkim problemima.
- OSSTMM proces može biti usklađen s industrijskim propisima, poslovnom politikom i vladinim zakonima. Uz to, ovlaštena revizija može biti izravno prihvatljiva za akreditaciju od strane ISECOM-a (Instituta za sigurnost i otvorene metodologije).

[15] Kali Linux 2: Assuring Security by Penetration Testing Third Edition, Lee Allen, 2016: 54
[16] http://www.isecom.org/mirror/STAR.3.pdf , 24.6.2018. 19:54:38

3.4.2. Information Systems Security Assessment Framework (ISSAF)

[17]Information Systems Security Assessment Framework (ISSAF) (www.oissg.org/issaf) je još jedna open source metodologija za testiranje i analizu sigurnosti sustava. Testovi su kategorizirani u nekoliko domena kako bi se procjena sigurnosti vršila logičnim redoslijedom. Svaka od tih domena procjenjuje različite dijelove ciljnog sustava i daje smjernice za postizanje adekvatne razine sigurnosti.

Integracijom ISSAF –a u redoviti poslovni ciklus organizacije mogu se pružiti točnost, potpunost i učinkovitost potrebni za ispunjavanje uvjeta testiranja sigurnosti organizacije. ISSAF je razvijen s fokusom na dva područja testiranja sigurnosti - tehničko i upravljačko. Tehnička strana uspostavlja temeljni skup pravila i postupaka koji prate i stvaraju odgovarajući proces procjene sigurnosti, dok se upravljačka strana bavi suradnjom s upravom i najboljim praksama koje treba slijediti tijekom procesa testiranja. Važno je napomenuti da ISSAF definira procjenu sigurnosti kao proces, a ne reviziju.

Iako provedba revizije zahtijeva angažman akreditirane ustanove za procjenu potrebnih standarda, prednost je što kod ocjenjivanja uključuje faze planiranja, procjene, obrade, akreditacije i održavanja. Svaka od tih faza sadrži opće smjernice koje su učinkovite i fleksibilne za svaku organizacijsku strukturu.

Rezultat revizije je kombinacija operativnih aktivnosti, sigurnosnih inicijativa i kompletnog popisa ranjivosti koje bi mogle postojati u ciljanom okruženju. Za razliku od revizije kod procesa procjene se odabire najkraći put kako bi se testiranja obavilo u definiranom roku s ciljem analize sustava protiv kritičnih ranjivosti koje se mogu iskoristiti uz minimalan napor. ISSAF sadrži bogat set osnovnih tehničkih procjena za testiranje različitih tehnologija i procesa. Međutim, to uvodi problem održavanja tj. ažuriranja metodologije kako bi bila u skladu s novim ili ažuriranim kriterijima za procjenu tehnologije. U usporedbi s metodologijom OSSTMM, ovi problemi zastarijevanja imaju manji utjecaj na OSSTMM, iz razloga jer je revizor u mogućnosti koristiti istu metodologiju uz korištenje različitih

[17] Kali Linux 2: Assuring Security by Penetration Testing Third Edition, Lee Allen, 2016: 55

skupove alata i tehnika. S druge strane, [18]IOISSG (Open Information Systems Security Group) tvrdi da je njihova metodologija usklađena s najnovijim informacijama o sigurnosnim alatima, najboljim praksama i administrativnim problemima koji nadopunjuju program procjene sigurnosti. Također se može uskladiti s OSSTMM ili bilo kojom sličnom metodologijom testiranja, čime se kombiniraju njihove međusobne snage.

3.4.2.1 Ključne značajke i prednosti

[19]Ključne značajke i prednosti ISSAF-a:

- ISSAF daje prijedlog od značajne važnosti za osiguranje infrastrukture procjenom postojećih sigurnosnih kontrola naspram kritičnih ranjivosti.
- Bavi se ključnim područjima informacijske sigurnosti. To uključuje procjenu rizika, strukturu i upravljanje poslovanjem, procjenu kontrola, upravljanje angažmanom(praćenje da se tijekom provođenja penetracijskog testiranja klijentu isporuči ugovorena usluga), razvoj sigurnosnih politika i opće najbolje prakse.
- Ispituje sigurnost mreže, sustava ili aplikacija. Testiranje se može transparentno usredotočiti na specifičnu tehnologiju koja može uključivati usmjernike, preklopnike, vatrozid, IDS/IPS sustave (sustavi za otkrivanje i sprječavanje upada), SAN(Storage area network), VPN, različite operacijske sustave, poslužitelje na kojima se nalaze web aplikacije, baze podataka itd. .
- Uklanja granice između tehničkog i upravljačkog gledišta testiranja sigurnosti provođenjem potrebnih kontrola u oba područja.
- Omogućuje upravi lakše razumijevanje postojećih rizika koji potencijalno mogu dovesti do proboja/zaobilaženja obrambenih mehanizme organizacije te ih proaktivno smanjuje identificirajući ranjivosti koje mogu utjecati na integritet poslovanja. OSSTMM i ISSAF se mogu koristiti u kombinaciji za procjenu sigurnosti poslovnog okruženja.

[18] http://www.oissg.org/issaf.html , 1.7.2018. 14:35:20
[19] Kali Linux 2: Assuring Security by Penetration Testing Third Edition, Lee Allen, 2016: 56

3.4.3. Open Web Application Security Project (OWASP)

[20]Open Web Application Security Project (OWASP) zasniva se na 10 projekata kroz koje se nastoji povećati svijest o sigurnosti web aplikacija. Ova metodologija pruža potrebne temelje za integraciju sigurnosti putem smjernica i dobrih praksi kojih bi se trebalo pridržavati prilikom [21] razvoja web aplikacija. OWASP također pruža detaljni vodič za testiranje kao dio OWASP projekta (https://www.owasp.org/index.php/ OWASP_Testing_Project) .

[22]OWASP Top 10 projekt kategorizira sigurnosne rizike aplikacija ocjenjivanjem napada i slabih točaka u sigurnosti s obzirom na njihov tehnički i poslovni utjecaj. Prilikom procjene aplikacija, svaki od tih rizika demonstrira metodu napada neovisnu o tehnologiji ili platformi koja se koristi. Uz to OWASP pruža specifične upute o tome kako testirati, potvrditi i riješiti ranjivosti tj. ranjive elemente unutar aplikacije. OWASP top 10 se uglavnom usredotočuje na područja visokoga rizika, a ne rješavanju svih pitanja koja obuhvaćaju sigurnost web aplikacija. Međutim, OWASP zajednica daje neke osnovne za programere i sigurnosne revizore za učinkovito upravljanje sigurnošću web aplikacija:

- Vodič za testiranje: https://www.owasp.org/index.php/OWASP_Testing_Guide_v3_Table_of_Content
- Vodič za razvojne programere: www.owasp.org/index.php/Guide
- Vodič za reviziju programskoga koda: www.owasp.org/index.php/Category:OWASP_Code_Review_Project

3.4.3.1 Ključne značajke i prednosti

[23]Sljedeće su ključne značajke i prednosti OWASP-a:

- Testiranjem web aplikacija po metodologiji OWASP Top 10 sigurnosnih rizika umanjuje se rizik od najčešćih napada i osigurava **se** povjerljivost, integritet i dostupnost aplikacije.
- OWASP zajednica razvila je niz sigurnosnih alata koji se usredotočuju na automatizirane i ručne testove za ispitivanje sigurnosti web aplikacija. Neki od tih

[20] Kali Linux 2: Assuring Security by Penetration Testing Third Edition, Lee Allen, 2016: 57
[21] https://www.owasp.org/index.php/OWASP_Secure_Coding_Practices_-_Quick_Reference_Guide , 2.7.2018. 17:41:28
[22] https://www.owasp.org/images/7/72/OWASP_Top_10-2017_%28en%29.pdf.pdf , 2.7.2018. 18:02:41
[23] Kali Linux 2: Assuring Security by Penetration Testing Third Edition, Lee Allen, 2016: 58

alata su WebScarab, Wapiti, JBroFuzz i SQLiX, koji su također dostupni u Kali Linux operativnom sustavu.

- Kod razmatranja postupka sigurnosne procjene web infrastrukture, OWASP vodič za testiranje daje uvid u specifičnosti za određene tehnologije, na primjer, testiranju Oracle baze podataka pristupa se različito od MySQL baze podataka. Na taj način revizor si može olakšati odabir najprikladnijeg postupka testiranja.
- Potiče programere da se pridržavaju dobrih praksi i smjernica prilikom razvoja web aplikacija integriranjem sigurnosnih testova u svakoj fazi razvoja. Time se osigurava da aplikacija u produkcijskom okruženju bude robusna, bez pogrešaka i sigurna.
- Općenito prihvaćen od strane industrije. OWASP top 10 se također može uskladiti s drugim standardima procjene sigurnosti web aplikacija, čime se uz malo više napora može postići sukladnost sa više od jednog standarda istodobno.

3.4.4. Web Application Security Consortium Threat Classification (WASC-TC)

[24]Identificiranje sigurnosnih rizika web aplikacija zahtijeva temeljit i rigorozan postupak testiranja, koji se može pratiti tijekom životnog ciklusa razvoja aplikacije. Uz OWASP, WASC klasifikacija prijetnji je još jedan takav otvoreni standard za procjenu sigurnosti web aplikacija. Slično OWASP standardu, također klasificira brojne napade i slabosti web aplikacija, ali ih detaljnije obrađuje. Preduvjet za metodologije testiranja koje se bave identifikacijom i provjerom prijetnji koje obuhvaćaju web aplikacije je razumijevanje standardne terminologije koja se brzo može prilagoditi tehnologijskom okruženju. Ovo je segment gdje WASC-TC standard dolazi do izražaja.

Standard je prikazan kroz tri različita pogleda na sigurnost kako bi pomogao razvojnim programerima i sigurnosnim revizorima da lakše vizualiziraju sigurnosne prijetnje web aplikacija:

- Prikaz numeracije: kroz ovaj prikaz nastoji se pružiti uvid u osnove za napade i slabosti web aplikacija. Svaki od tih napada i slabosti definira se pojedinačno sa svojom sažetom definicijom, vrstom i programskim platformama na koje je napad

[24] Kali Linux 2: Assuring Security by Penetration Testing Third Edition, Lee Allen, 2016: 58

primjenjiv. Svakom od napada i slabosti web aplikacija dodijeljen je jedinstveni identifikator koji može biti koristan prilikom referenciranja tj. svakom od ukupno 49 napada i slabosti dodjeljuje se statički WASC-ID broj (1 do 49). Potrebno je imati na umu da se ovaj numerički prikaz ne fokusira na težinu rizika, već služi za potrebe referenciranja.

- Razvojni prikaz: Kroz razvojni prikaz sigurnost se sagledava iz perspektive razvojnog programera tako što se kombinira niz napada i slabosti u ranjivosti koje se mogu pojaviti u bilo kojoj od tri uzastopne faze razvoja web aplikacije. To može biti faza projektiranja/dizajniranja, implementacije ili puštanje aplikacije u produkcijsko okruženje. Ranjivosti u fazi dizajna javljaju su se kada aplikacije ne ispunjavaju sigurnosne zahtjeve tj. u početnoj fazi nisu prikupljanje sve relevantni podaci potrebni za stvaranje liste sigurnosnih zahtjeva. Ranjivosti u fazi implementacije javljaju se zbog nepridržavanja potrebnih načela i dobrih praksi prilikom razvoja aplikacije. Ranjivosti kod faze uvođenja aplikacije u produkcijsko okruženje rezultat su: pogrešne konfiguracije aplikacije, web poslužitelja i drugih vanjskih sustava. Cilj ovog prikaza je proširiti pogled razvojnih programera tj. potaknuti ih na integraciju WASC-TC standarda u redovni životni ciklus razvoja aplikacije.
- [25] Unakrsno referentni prikaz: Unakrsno referentni prikaz više standarda sigurnosti web aplikacija može pomoći revizorima i razvojnim programerima da mapiraju terminologiju iz jednog standarda s terminologijama iz drugih standarda. Uz malo više uloženog napora može se istovremeno ostvariti sukladnost sa više standarda. Međutim, svaki sigurnosni standard ima vlastite kriterije za procjenu aplikacija iz različitih aspekta. Dakle, svaki standard zahtijeva različite napore kako bi se identificirali rizici i njihova težina tj. potencijalna šteta koja nastaje ukoliko se rizik ostvari. Kroz ovaj prikaz [26]WASC-TC napadi i slabosti mapirani su pomoću OWASP top 10, [27]Mitra's Common Debt Enumeration (CWE), [28]Mitra's Common Perpetuum Enumeration and Classification (CAPEC) i [29]SANS-CWE top 25 popisa.

---

[25] http://projects.webappsec.org/w/page/13246975/Threat%20Classification%20Taxonomy%20Cross%20Reference%20View , 12.7.2018. 18:52:14

[26] http://projects.webappsec.org/w/page/13246978/Threat%20Classification , 13.7.2018. 15:48:59

[27] https://cwe.mitre.org/ , 13.7.2018. 15:50:05

[28] http://capec.mitre.org/ , 13.7.2018. 15:51:41

[29] https://www.sans.org/top25-software-errors , 13.7.2018. 15:53:49

3.4.4.1 Ključne značajke i prednosti

[30]Ključne značajke i prednosti WASC-TC:

- WASC-TC pruža temeljita znanja potrebna za procjenu radne okoline web aplikacija protiv najčešćih napada i slabosti.
- Napadi i slabosti koje obuhvaća WASC-TC mogu se koristiti za testiranje i provjeru bilo koje web aplikacije neovisno o platformi na kojoj aplikacija radi pomoću alata Kali Linux operativnog sustava.
- Pruža prošireni pogled na sigurnost web aplikacija kroz tri različita prikaza: numerički, razvojni i unakrsno referentni prikaz.
- Standard prihvaćen na razini industrije i njegova integracija se može naći u raznim open source i komercijalnim rješenjima.
- Može se uskladiti s drugim poznatim standardima sigurnosti web aplikacija, kao što su OWASP i SANS-CWE.

3.4.5. Penetration Testing Execution Standard (PTES)

[31]The Penetration Testing Execution Standard (PTES) stvorili su neki od najsjajnijih umova i stručnjaka u industriji penetracijskog testiranja. Standard se sastoji od sedam faza penetracijskog testiranja i može se koristiti za provedbu učinkovitog penetracijskog testiranja neovisno o okruženju.

[32]Sedam faza penetracijskog testiranja koje su detaljno opisane ovim standardom su kako slijedi :

1. Interakcija s klijentom prije angažmana
2. Prikupljanje podataka
3. Modeliranje prijetnji
4. Analiza ranjivosti
5. Iskorištavanje ranjivosti
6. Zadržavanje pristupa tj. održavanje pristupa kompromitiranom stroju

---

[30] Kali Linux 2: Assuring Security by Penetration Testing Third Edition, Lee Allen, 2016: 60
[31] Kali Linux 2: Assuring Security by Penetration Testing Third Edition, Lee Allen, 2016: 60
[32] http://www.pentest-standard.org/index.php/Main_Page, 15.7.2018. 19:12:07

7. Izrada izvještaja

Svaka od tih faza detaljno je opisana na web stranicama PTES zajednice zajedno s specifičnim mentalnim mapama (specifična forma dijagrama) koje detaljno opisuju potrebne korake za svaku fazu. Time se omogućuje prilagodba PTES standarda kako bi odgovarao zahtjevima testiranja okoline koja se ispituje.

3.4.5.1 Ključne značajke i prednosti

[33]Ključne značajke i prednosti PTES-a:

- Vrlo temeljiti model za penetracijsko testiranje koji pokriva tehničke i druge važne aspekte penetracijskog testiranja.
- Sadrži detaljne upute o tome kako izvršiti zadatke potrebne za precizno testiranje sigurnosti okruženja.
- Standard sastavljen od strane iskusnih stručnjaka u području penetracijskog testiranja koji svakodnevno obavljaju te zadatke.
- Obuhvaća najčešće korištene tehnologije, ali i one koji nisu uobičajene.
- Može se prilagoditi vlastitim potrebama za testiranjem.

3.5. Opći model penetracijskog testiranja

[34]Kali Linux je svestran operacijski sustav koji dolazi s nizom alata za penetracijsko testiranje i procjenu sigurnosti. Korištenje ovih alata bez odgovarajućeg modela može dovesti do neuspješnog testiranja i nezadovoljavajućih rezultata. Dakle, formalizacija sigurnosnog testiranja s dobro strukturiranim modelom izuzetno je važna je iz tehničke i upravljačke perspektive.

Opći model testiranja najčešće se primjenjuje kod black box i white box pristupa penetracijskog testiranja. Omogućuje osnovni pregled tipičnih faza kroz koje bi revizor/penetracijski tester trebao proći tokom testiranja. Bilo koji od ovih pristupa može se prilagoditi zadanom cilju procjene. Model se sastoji od koraka koje treba slijediti ovisno o

[33] Kali Linux 2: Assuring Security by Penetration Testing Third Edition, Lee Allen, 2016: 61
[34] Kali Linux 2: Assuring Security by Penetration Testing Third Edition, Lee Allen, 2016: 61

fazi ispitivanja kako bi se postigla uspješna procjena sigurnosti. Koraci kroz koje revizor treba proći tokom testiranja su sljedeći:

1. Definiranje područje primjene
2. Prikupljanje informacija
3. Otkrivanje ciljnog područja
4. Enumeracija
5. Mapiranje ranjivosti
6. Socijalni inženjering
7. Iskorištavanje ranjivosti
8. Eskalacija privilegija
9. Održavanje pristupa
10. Dokumentacija i izvješćivanje

Bez obzira na to primjenjuje li bilo koju kombinaciju navedenih koraka s black box ili white box pristupom, penetracijski tester mora prepoznati i odabrati najbolji strateški put u skladu s zadanim ciljem i svojim znanjem prije nego li testiranje započne. Ovaj opći pristup može se kombinirati s bilo kojom od postojećih metodologija penetracijskog testiranja i trebao bi služiti kao smjernica, a ne sveobuhvatno rješenje za penetracijsko testiranje.

3.5.1. Definiranje područje primjene

[35]Prije nego li se započne s evaluacijom tehničke sigurnosti, važno je temeljito se upoznati i razumjeti mrežno okruženje koje je predmet ispitivanja. Uz to potrebno je naglasiti da se područje primjene može definirati za jedinstveni entitet ili skup entiteta koji se daju revizoru. Prilikom donošenja odluka u ovoj fazi testiranja revizor treba razmotriti sljedeća pitanja:

- Što se testira?
- Kako provoditi postupak testiranja?
- Koji uvjeti moraju biti ispunjeni tijekom procesa testiranja?
- Što ograničava proces testiranja?
- Koliko će vremenski trajati testiranje?
- Koji se poslovni ciljevi postižu testiranjem?

[35] Kali Linux 2: Assuring Security by Penetration Testing Third Edition, Lee Allen, 2016: 62

Da bi uspješno proveo postupak penetracijskog testiranja, revizor mora biti dobro upoznat s tehnologijom koja se procjenjuje, njezinim osnovnim funkcionalnostima i interakcijom s mrežnim okruženjem. Stoga je iskustvo revizora od velikog značaja kod bilo koje vrste procjene sigurnosti.

3.5.2. Prikupljanje informacija

[36]Nakon što se definira područje testiranja slijedi faza izviđanja. Tijekom ove faze, revizor koristi brojne javno dostupne resurse kako bi naučio više o svom cilju. Te informacije mogu se pribaviti iz internetskih izvora kao što su:

- Forumi
- Oglasne ploče
- Interesne grupe (engl. Newsgroups)
- Članci
- Blogovi
- Društvene mreže
- Komercijalne ili nekomercijalne web stranice

Osim toga, podaci se mogu prikupiti i kroz različite tražilice, kao što su Google, Yahoo !, MSN Bing, Baidu itd. . Štoviše, revizor može koristiti i alate Kali Linuxa kako bi prikupio mrežne informacije o cilju. Ovi alati koriste razne metode za prikupljanje informacija putem DNS poslužitelja, mrežnih putanja, Whois baze podataka, adresa e-pošte, telefonskih brojeva, osobnih podataka i korisničkih računa. Kako opseg prikupljenih informacija raste tako raste i vjerojatnost uspješnog provođenja penetracijskog testiranja.

3.5.3. Otkrivanje ciljnog područja

[37]U ovoj faza glavni zadaci su identifikacija statusa mreže, operativnog sustava i stvaranje približne slike mrežne arhitekture što omogućuje prikaz međusobno povezanih tehnologija ili uređaja. Uz to doprinosi smanjenju potrebnih napora kod stvaranja liste servisa koji rade preko mreže. Koristeći napredne alate Kali Linuxa, moguće je dobiti uvid u hostove/računala priključena na mrežu, operacijski sustav/e koji se koriste na tim računalima i karakterizirati

[36] Kali Linux 2: Assuring Security by Penetration Testing Third Edition, Lee Allen, 2016: 63
[37] Kali Linux 2: Assuring Security by Penetration Testing Third Edition, Lee Allen, 2016: 64

svaki uređaj prema svojoj ulozi u mreži. Ovi alati koriste aktivne i pasivne tehnike detekcije mrežnih protokola kako bi izvukli korisne informacije.

3.5.4. Enumeracija

[38]Glavni zadatak ove faze je sastaviti listu otvorenih portova na ciljanom sustavu . Nakon što se identificiraju otvoreni portovi, moguće je napraviti asocijaciju pojedinog porta sa servisom tj. mrežnim protokolom koji koristi taj port . Portove je moguće skenirati korištenjem brojnih tehnika skeniranja kao što su otvoreno, polu otvoreno i skriveno skeniranje, a koje omogućavaju uvid u stanje portova čak i ako se računalo nalazi iza vatrozida ili sustava za otkrivanje upada (IDS). Asocijacija servisa sa pripadajućim otvorenim portovima pomaže u daljnjem istraživanju ranjivosti koje bi mogle postojati u mrežnoj infrastrukturi. Stoga moglo bi se reći da ova faza služi kao osnova za pronalaženje ranjivosti na različitim mrežnim uređajima koje potencijalno mogu dovesti do uspješnog penetriranja sustava. Revizor može koristiti automatizirane alate Kali Linuxu kako bi ostvario cilj ove faze.

3.5.5. Mapiranje ranjivosti

[39]U prethodnim fazama glavni zadatak bio je prikupljanje podataka o ciljanoj mreži. Nakon toga slijedi faza mapiranja ranjivosti u kojoj se identificiraju i analiziraju ranjivosti na temelju otkrivenih portova i servisa. Postupak se može provesti pomoću brojnih automatiziranih alata Kali linux-a za procjenu ranjivosti mreže i aplikacija. Može se provesti i ručno, ali vremenski traje značajno duže nego kod korištenja automatiziranih alata i zahtijeva stručno znanje. Međutim, kombiniranjem oba pristupa revizor si može stvoriti jasnu viziju i pažljivo ispitati svaku poznatu i/ili nepoznatu ranjivost koja postoji na mrežnim sustavima.

3.5.6. Socijalni inženjering

[40]Socijalni inženjering ima značajnu ulogu za revizora kada ne postoji alternativni ulaz u ciljnu mrežu. Za razliku od ranjivosti aplikacija socijalni inženjering koristi ljudski vektor

[38] Kali Linux 2: Assuring Security by Penetration Testing Third Edition, Lee Allen, 2016: 64
[39] Kali Linux 2: Assuring Security by Penetration Testing Third Edition, Lee Allen, 2016: 64
[40] Kali Linux 2: Assuring Security by Penetration Testing Third Edition, Lee Allen, 2016: 65

za upad/provalu u ciljni sustav, zavaravajući korisnika da izvrši zlonamjerni kod koji bi trebao dati „backdoor“ pristup revizoru te je na taj način moguće prodrijeti u ciljni sustav čak i ako u servisima/protokolima ne postoji ranjivost. Postoje razni oblici socijalnog inženjeringa. Na primjer to može biti osoba koja se pretvara da je mrežni administrator i putem telefona navodi korisnika da otkrije svoje podatke o korisničkom računu ili e-mail phishing prijevara u kojoj se korisnika navodi da klikne na poveznicu koja na prvi pogled izgleda kao legitimna web stranica međutim služi malicioznom korisniku da ukrade podatke ili e-mail u kojem se navodi korisnika da preuzme i pokrene u privitku zlonamjerni kod koji malicioznom korisniku omogućava udaljeni pristup kompromitiranom računalu. Netko tko imitira službeno osoblje tvrtke/organizacije kako bi dobio pristup fizičkoj prostoriji također se smatra socijalnim inženjeringom. Postoji čitav niz mogućnosti koje se mogu primijeniti za postizanje željenog cilja. Međutim potrebno je imati na umu da bi provođenje ove faze penetracijskog testiranja bilo uspješno potrebno je uložiti dodatno vrijeme za razumijevanje ljudske psihologije prije primjene bilo kakvog oblika obmane. Uz to također je važno u potpunosti razumjeti zakone države vezane za socijalno inženjerstvo prije provođenja ove faze.

3.5.7. Iskorištavanje ranjivosti

[41]Nakon pažljivog ispitivanja otkrivenih ranjivosti, moguće je odrediti točke upada u ciljni sustav na temelju dostupnih „eksploita“ (ranjivi dio software-a, greška u aplikaciji, niz naredbi koji mogu uzrokovati nepredviđeno ponašanje aplikacija, servisa ili računala) . Postoje razni repozitoriji koji sadrže već gotove exploite kao što je npr. [42]exploit-db i uz nešto dodatnog istraživanja moguće je izmijeniti postojeći eksploit kako bi ispravno radio na sustavu koji je predmet testiranja uz pretpostavku da na sustavu postoji ranjivost za koju već postoji exploit. Štoviše, revizor može kombinirati socijalni inženjering sa exploitim-a za koje je potrebna klijentska interakcija kako bi uz što manje napora preuzeo kontrolu nad ciljnim sustavom.

[41] Kali Linux 2: Assuring Security by Penetration Testing Third Edition, Lee Allen, 2016: 65
[42] https://www.exploit-db.com/ , 25.7.2018. 19:25:21

3.5.8. Eskalacija privilegija

[43]Kada revizor dobije pristup ciljnom sustavu penetracijsko testiranje se može proglasiti uspješnim. Ovisno o privilegijama korisničkog računa preko kojega je revizor pristupio sustavu moguće je izvoditi određene aktivnosti. Međutim privilegije je moguće eskalirati pomoću lokalnih eksploita koji su predviđeni za rad u okruženju sustava, a ukoliko se uspješno izvrše mogu podići razinu privilegija na razinu administratora ili [44]LocalSystem korisničkog računa . Ukoliko revizor uspije dobiti dovoljno visoku razinu privilegija otvara mu se mogućnost daljnjih napada na lokalne mrežne sustave ukoliko u ugovoru s klijentom nisu definirana ograničenja za pojedine sustave organizacije

3.5.9. Održavanje pristupa

[45]Ponekad klijent može zatražiti od revizora da zadrži pristup sustavu na određeno vremensko razdoblje. Na taj način ukoliko je potrebno može se ponovno dokazati aktivnost nelegitimnog pristupa sustavu bez potrebe za ponavljanjem postupka penetracijskog testiranja. Time se štedi vrijeme, resursi i troškovi koje je potrebno izdvojiti za ponavljanje penetracijskog testiranja tj. za simulaciju nelegitimnog pristupa sustavu u sigurnosne svrhe. Korištenjem protokola za tuneliranje, proxy-a, ili „end-to-end“ veze moguće je uspostaviti „backdoor“ pristup koji može poslužiti revizoru da zadrži pristup ciljnom sustavu dokle god je to potrebno. Ovakav pristup sustavu pruža jasan uvid u to kako napadač potajno može održavati pristup sustavu.

3.5.10. Dokumentacija i izvješćivanje

[46]Ova faza dokumentiranja, izvještavanja i prikazivanja pronađenih, provjerenih i iskorištenih ranjivosti predstavlja zaključak penetracijskog testiranja. Iz etičke perspektive ima značajnu važnost zato jer rukovoditeljskom i tehničkom timu organizacije daje u uvid u metode prodiranja u sustav i smjernice za zatvaranje sigurnosnih rupa. Izvješća se izrađuju u različitim oblicima ovisno o tome za koja su nadležna tijela u organizaciji namijenjena kako bi pomogla poslovnom i tehničkom osoblju u razumijevanju i analizi slabih točaka koje

---

[43] Kali Linux 2: Assuring Security by Penetration Testing Third Edition, Lee Allen, 2016: 65
[44] https://docs.microsoft.com/en-us/windows/desktop/services/localsystem-account , 27.7.2018. 17:01:16
[45] Kali Linux 2: Assuring Security by Penetration Testing Third Edition, Lee Allen, 2016: 66
[46] Kali Linux 2: Assuring Security by Penetration Testing Third Edition, Lee Allen, 2016: 66

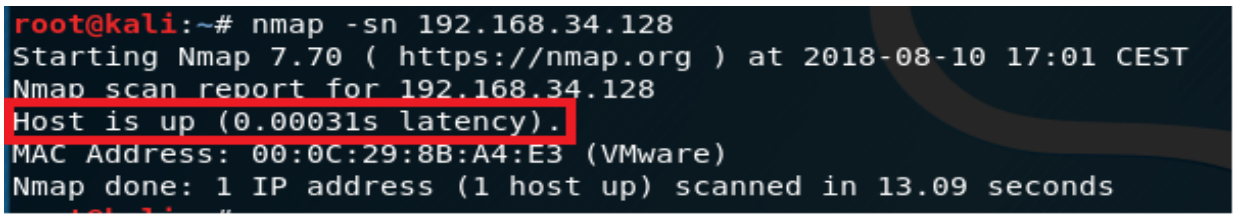
postoje u njihovoj IT infrastrukturi. Osim toga, ova izvješća mogu poslužiti kod usporedbe integriteta ciljanog sustava prije i poslije procesa penetracijskog testiranja.

4. Izviđanje

Izviđanje je prvi i osnovni korak kod svakog penetracijskoga testiranja. Čim je opseg prikupljenih informacija o cilju veći tako rastu i mogućnosti vezane za pronalaženje i iskorištavanje ranjivosti. Kod penetracijskog testiranja web aplikacija u ovoj fazi cilj je prikupiti informacije o aplikacijama, bazama podataka, korisnicima, serveru te kako se odvija interakcija između korisnika i aplikacija.

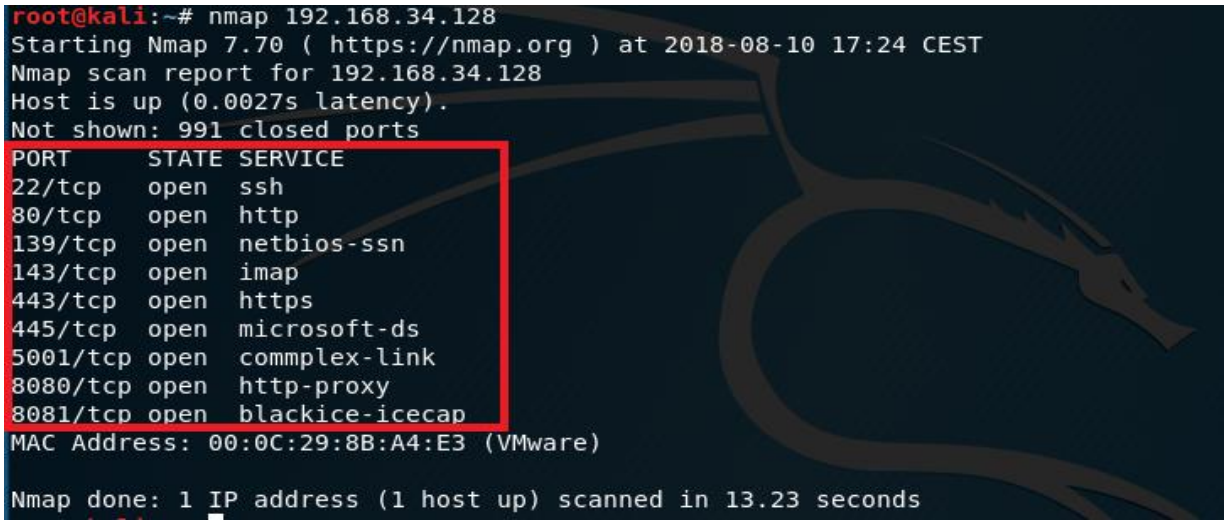
4.1. Nmap

Nmap je jedan od najpoznatijih alata korištenih u ovoj fazi. Koristi se za otkrivanje statusa servera koji se cilja, skeniranje otvorenih TCP i UDP portova, otkrivanje vatrozida, otkrivanje servisa i njihovih verzija itd. . Prvi korak bio bi provjera funkcionalnosti servera tj. odgovara li na zahtjeve, a to se postiže naredbom nmap -sn <ip adresa servera/ime servera> , u ovom slučaju 192.168.34.128.



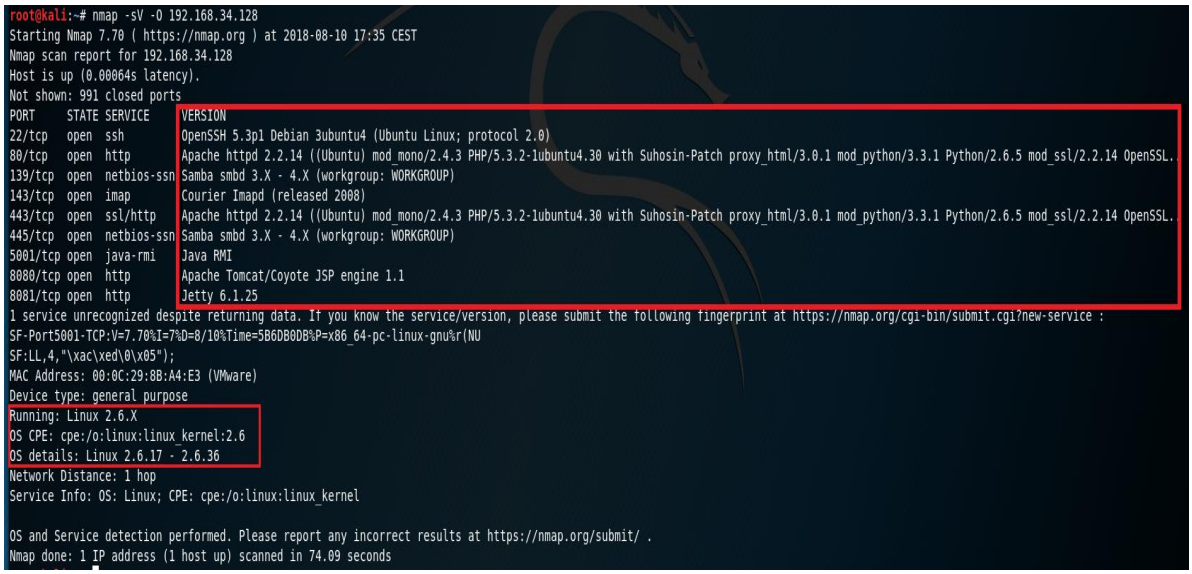
Slika 4.1 Testiranje statusa servera

Iz slike 4.1 može se vidjeti da je server trenutno funkcionalan. Ako se ustanovi da je server u funkciji slijedi idući korak, a to je ispitivanje postoje li otvoreni portovi na serveru. Navedeno se postiže jednostavnom naredbom npam <ip adresa servera/ime servera>.



Slika 4.2 Identifikacija otvorenih portova na serveru

Nakon što su identificirani otvoreni portovi i njihovi pripadajući servisi idući logičan korak je pokušati otkriti verzije servisa i operacijski sustav na kojem se izvode ti servisi koristeći naredbu nmap –sV –O <ip adresa servera/ime servera>.



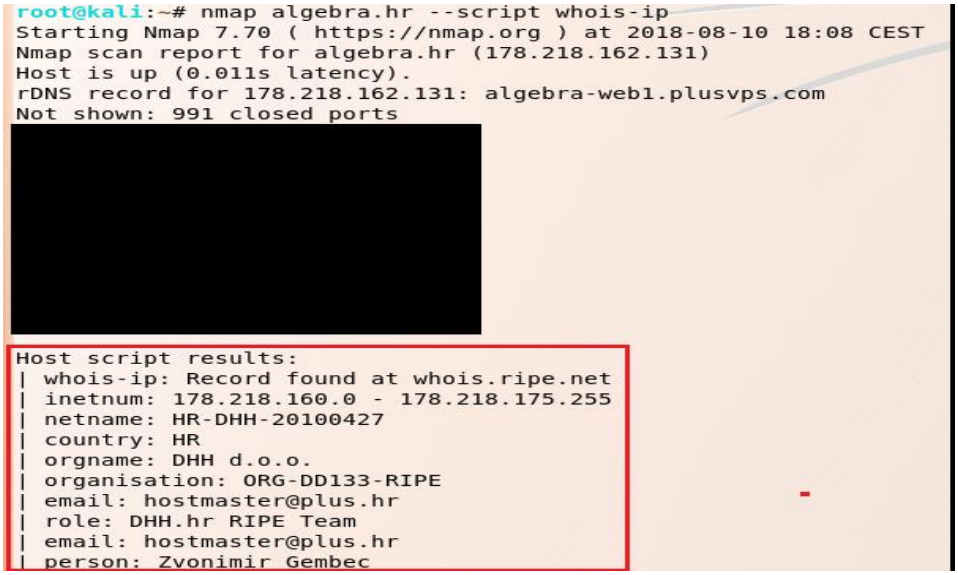
Slika 4.3 Identifikacija verzije servisa i operacijskog sustava

Iz slike 4.3 je vidljivo da server radi na Linux operacijskom sustavu, verzija kernela je 2.6.x, dok je npr. verzija Apache-a 2.2.14, php-a 5.3.2 itd. .

Uz to moguće je koristiti i brojne gotove skripte iz [47]nmap repozitorija. Na Slici 4.4 nalazi se primjer korištenja skripte koja šalje upit na WHOIS regionalne internetske registre i

[47] https://nmap.org/nsedoc/scripts/ , 9.8.2018. 18:15:23

pokušava dohvatiti informacije kao što su npr. raspon dodijeljenih ip adresa, ip adresa servera, kontakt podaci osobe zadužene za domenu itd. .



Slika 4.4 primjer korištenja [48]whois-ip skripte za dobivanje informacija o domeni algebra.hr

4.2. Identifikacija Vatrozida za zaštitu web aplikacija

Vatrozid za zaštitu web aplikacija (WAF) je uređaj ili softver koji provjerava mrežne pakete obično na temelju potpisa ili regularnih izraza koji dolaze prema web serveru kako bi se identificiralo i blokiralo zlonamjerne pakete.

Revizor se može suočiti sa nizom problema tijekom penetracijskog testiranja ako vatrozid neopaženo blokira zahtjeve prema serveru ili brani pristup s IP adrese koju koristi revizor. Pri provođenju penetracijskog testa, faza izviđanja obavezno mora uključivati proces otkrivanja i identifikacije vatrozida, IDS/IPS sustava, iz razloga kako bi se poduzele potrebne mjere da se testiranje odvija neometano.

Nmap repozitorij sadrži nekoliko skripti koje mogu poslužiti za testiranje prisutnosti vatrozida. Prvi primjer je skripta [49]http-waf-detect.

[48] https://nmap.org/nsedoc/scripts/whois-ip.html , 10.8.2018. 14:01:03
[49] https://nmap.org/nsedoc/scripts/http-waf-detect.html , 10.8.2018. 14:06:56

```
root@kali:~# nmap -p 80,443 --script=http-waf-detect 192.168.34.128
Starting Nmap 7.70 ( https://nmap.org ) at 2018-08-10 18:46 CEST
Nmap scan report for 192.168.34.128
Host is up (0.00028s latency).

PORT    STATE SERVICE
80/tcp  open  http
443/tcp open  https
MAC Address: 00:0C:29:8B:A4:E3 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 13.39 seconds
```

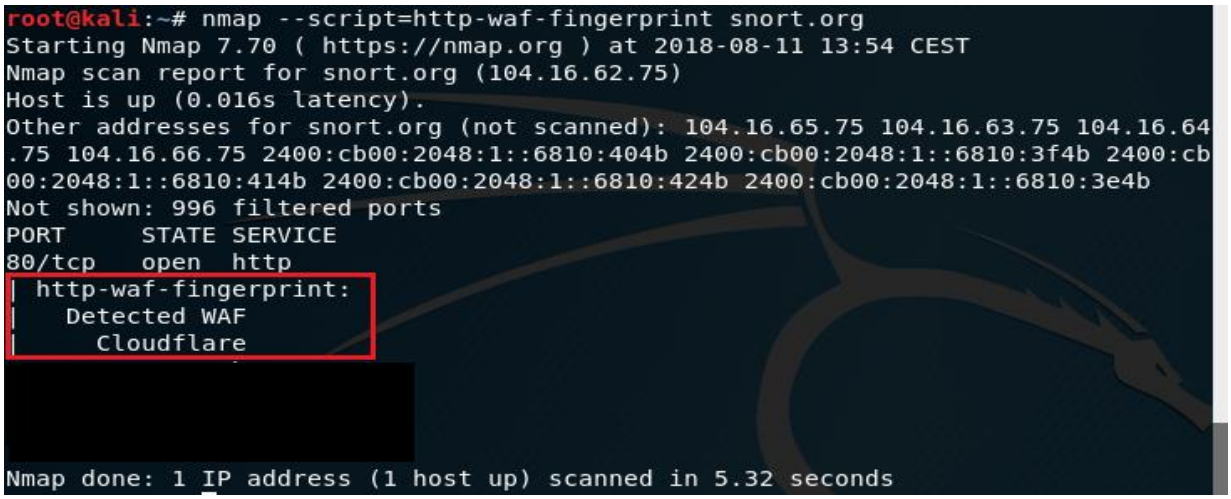
Slika 4.5 Testiranje prisutnosti vatrozida, IDS/IPS sustava na serveru koristeći nmap skriptu http-waf-detect

Iz Slike 4.5 može se zaključiti da se na serveru ne nalazi vatrozid, IDS/IPS sustav. Međutim ako se ista komanda izvrši prema serveru koji ima prisutan vatrozid, IDS/IPS sustav, rezultat je**:**



Slika 4.6 Otkrivanje prisutnosti vatrozida, IDS/IPS sustava na serveru

Postoji još jedna skripta u Nmap-u koja može pomoći kod identifikacije vatrozida, IDS/IPS sustava na serveru, a to je skripta [50]http-waf-fingerprint .

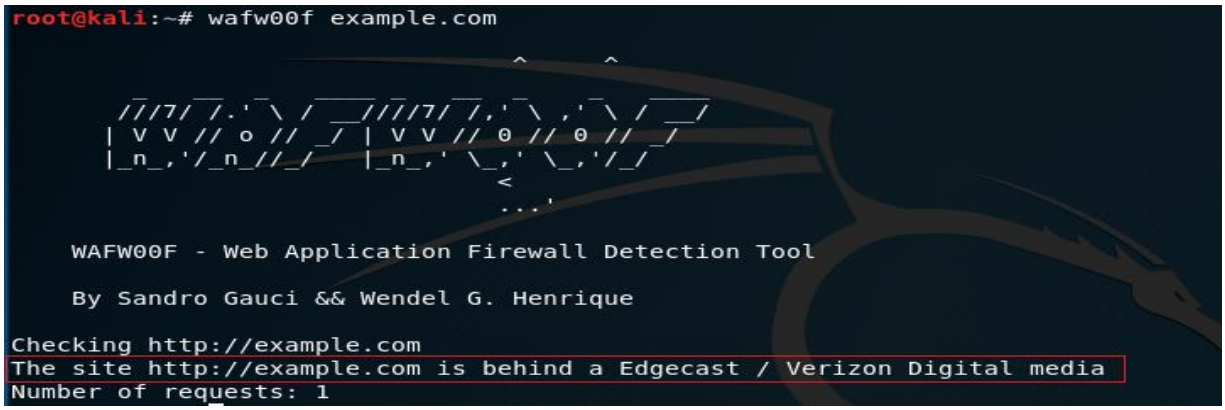


[50] https://nmap.org/nsedoc/scripts/http-waf-fingerprint.html , 10.8.2018. 14:07:54

Slika 4.7 Identifikacija vatrozida na serveru

Za razliku od http-waf-detect skripte skripta http-waf-fingerprint daje uvid o kojem se vatrozidu, IDS/IPS sustavu radi. Iz slike 4.7 je vidljivo da se u ovom primjeru radi u vatrozidu tvrtke [51]CloudFlare.

Još jedan popularan alat za identifikaciju vatrozida je wafw00f, a primjenjuje se na način wafw00f <ip adresa servera/hostname> .



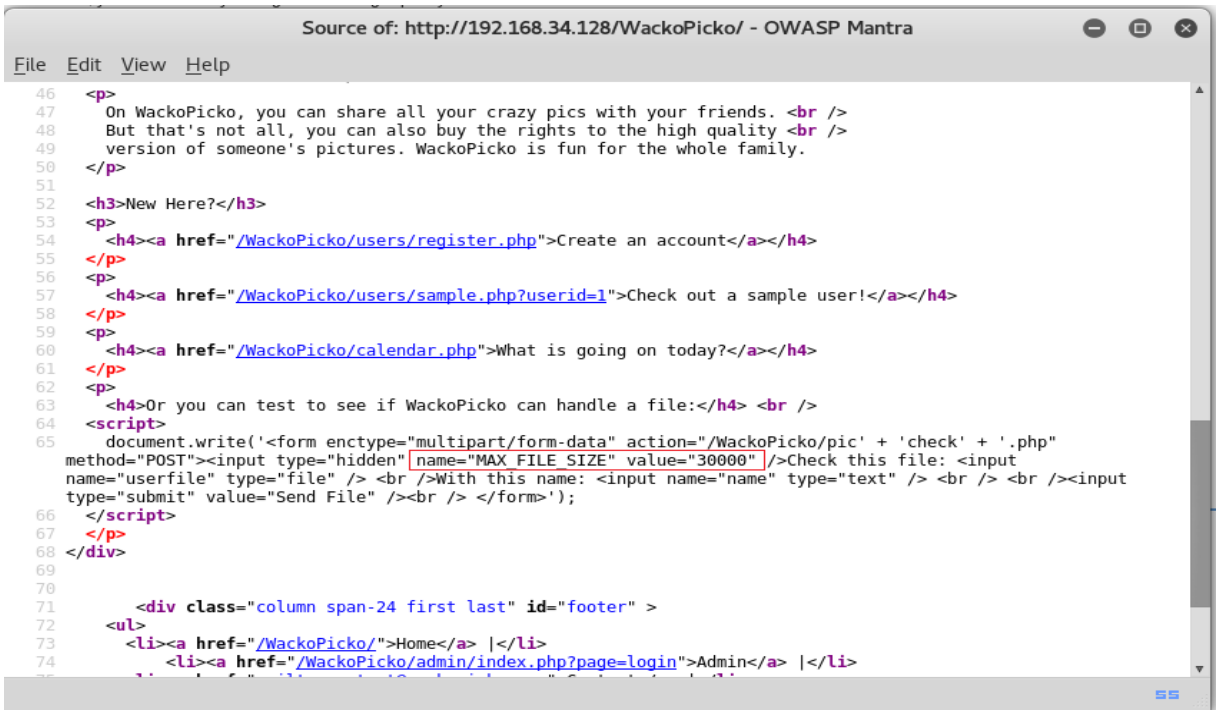
Slika 4.8 Identifikacija vatrozida uz pomoć alata wafw00f

Važno ja napomenuti da skripte http-wat-detect i http-wat-fingerprint nisu bile u mogućnosti identificirati vatrozid za domenu [52]example.com dok je alat wafw00f uspješno otkrio da se radi o [53]verizon-ovom vatrozidu.

4.3. Analiza izvornog koda web stranice

Analizom izvornog koda web stranice revizor može dobiti uvid u programsku logiku tj. kako otprilike aplikacija radi, a samim time i potencijalne ranjivosti koje postoje u web aplikaciji. Stoga analiza izvornog koda web stranice predstavlja važan korak u fazi izviđanja. Za potrebe testiranja u ovom koraku koristit će se WackoPicko web aplikacija koja je dio [54]OWASP Broken Web Applications Project-a tj. virtualnog računala na kojem se nalazi niz ranjivih web aplikacija za potrebe prakticiranja penetracijskog testiranja.

[51] https://www.cloudflare.com/ , 10.8.2018. 14:13:02
[52] http://example.com/ . 10.8.2018. 14:30:18
[53] https://www.verizondigitalmedia.com/platform/protect/ , 10.8.2018. 14:32:20
[54] https://www.owasp.org/index.php/OWASP_Broken_Web_Applications_Project , 10.8.2018. 14:50:42



Slika 4.9 Izvorni kod web stranice WackoPicko

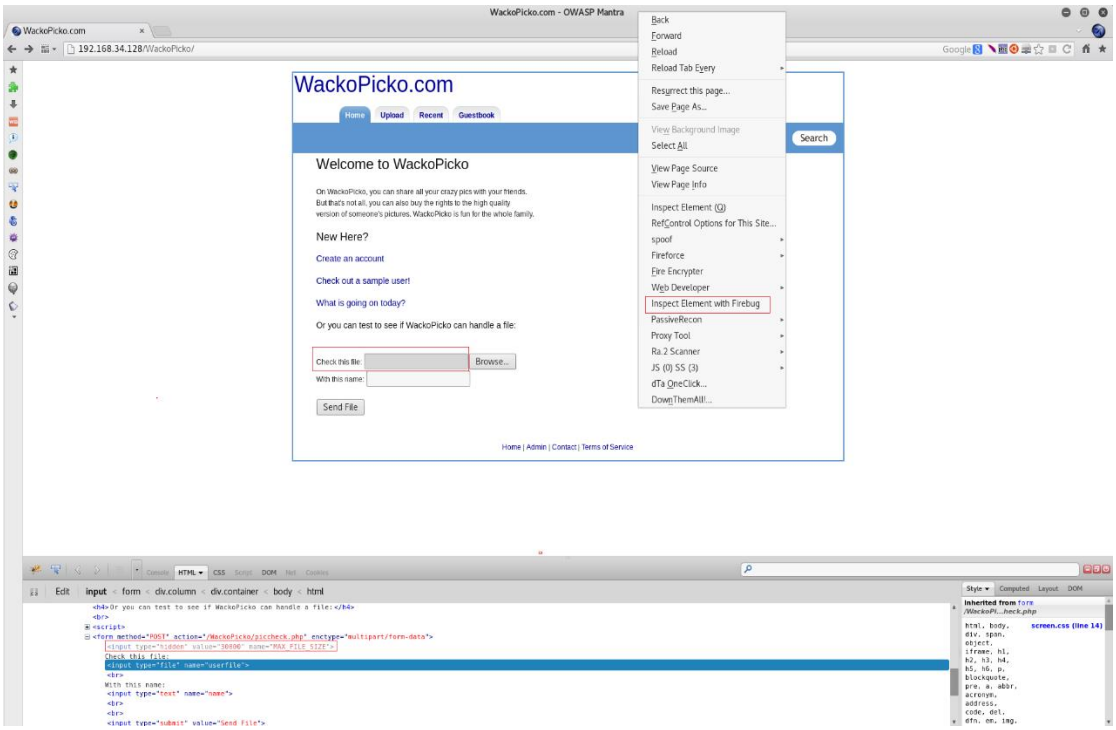
Analizom izvornoga koda mogu se otkriti vanjske datoteke i poveznice koje stranica koristi . Uz to kao što se može vidjeti na Slici 4.9 ova web stranica ima skrivena polja za unos kao što je npr. MAX_FILE_SIZE. Ovo polje određuje maksimalnu veličinu datoteke koja se može prenijeti na server. Znači, ako se izmijeni vrijednost tog polja, moglo bi se prenijeti veću datoteku nego što aplikacija očekuje, a to može predstavljati sigurnosni problem.

4.4. Firebug alat za analizu i modifikaciju elemenata web stranice

Firebug je dodatak za web preglednike koji omogućuje analizu unutarnjih dijelova web stranice, kao što su tablice, CSS klase, okviri itd. . Također ima mogućnost prikaza DOM objekata, šifre pogrešaka i komunikaciju između preglednika i servera.

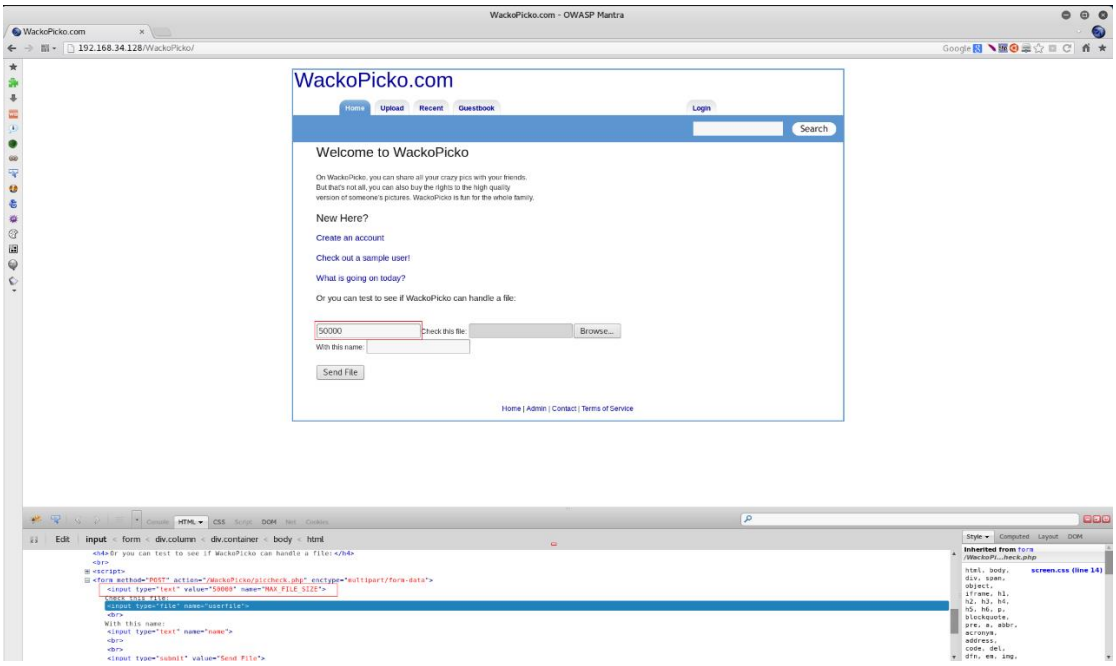
U prethodnom primjeru prilikom analize izvornog HTML koda web stranice pronađeno je skriveno polje za unos koje ima definiranu zadanu vrijednost za maksimalnu dopuštenu veličinu datoteke. U ovom primjeru bi ti će demonstrirano kako manipulirati vrijednostima skrivenih polja. Za potrebe testiranja koristit će se OWASP-Mantra web preglednik.

Prvi korak bi bio desni klik u polje „Check this file“ i u OWASP-Mantra web pregledniku odabrati opciju Inspect Element with Firebug kao što je to prikazano na slici 4.10 .



Slika 4.10 Analiza polja web stranice koristeći firebug dodatak za preglednike

Ukoliko promijenimo vrijednost parametra type="hidden" → type="text" i value="30000" → value="50000" na početnoj stranici pojavit će se pored polja „check this file" pojavit će se tekstualno polje sa vrijednosti 50000 što znači da je uspješno promijenjeno ograničenje veličine datoteke koja se može prenijeti na server.

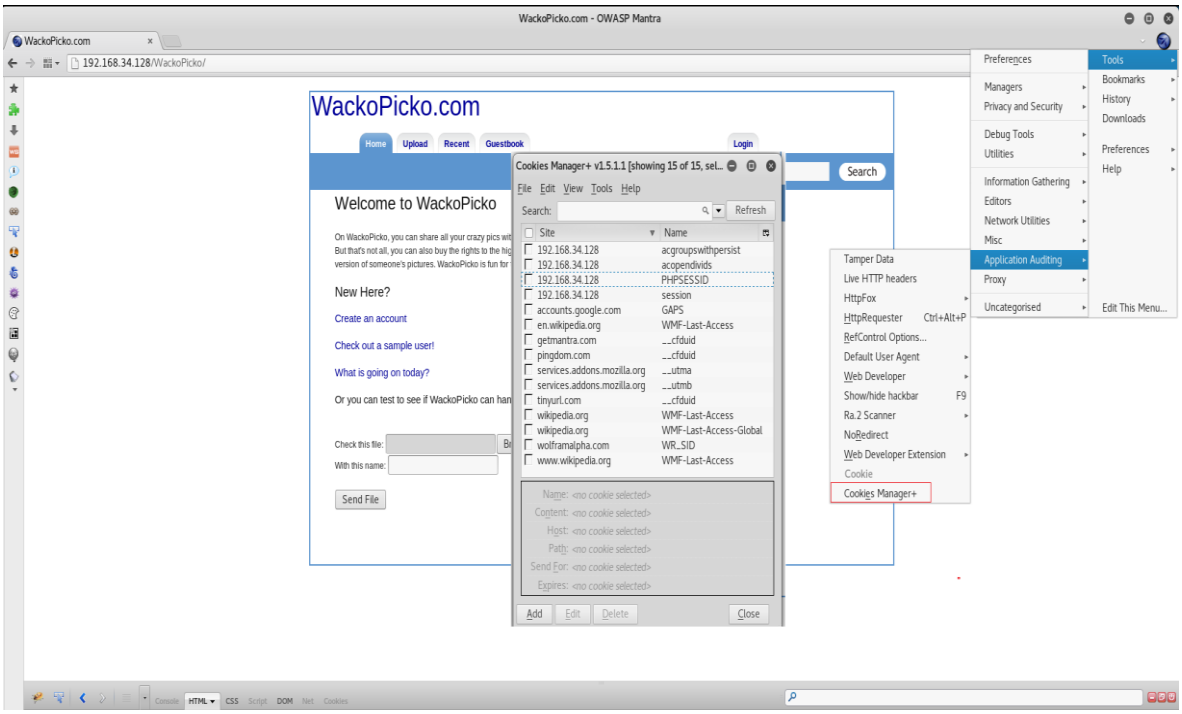


Slika 4.11 Web stranica nakon izmjene parametra

4.5. Dohvaćanje i modifikacija cookie-a

[55]Kolačići su dijelovi informacija koje web server šalje klijentu (pregledniku) za lokalnu pohranu određenih informacija, vezanih za određenog korisnika. U modernim web aplikacijama, kolačići se koriste za pohranu korisničkih podataka, kao što su konfiguracija tema, raspored objekta na web stranici, prethodne aktivnosti i (što je najvažnije za penetracijskog testera) identifikatore sesije.

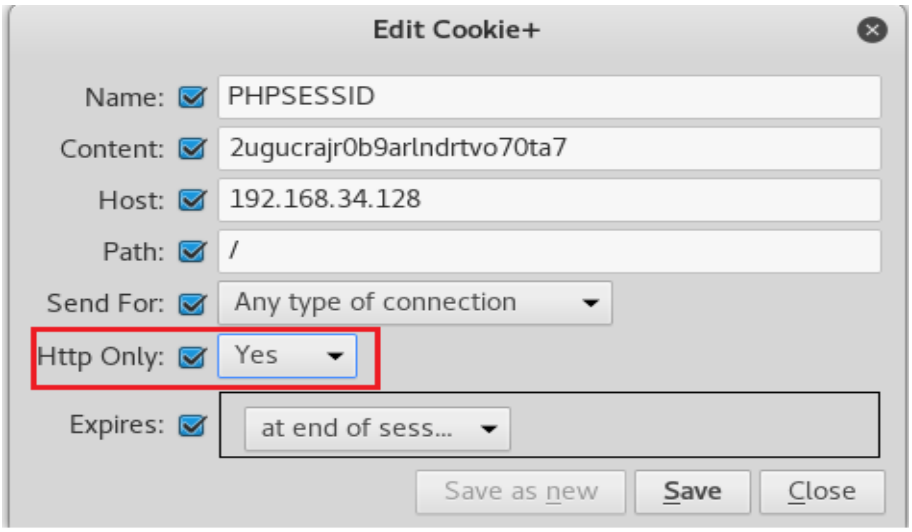
OWASP-Mantra sadrži razne alate pomoću kojih je moguće vidjeli vrijednost kolačića, kako se pohranjuju, i kako se mogu mijenjati. Jedan od popularnijih alata koji omogućuje navedene radnje navedeno je Cookies Manager +



Slika 4.12 Cookies Manager +

Na slici 4.12 mogu se vidjeti svi trenutno pohranjeni kolačići pohranjene i web stranice kojima pripadaju. Uz to moguće je mijenjati vrijednost kolačića, brisati postojeće kolačiće i dodavati nove . Na primjer ukoliko se u kolačiću promijeni vrijednost parametra Http Only: No → Yes web preglednik neće dopustiti pristup kolačiću od strane klijentskih skripti.

[55] Kali Linux Web Penetration Testing Cookbook, Gilberto Najera-Gutierrez, 2016: 38



Slika 4.13 Primjer izmjene Http only parametra u kolačiću

4.6. Otkrivanje skrivenih web stranica i direktorija uz pomoć robots.txt

[56]Tijekom procesa penetracijskog testiranja revizor mora utvrditi postoji li web stranica ili direktorij na web mjestu koji nisu prikazani običnom korisniku. Na primjer to može biti stranica za prijavu u intranet ili stranica za administraciju sustava za upravljanje sadržajem (CMS) itd. . Otkrivanjem takvih web stranica revizor može dobiti uvid o aplikacijama i infrastrukturi na kojoj rade te aplikacije .

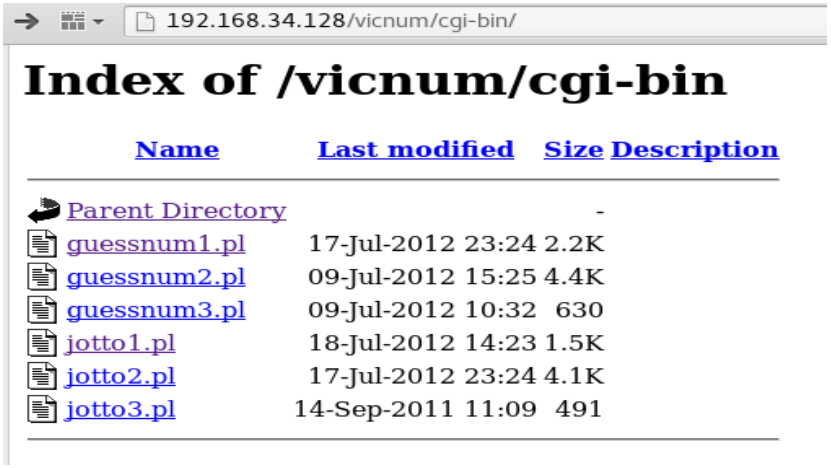
Robots.txt može pružiti informacije o datotekama i direktorijima za koje ne postoji poveznica u glavnoj aplikaciji. Za potrebe ovog testiranja koristit će se Vicinum web aplikacija koja je dio OWASP Broken Web Applications Project-a.



Slika 4.14 Otkrivanje skrivenih direktorija pomoću robots.txt

Robots.txt datoteka daje naputak tražilicama da indeksiranje direktorija jotto i cgi-bin nije dozvoljeno svakome. Međutim, to ne znači da se oni ne mogu pregledavati.

[56] Kali Linux Web Penetration Testing Cookbook, Gilberto Najera-Gutierrez, 2016: 40



Slika 4.15 Sadržaj direktorija cgi-bin



Slika 4.16 Sadržaj direktorija jotto

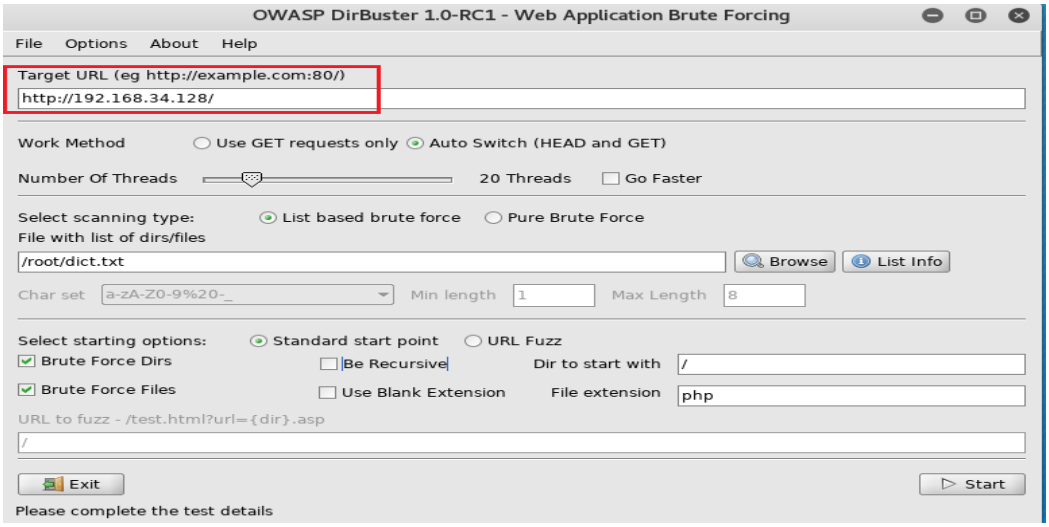


Slika 4.17 Lista mogućih odgovora za igru jotto

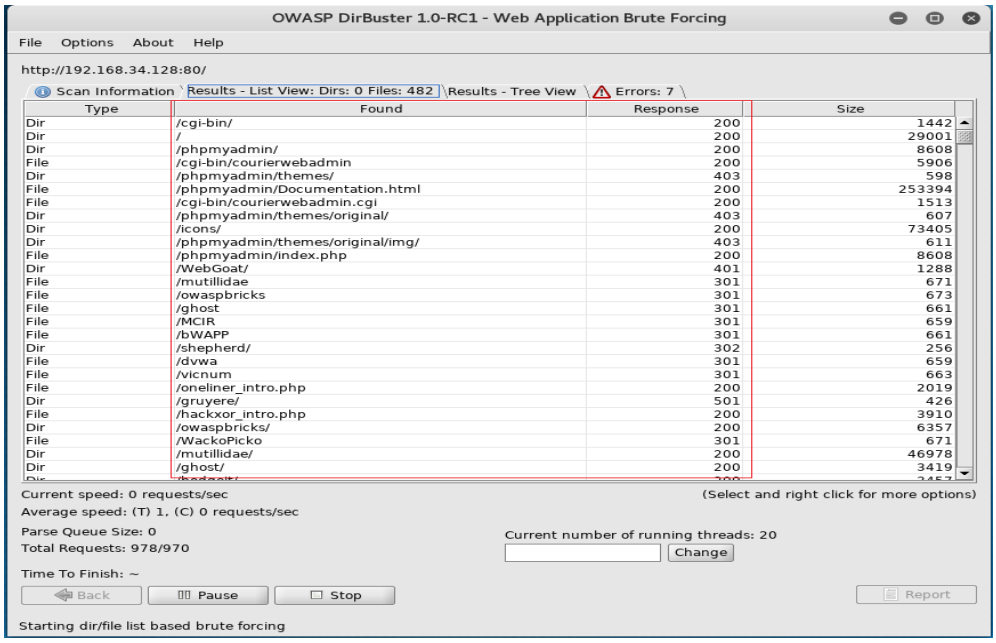
Jotto je igra u kojoj je cilj pogoditi riječ koja se sastoji od pet znakova, a na slici 18. može se vidjeti lista mogućih odgovora koje aplikacija očekuje.

4.7. DirBuster-alat za pronalaženje datoteka i direktorija na web serveru

DirBuster je alat kali linux-a za pronalaženje postojećih datoteka i direktorija na web serveru. Ovaj alat radi na sljedeći način: prvo je potrebno odrediti cilj tj. web server za koji se radi pretraga prisutnih datoteka i direktorija. Zatim je potrebno odrediti hoće li alat za pretragu direktorija koristiti rječnik sa ključnim pojmovima koje je definirao korisnik (List Based Brute force) ili će alat sam generirati rječnik (Pure Brute Force) . Nakon toga potrebno je još odabrati dodatne opcije ovisno o tome što se nastoji postići. Kada je sve konfigurirano moguće je započeti pretragu.



Slika 4.18 DirBuster početni zaslon



Slika 4.19 Pronađene datoteke i direktoriji web servera

Kada alat završi sa pretragom datoteka i direktorija moguće je otići na karticu rezultati koja daje uvid u pronađene datoteke i direktorije web servera. Kako bi utvrdio da li datoteka postoji ili ne, DirBuster koristi kodove koje server povratno šalje kao odgovor na zahtjev .[57]Neki od najčešćih kodova su:

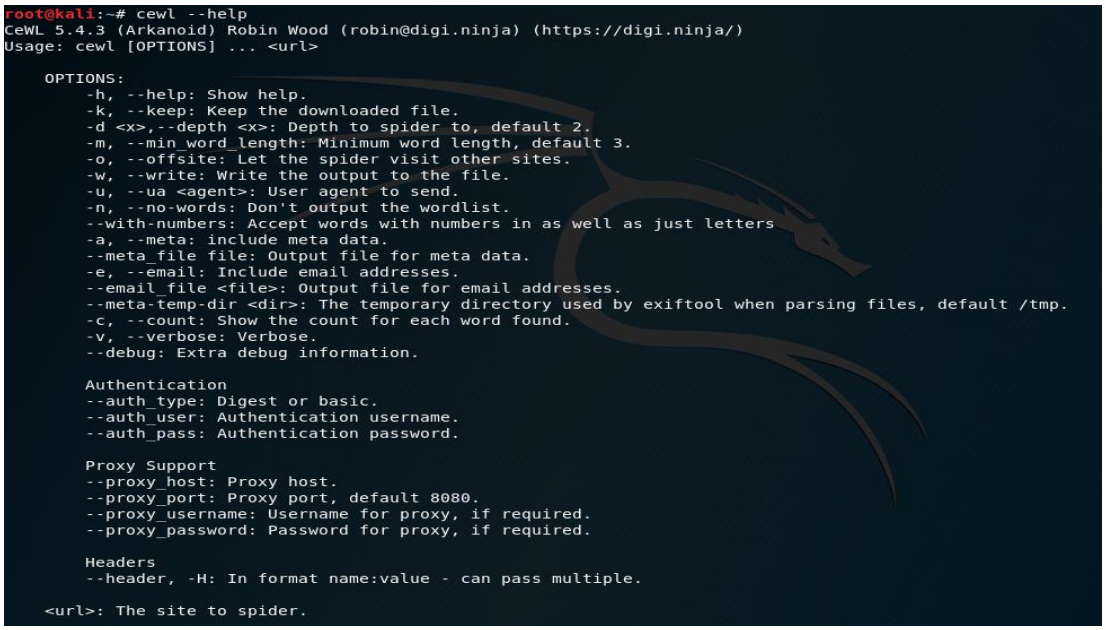
- 200. OK: Datoteka postoji i korisnik ju može pročitati.
- 404. Datoteka nije pronađena: datoteka ne postoji na serveru.
- 301. Trajno premješteno: Preusmjeravanje na određeni URL.
- 401. Neovlašteno: Za pristup ovoj datoteci potrebna je provjera autentičnosti.
- 403. Zabranjeno: Zahtjev je valjan, ali poslužitelj odbija odgovoriti na zahtjev.

Iz slike 4.19 moguće je uočiti da na web serveru postoji PhpMyAdmin direktorij što ukazuje na prisutnost web bazirane MySQL aplikacije za administriranje baza podataka koja može sadržavati relevantne informacije o korisnicima.

4.8. CeWL-alat za profiliranje lozinki

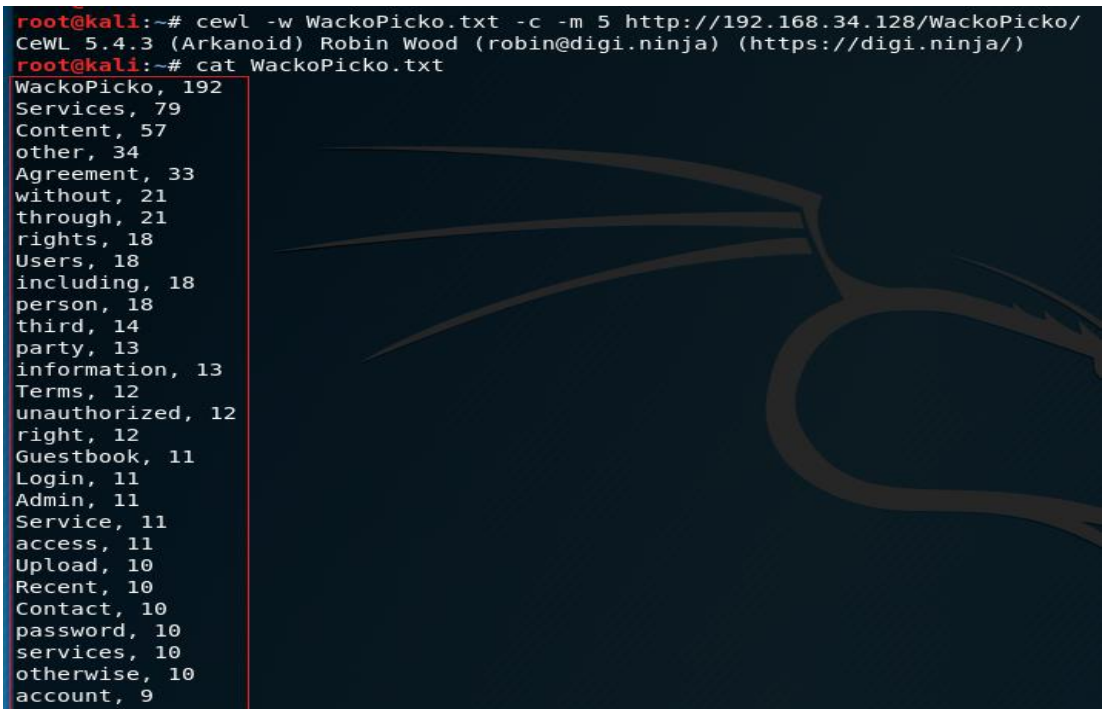
U svakom penetracijskom testu tijekom faze izviđanja potrebno je provesti profiliranje u kojem se analiziraju aplikacije, odjeli ili nazivi procesa i ostale riječi koje koristi ciljana organizacija. Taj proces može pomoći revizoru odrediti kombinacije koje će se vjerojatno koristiti npr. za korisničko ime ili lozinku zaposlenika/osoblja organizacije.

[57] Kali Linux Web Penetration Testing Cookbook, Gilberto Najera-Gutierrez, 2016: 63



Slika 4.20 Lista opcija alata CeWL

Za potrebe testiranja ovog alata koristit će se WackoPicko web aplikacija kako bi se dobila lista riječi koje se pojavljuju minimalno jedanput na web aplikaciji. Parametri pretrage su sljedeći: traže se riječi s minimalnom duljinom od pet znakova, kraj svake riječi biti će prikazan broj ponavljanja i rezultati će se spremiti u tekstualnu datoteku WackoPicko.txt. Navedeno se postiže naredbom: cewl -w WackoPicko.txt -c -m 5 http://192.168.34.128/WackoPicko/

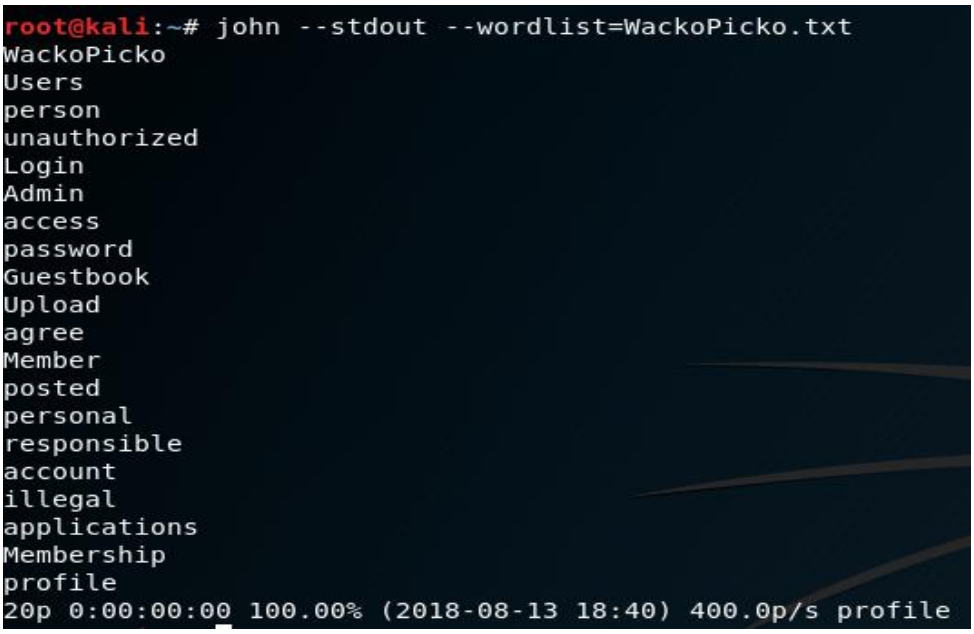


Slika 4.21 Lista riječi web aplikacije WackoPicko

Međutim nije dovoljno samo izraditi listu riječi već ju je potrebno i filtrirati kako bi se odbacile riječi koje se često javljaju, ali je mala vjerojatnost da će se koristiti kao lozinke.

4.9. JohnTheRipper

John the Ripper je jedan od najpopularnijih i najčešće korištenih alata među penetracijskim testerima i hakerima kada je riječ o probijanju lozinki . Alat Ima puno mogućnosti od kojih su najznačajnije automatsko prepoznavanje najčešće korištenih algoritama za enkripciju i hashing, sposobnost korištenja rječnika i brute force napada. Uz to omogućuje generiranje vlastitih rječnika na temelju gotovih lista riječi te koristi permutacije i razna pravila za oblikovanje riječi kako bi proširio rječnik te na taj način stvorio bogatiji rječnik koji će se koristit prilikom pokušaja pogađanja korisnikovih lozinki na web login portalima. Ova posljednja navedena značajka će se koristiti u ovom koraku za generiranje opsežnog rječnika na temelju vrlo jednostavne liste riječi koja je stvorena u prethodnom koraku pomoću CeWL alata .



Slika 4.22 Lista riječi koja će se koristit za generiranje kompleksnog rječnika

Koristeći naredbu john --stdout --wordlist=WackoPicko.txt --rules > dict_WackoPicko.txt od navedene liste riječi (20 riječi) alat će stvoriti rječnik od 999 riječi kroz zamjenu redoslijeda znakova, malih i velikih slova, dodavanje sufiksa i prefiksa te zamjenu slova sa brojevima i simbolima/specijalnim znakovima.

```
root@kali:~# john --stdout --wordlist=WackoPicko.txt --rules > dict_WackoPicko.txt
Press 'q' or Ctrl-C to abort, almost any other key for status
999p 0:00:00:00 100.00% (2018-08-13 18:44) 24975p/s Profiling
```

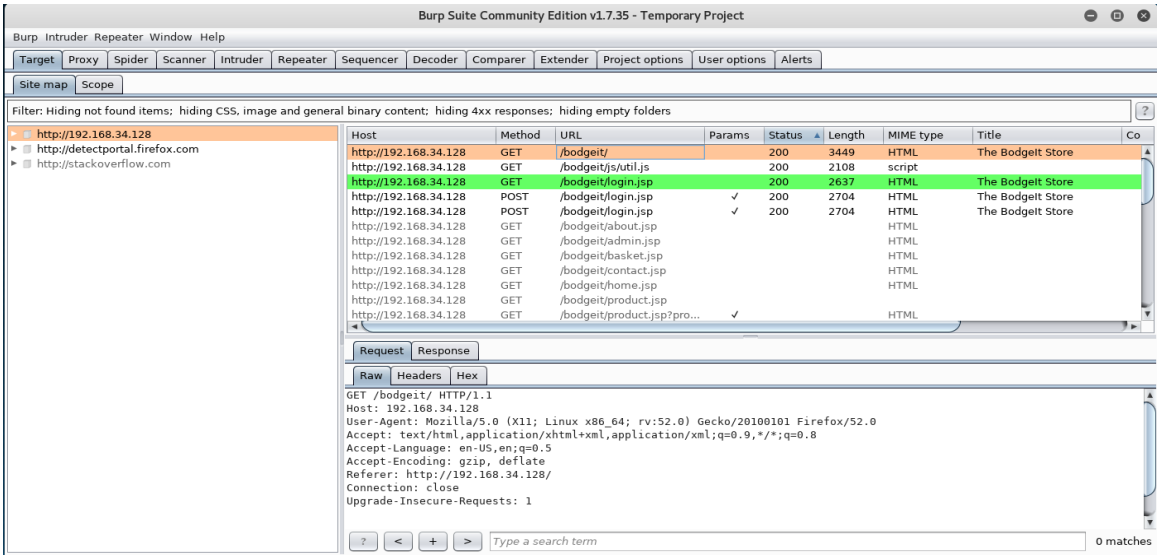
Slika 4.23 Novostvoreni rječnik na temelju zadane liste riječi

Rezultat naredbe je novostvoreni rječnik od 999 riječi koji će se u kasnijim fazama testiranja koristiti za pokušaj pogađanja lozinke za prijavu preko login stranica raznih web aplikacija.

5. Indeksiranje web stranica i direktorija

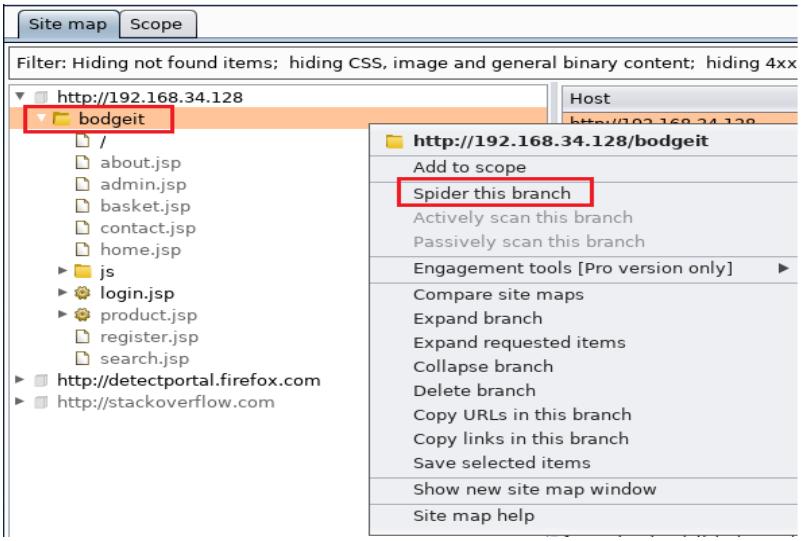
Obavezni korak u svakoj fazi izviđanja penetracijskog testiranja kojeg revizor ne smije preskočiti je pregledati svaku poveznicu koja se nalazi unutar web stranice i vođenje evidencije o svakoj datoteci koju te poveznice prikazuju. Kali linux sadrži Web crawler i Web spider alate koji pomažu automatizirati i ubrzati ovaj zadatak. Princip rada je sljedeći: alati pregledavaju web stranicu pritom slijedeći sve veze i reference na vanjske datoteke, ponekad ispunjavaju obrasce i šalju ih serveru dok pritom lokalno pohranjuju sve zahtjeve upućene prema serveru i odgovore servera, što revizoru kasnije omogućava izvan mrežnu analizu. Jedan od najčešće korištenih alata za crawling i spidering web stranica je Burb Suite koji će se koristiti u ovoj fazi za demonstraciju na web aplikaciji Bodgeit koja je dio OWASP Broken Web Applications Project-a.

Prvi korak je preko web preglednika posjetiti stranicu na kojoj se nalazi web aplikacija Bodgeit http://192.168.34.128/bodgeit/.



Slika 5.1 Analiza zahtjeva sa Burp suite-om

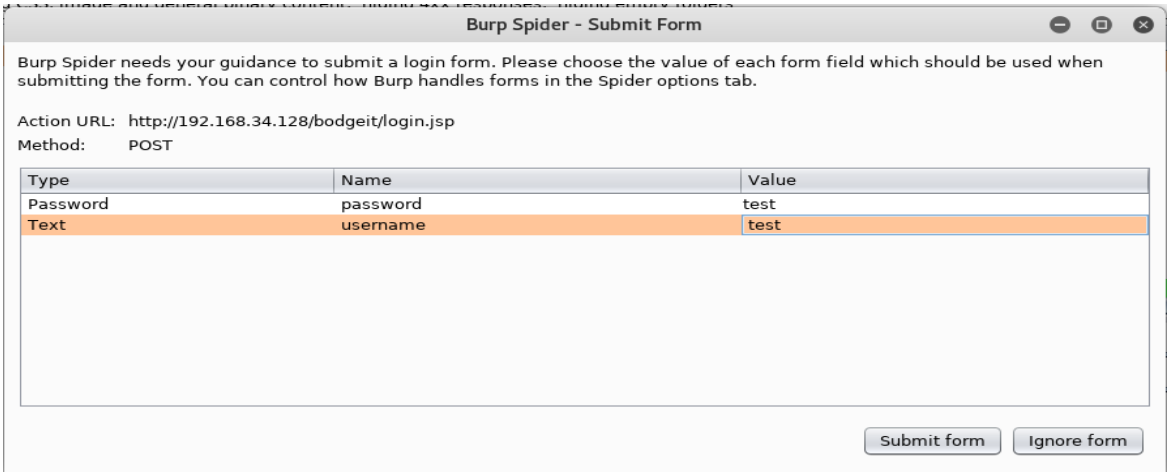
Kartica target daje uvid u informacije o web stranicama koje se pregledavaju i zahtjeve koje generira web preglednik. Idući korak je indeksiranje web stranica koristeći „Spider this branch“ opciju u Burp-u.



Slika 5.2 Indeksiranje web stranica

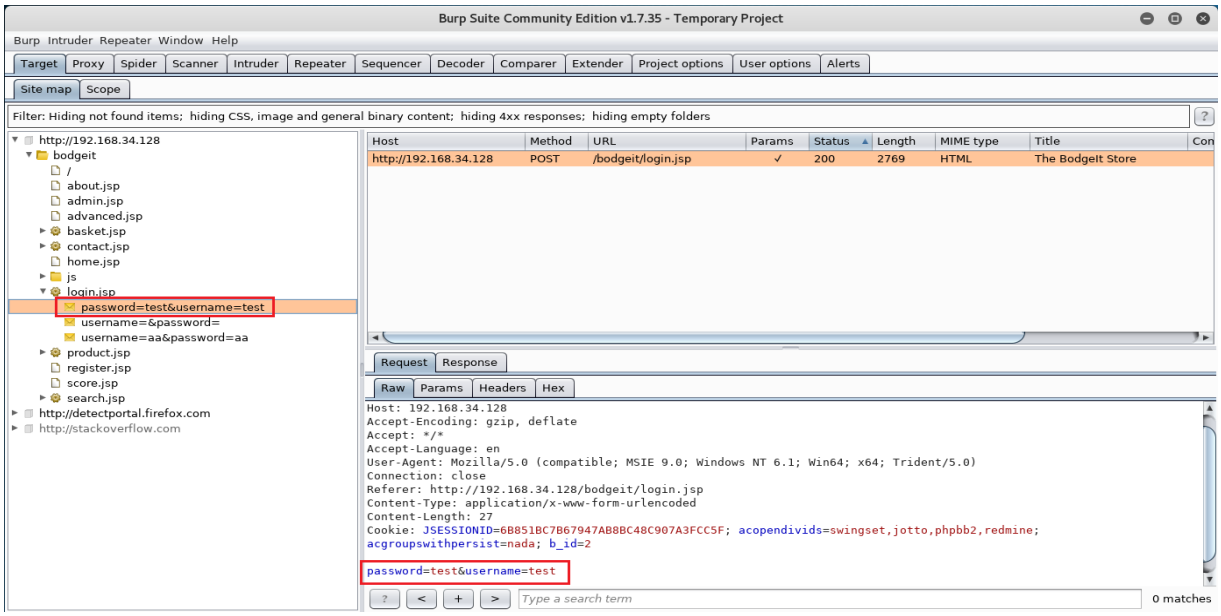
Prije nego li započne s indeksiranjem Burp će korisniku ponuditi opciju za dodavanje web stranice koju želi indeksirati (u ovom slučaju http://192.168.34.128/bodgeit) u zadani opseg. Kako bi indeksirao željenu web stranicu korisnik mora potvrditi ponuđenu opciju. Prema zadanim postavkama, Burp će samo indeksirati one web stranice koje su definirane unutar kartice Scope (opseg).

Nakon toga alat će započeti s radom. Ako tijekom indeksiranja otkrije obrazac za prijavu, zatražit će od korisnika korisničko ime i lozinku za prijavu. Korisnik obrazac može ignorirati i alat će nastaviti s radom ili može podnijeti neke vrijednosti i Burp će ispuniti obrazac s tim vrijednostima. Na slici 5.3 može se vidjeti da je korisnik definirao Burp-u da za korisničko ime i lozinku koristi vrijednost test .



Slika 5.3 Otkriveni obrazac za prijavu

Jednom kada Burb završi sa indeksiranjem rezultati će biti prikazani unutar kartice Site Map (Mapa web stranice). Na slici 5.4 može se vidjeti kako izgleda zahtjev za prijavom poslan preko obrasca za prijavu web servera.

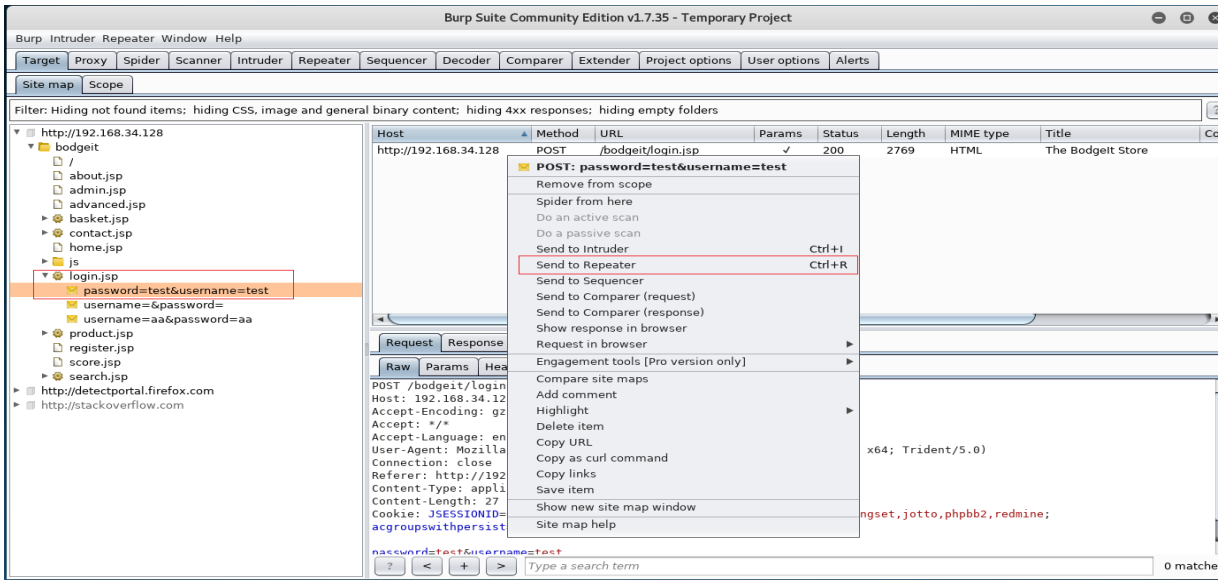


Slika 5.4 Zahtjev za prijavu na web stranicu The Bodgeit Store

5.1. Ponavljanje zahtjeva sa Burp repeater-om

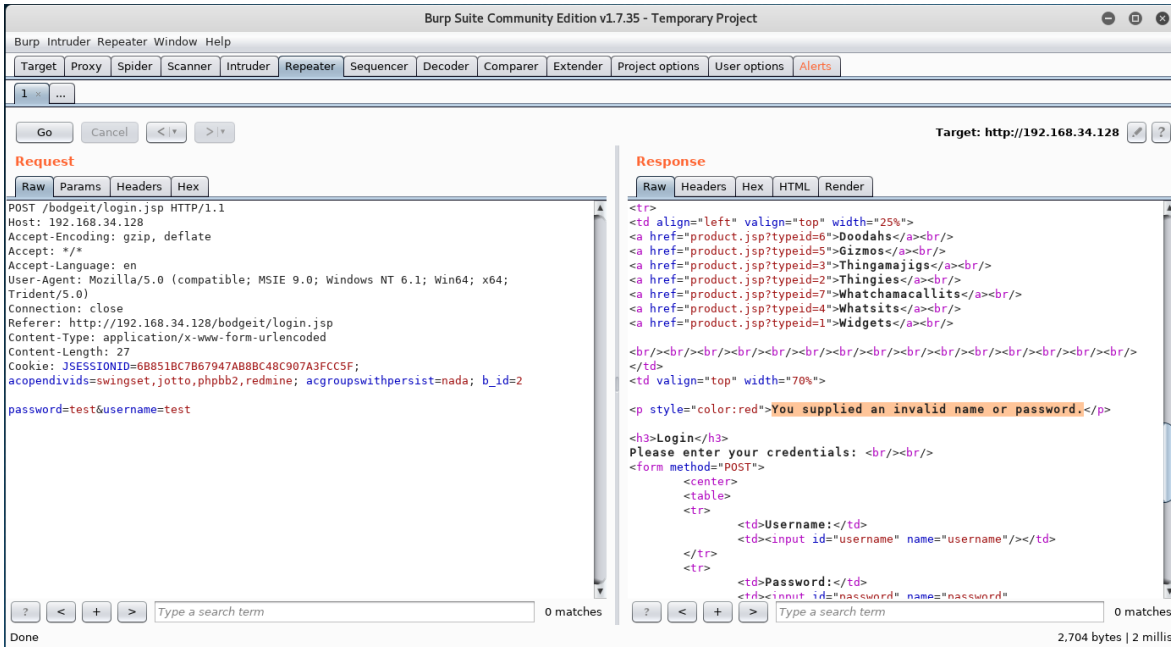
Prilikom analize rezultata Burp Spider-a može biti korisno slati različite verzije istog zahtjeva u kojemu se mijenjaju određeni parametri.

Prvi korak je otići na karticu Target, a zatim na zahtjev koji je Burp-ov Spider napravio za obrazac preko kojeg se vrši prijava na web stranicu(http://192.168.34.128/bodgeit/login.jsp) tj. onaj obrazac u kojemu je u prethodnom koraku za parametre prijave definirana vrijednost test .



Slika 5.5 Slanje zahtjeva na Burp Repeater tj. priprema zahtjeva za ponavljanje

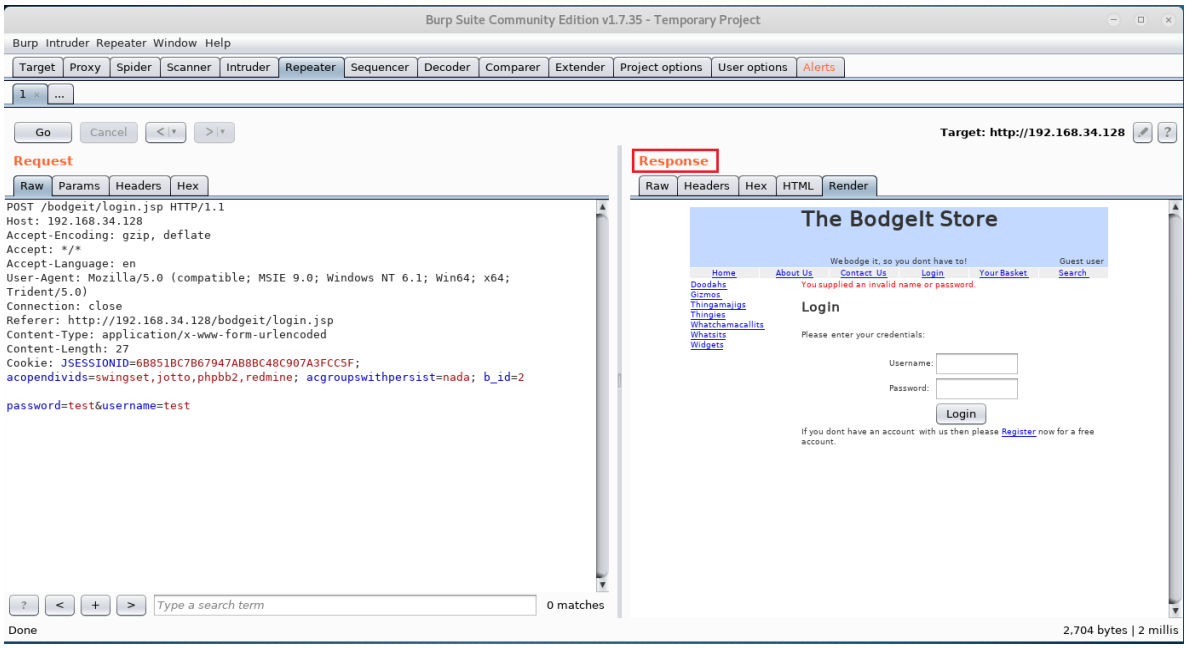
Nakon toga potrebo je otići na karticu repeater i pokrenuti ponavljanje zahtjeva sa klikom na tipku go.



Slika 5.6 Ponavljanje zahtjeva

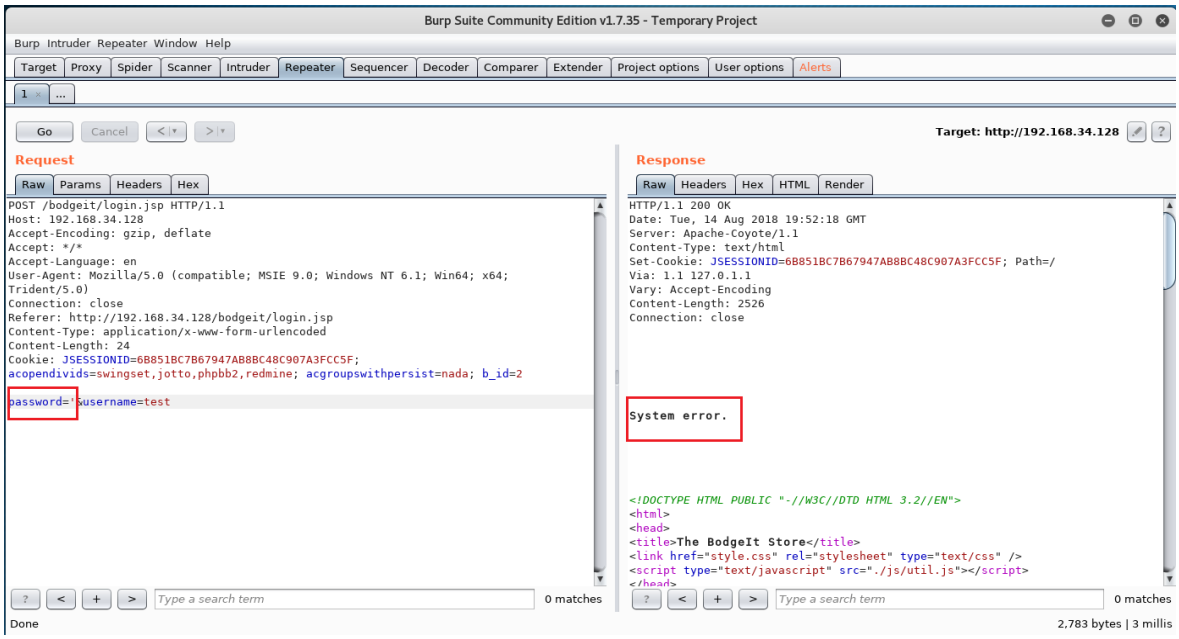
U odjeljku Request (zahtjev) može se vidjeti neobrađeni zahtjev koji će biti poslan prema serveru poslužitelj. U prvom retku prikazani su metoda koja se koristi (POST), traženi URL (/bodgeit/login.jsp) i protokol (HTTP 1.1). Sljedeći redci pa sve do retka Cookie predstavljaju parametre zaglavlja, nakon njih slijedi prekid linije, a zatim POST parametri s vrijednostima koje se šalju obrascu za prijavu.

U odjeljku Response (odgovor) nalaze se kartice: Raw, Header(zaglavlje), Hex, HTML i Render koje daju prikaz istih informacija u različitim formatima.



Slika 5.7 Prikaz web stranice u render modu

U odjeljku Request moguće je izmijeniti vrijednost parametra i poslati novi zahtjev prema serveru. Na primjer ukoliko se izmjeni vrijednost parametra lozinka (prethodno test) sa apostrof-om (') nakon slanja zahtijeva server će generirati sljedeći odgovor (response):



Slika 5.8 Slanje modificiranog zahtjeva i odgovor servera na zahtjev.

Kao što se može vidjeti iz slike 5.8 slanjem modificiranog zahtjeva izazvana je grešku na strani servera što može indicirati da postoji potencijalna ranjivost u web aplikaciji.

5.2. Identificiranje relevantnih datoteka i direktorija iz rezultata indeksiranja

[58]Nakon što Burp završi s radom i složi popis datoteka i direktorija web aplikacije, sljedeći korak je identificirati koje od tih datoteka sadrže relevantne informacije ili povećavaju šanse za pronalazak potencijalnih ranjivosti. Redoslijed filtriranja je sljedeći:

1. Web stranice za prijavu i registraciju: one web stranice preko kojih je moguće postati legitimni korisnik aplikacije ili se lažno predstaviti web aplikaciji pogađanjem (brute force napad) korisničkog imena i lozinke. Neki od primjera su obično web stranice koje u nazivu sadrže: Account, Auth, Login, Logon, Registration, Register, Signup, Signin.
2. Stranice za oporavak lozinke: web stranice koje u nazivu sadrže: change, forgot, lost-password, password, recover, reset.
3. Stranice za administriranje web aplikacije kao što su Admin, Config, Manager, Root... .
4. Sustavi za upravljanje sadržajem (CMS), baze podataka itd. . Neki od primjera Admin console, Adminer, Administrator, Couch,Manager, Mylittleadmin, PhpMyAdmin, SqlWebAdmin, Wp-admin .
5. Aplikacije u fazi testiranja i razvoja koje su obično slabije zaštićene i bolje sklone ranjivostima od konačnih izdanja. Navedene u nazivu često sadrže: Alpha, Beta, Dev, Development, QA, Test .
6. Konfiguracijske datoteke i one koje sadrže Informacije o web poslužitelju kao što su : config.xml, info, phpinfo, server-status, web.config.

6. Identifikacija ranjivosti

Jednom kada faza izviđanja završi tj. kada se prikupe informacije o serveru i okruženju u kojem rade web aplikacije i njihovim potencijalnim slabim točkama moguće je preći na fazu testiranja i otkrivanje ranjivosti web aplikacija.

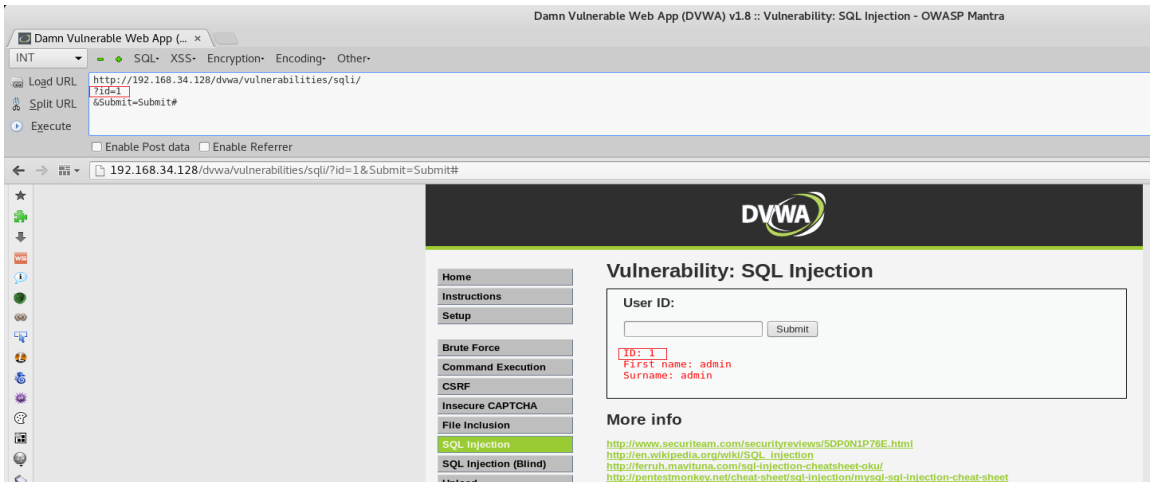
[58] Kali Linux Web Penetration Testing Cookbook, Gilberto Najera-Gutierrez, 2016: 73

6.1. Hackbar

[59]Prilikom testiranja web aplikacije interakcija s adresnom trakom web preglednika je neizbježan zadatak svakog testera a uključuje dodavanje, izmjenu parametara te izmjenu URL-a . Neki serveri će na zahtjev odgovoriti s preusmjeravanjem na drugu web stranicu, ponovnim učitavanjem web stranice ili promjenom parametara. Kada bi ručno isprobavao različite vrijednosti za pojedine parametre testeru bi trebala značajna količina vremena da isproba razne kombinacije . Iz tog razloga razvijeni su razni alati koji ubrzavaju i olakšavaju ovaj postupak. Jedan od takvih alata je Hackbar.

Hackbar je dodatak za Firefox web preglednik koji radi poput adresne trake, međutim na njega ne utječu preusmjeravanja ili druge promjene uzrokovane odgovorom servera, što ga čini idealnim alatom za ovaj tip testiranje web aplikacija.

Za potrebe ovog testiranje koristit će se [60]OWASP DVWA web aplikacija, a kao web preglednik koristit će se [61]OWASP mantra.



Slika 6.1 Hackbar

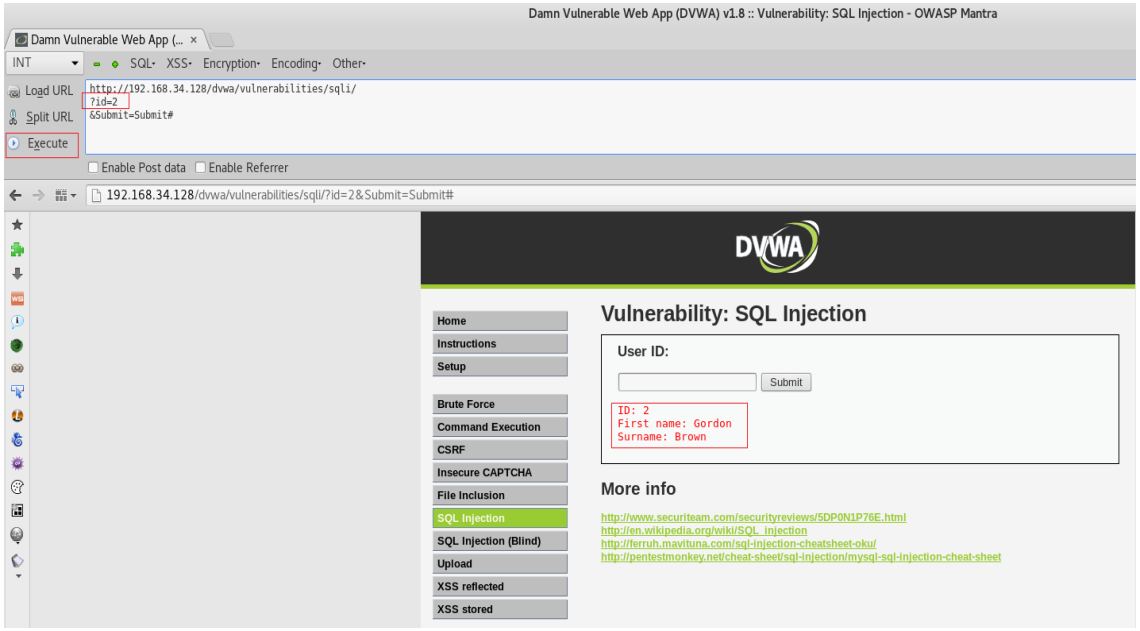
Na primjer ukoliko se u polje User ID: unese određena vrijednost i pošalje preko tipke Submit aplikacija će kao odgovor vratiti ime i prezime korisnika ovisno o njegovom jedinstvenom identifikatoru. Nakon što je zahtjev predan aplikaciji pritiskom na tipku F9 otvara se Hackbar adresna traka.

---

[59] Kali Linux Web Penetration Testing Cookbook, Gilberto Najera-Gutierrez, 2016: 78
[60] http://www.dvwa.co.uk/ , 15.8.2018. 19:16:16
[61] https://www.owasp.org/index.php/OWASP_Mantra_-_Security_Framework, 15.8.2018. 19:36:09

Hackbar kopira URL i njegove parametre. Također je moguće omogućiti opciju za modifikaciju POST zahtjeva (Enable Post data) i Referrer parametra (Enable Referrer) koji pruža serveru informaciju o URL-u preko kojeg je web stranica zatražena.

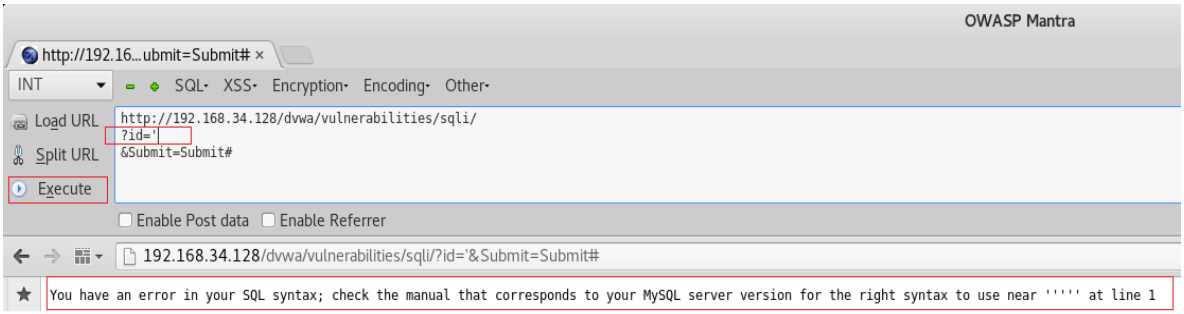
Na primjer ukoliko se preko Hackbar adresne trake izmjeni vrijednost parametra id iz 1 u 2 i klikne na tipku Izvrši (Execute) ili upotrijebi kombinacija tipki Alt + X rezultat će biti sljedeći:



Slika 6.2 Izmjena parametra id preko alata Hackbar

Iz ovoga se može zaključiti da parametar id odgovara vrijednosti koja se unosi preko tekstualnog okvira na stranici , što znači da je koristeći Hackbar moguće isprobati bilo koju vrijednost parametra ID bez potrebe za unošenjem vrijednosti u polje User ID i podnošenje zahtjeva. To može biti posebno korisno kod testiranja obrasca koji imaju puno ulaznih vrijednosti ili koji preusmjeravaju korisnika na druge web stranice ovisno o vrijednostima koje korisnik podnese.

Međutim što će se dogoditi ako se za vrijednost id stavi neispravna vrijednost npr. apostrof (’):



Slika 6.3 Greška u aplikaciji

Ukoliko se za id koristi vrijednost koju aplikacija ne očekuje doći će do pogreške što može upućivati na to da postoji potencijalna ranjivost u web aplikaciji.

6.2. Presretanje i modificiranje zahtjeva sa Tamper Data dodatkom za firefox web preglednik

[62]Ponekad web aplikacije imaju mehanizme koji služe za provjere valjanosti vrijednosti koje dolaze od strane klijenta putem JavaScripta, skrivenih obrasca ili POST parametara koji se ne mogu izravno vidjeti i/ili manipulirati u adresnoj traci. Za testiranje takvih vrijednost potrebo je prvo presresti zahtjeve koje web preglednik šalje i izmijeniti ih prije nego što dođu do servera. Jedan od alata koji može postići navedeno je Tamper Data dodatak za firefox web preglednik .

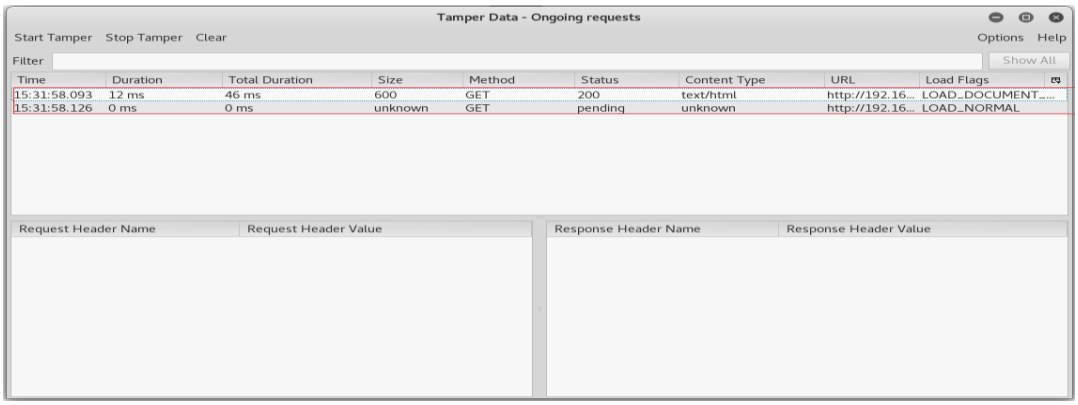
Prvi korak je otvoriti OWASP mantra web preglednik I otići na glavni izbornik te otvoriti karticu Tools→Application Auditing→Temper Data



Slika 6.4 Tamper Data

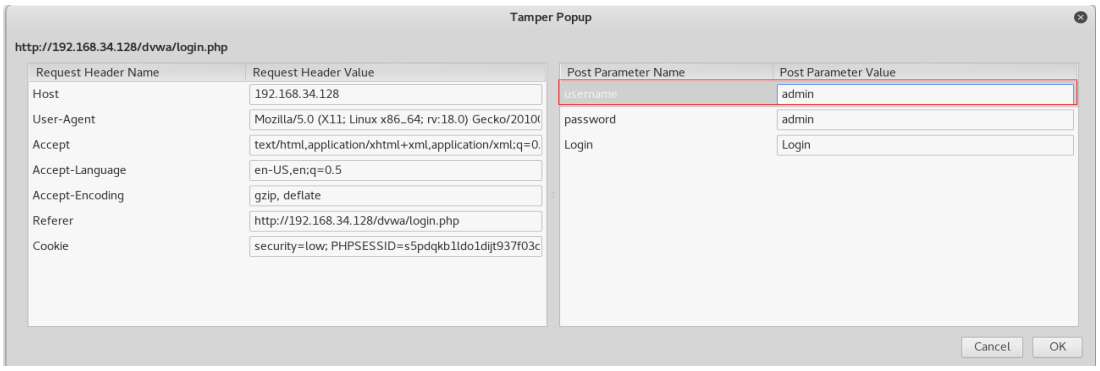
Pojavit će se prozor Tamper data. Zatim je potrebno posjetiti web stranicu http://192.168.34.128/dvwa/login.php. Nakon toga moguće je dobiti uvid u trenutno aktivne zahtjeve :

[62] Kali Linux Web Penetration Testing Cookbook, Gilberto Najera-Gutierrez, 2016: 80



Slika 6.5 Aktivni zahtjevi tj. zahtjevi u tijeku

Presretanje zahtjeva započinje klikom na opciju Start Tamper nakon čega je moguće izmijeniti njegove vrijednosti. Na primjer ako revizor prvi put unese neispravne podatke alat će mu ponuditi opciju da nastavi s modifikacijom zahtjeva te ukoliko potvrdno odgovori otvorit će mu se novi Tamper Popup prozor preko kojega je moguće izmijeniti podatke(uključujući zaglavlje zahtjeva i POST parametre) koji se šalju serveru.



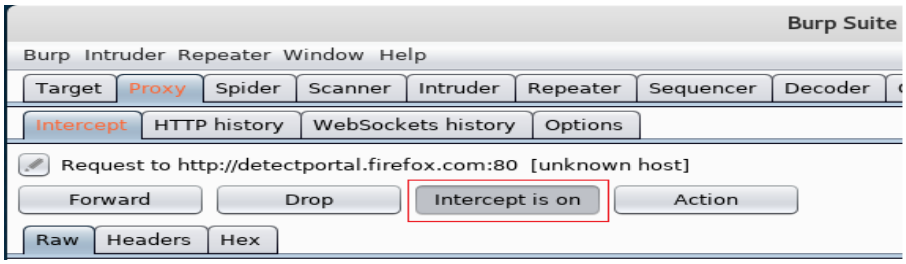
Slika 6.6 Presretanje i modificiranje zahtjeva

Na taj način izmijenjene su vrijednosti u obrascu neposredno prije nego se pošalju pregledniku.

6.3. Presretanje i modifikacija zahtjeva sa Burp Suite-om

Ranije je bilo moguće vidjeti kako Burp radi indeksiranje web stranice i ponavljanje zahtjeva preko repeater opcije. U ovom primjeru koristiti će se Burp proxy za presretanje i modifikaciju zahtjeva te zaobilaženje mehanizma za validaciju unosa od strane klijenta. Za potrebe testiranja koristit će se Multidae II web aplikacija.

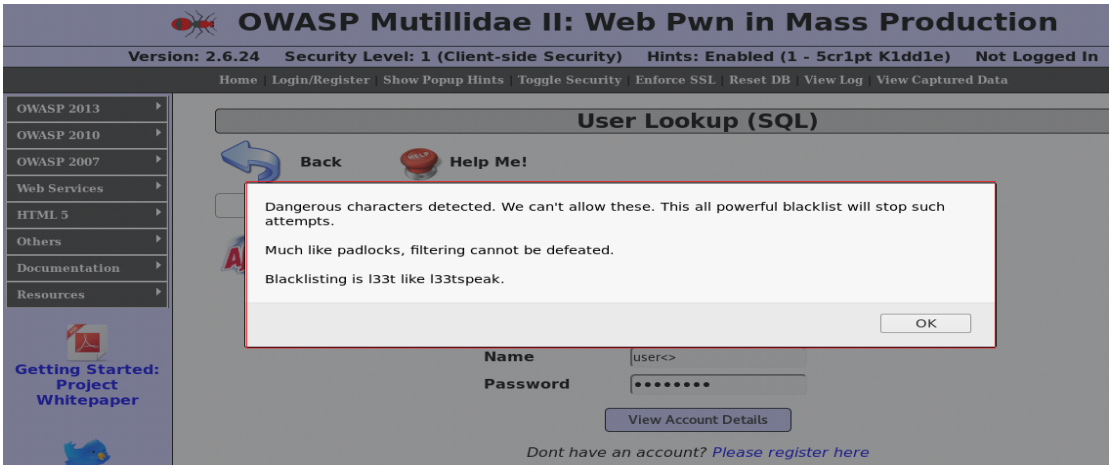
Po zadanim postavkama Burp-ov Proxy će automatski presretat zahtjeve te ga je potrebno ugasiti kako ne bi bilo potrebno svaki puta kad se u web pregledniku učita nova web stranica pritiskati tipku Forward (proslijedi zahtjev).



Slika 6.7 Burp Proxy

Prvo je potrebno posjetiti početnu stranicu Multidae II web aplikacije te iz izbornika otići na OWASP Top 10 2013 → A1 - SQL injection→ SQLi - Extract Data →User info (SQL).

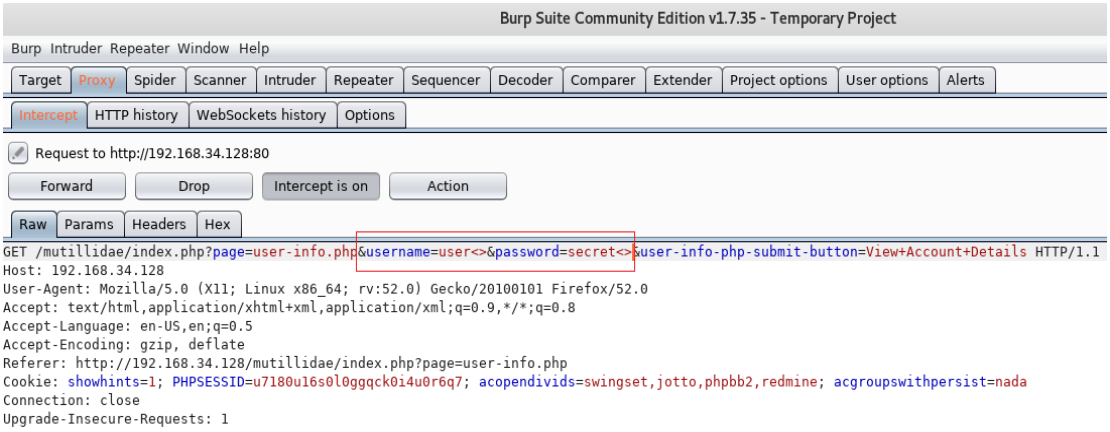
Otvorit će se stranica za pretraživanje korisnika na temelju njegovog korisničkog imena i lozinke. Na primjer ako se u polje Name upiše vrijednost user<> (uključujući simbole) i secret<> u polje Password te nakon toga potvrdi unos klikom na View Account Details (Prikaz podataka o računu) web aplikacija će vratiti poruku upozorenja da su korišteni nedozvoljeni tj. potencijalno opasni znakovi.



Slika 6.8 Pogreška o nedopuštenim znakovima u obrascu za unos korisničkog imena i lozinke

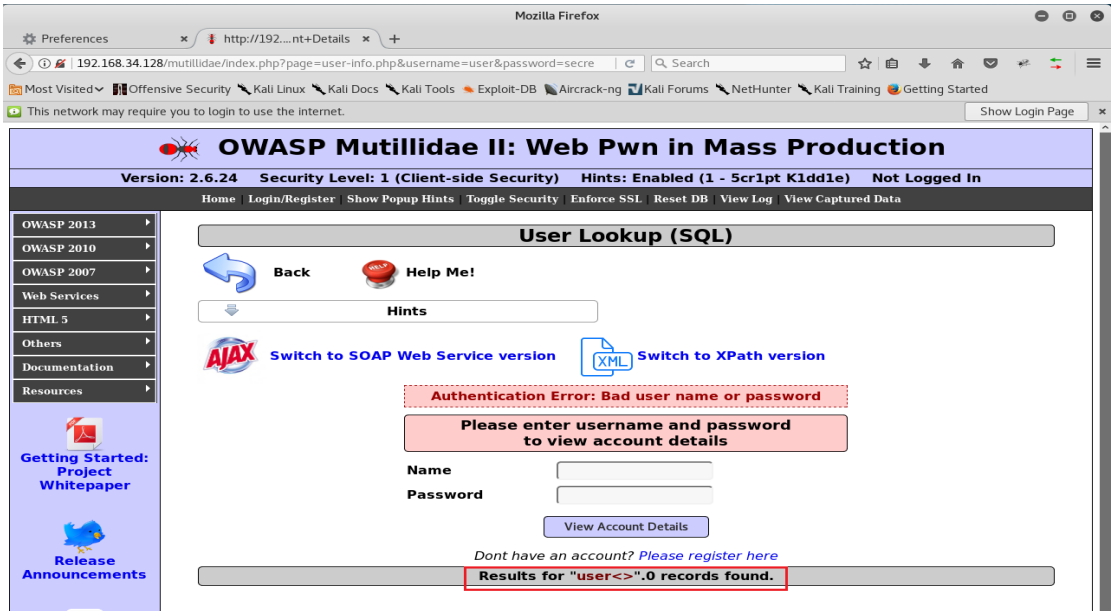
Iz ovoga se može zaključiti da web aplikacija ne dopušta unos specijalnih znakova u polja za korisničko ime i lozinku što upućuje na to da postoji mehanizam za provjera valjanosti unosa vrijednosti koje klijent šalje. Dodatna potvrda postojanja mehanizma za provjeru valjanosti unosa je da zahtjev nije zabilježen Burp Proxy HTTP history kartici . Međutim to ne znači da takav mehanizam nije moguće zaobići. Prvi korak je ponovno uključiti presretanje zahtjeva u kartici Burp Proxy.

Sljedeći korak je popunjavanje obrasca za pretraživanje korisnika sa ispravnim podatcima, kao što su npr. user i secret. Burp-ov proxy će presresti taj zahtjev i omogućiti izmjenu parametara korisničko ime i lozinka dodavanjem <> zabranjenih znakova.



Slika 6.9 Presretanje i modificiranje zahtjeva

Ukoliko se modificirani zahtjev proslijedi serveru rezultat je sljedeći:



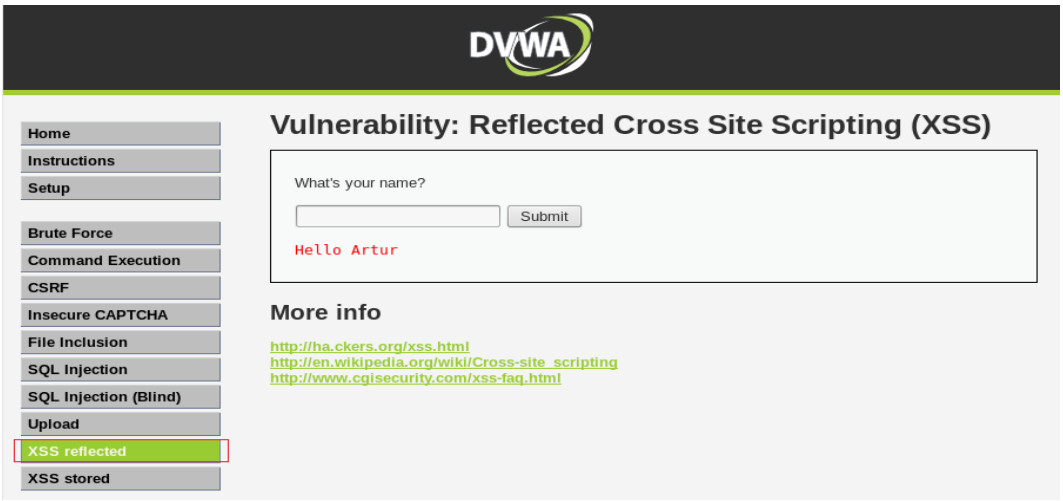
Slika 6.10 Odgovor web aplikacije na modificirani zahtjev

Iz slike 6.10 može se vidjeti uspješno zaobilaženje mehanizma za validaciju unosa jer proxy omogućava modifikaciju zahtjeva nakon što je prošao mehanizam za validaciju unosa.

6.4. Identificiranje cross-site scripting (XSS) ranjivosti

Cross-site scripting (XSS) je jedna od najčešćih ranjivosti u web aplikacijama. [63]OWASP Top 10 2017. stavlja ju na 7. mjesto najkritičnijih ranjivosti web aplikacija.

Za potrebe testiranja koristit će se OWASP DVWA aplikacija. Prvi korak u testiranju XSS ranjivosti je promatranje normalnog odgovora aplikacije.



Slika 6.11 Normalni odgovor web aplikacije

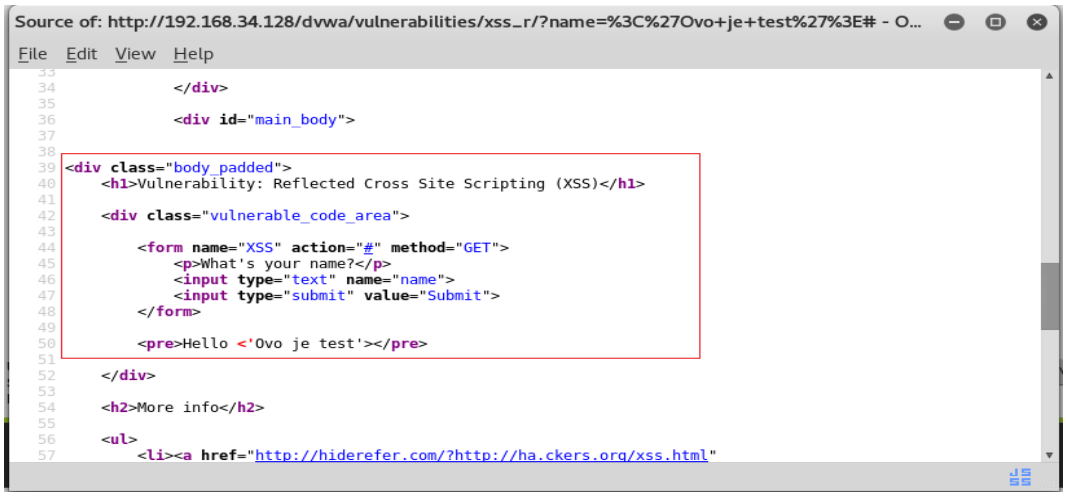
Iz slike 6.11 može se vidjeti da aplikacija koristi korisnikov unos za formiranje fraze. Međutim što ako se umjesto ispravnog unosa koriste specijalni znakovi npr. <'Ovo je test'>?



Slika 6.12 Odgovor web aplikacije na unos specijalnih znakova

Iz slike 6.12 moguće je zaključiti da će se sve što korisnik unese u tekstualno polje odraziti u odgovoru web aplikacije tj. postati će dio HTML koda web stranice. Navedeno je moguće potvrditi analizom izvornog koda web stranice (desni klik na web stranicu, pa View page Source).

[63] https://www.owasp.org/index.php/Top_10-2017_Top_10 , 18.8.2018. 17:48:55



Slika 6.13 HTML kod web stranice

Analizom Izvornog koda može se utvrditi da ne postoji kodiranje tj. filtriranje za specijalne znakove prilikom generiranja odgovora na korisnikov upit ,a specijalni znakovi koje klijent šalje projiciraju se na web stranici bez ikakve prethodne obrade. Kako se kod html-a znakovi <> koriste za definiranje HTML oznaka postoji mogućnost za ubacivanje koda u web stranicu.

Na primjer ako korisnik u polje za unos unese normalnu vrijednost (u primjeru „Artur“) popraćenu sa jednostavnim html kodom <script> alert ('XSS_test') </ script> i klikne na tipku Submit(podnesi):



Slika 6.14 Identificirana XSS ranjivost

Web stranica će uspješno izvršiti skriptu što dokazuje da je ova web stranica uistinu ranjiva na XSS tj. Cross Site Scripting. Što se točno dogodilo moguće je utvrditi daljnjom analizom izvornog koda web stranice.



Slika 6.15 Analiza nakon testiranja XSS ranjivosti

Iz slike 6.15 može se vidjeti da je korisnikov unos obrađen kao da je dio HTML koda. Web preglednik je interpretirao <script> oznaku i izvršio kod unutar nje.

6.5. Identifikacija SQL injection ranjivosti na temelju poruka o greškama

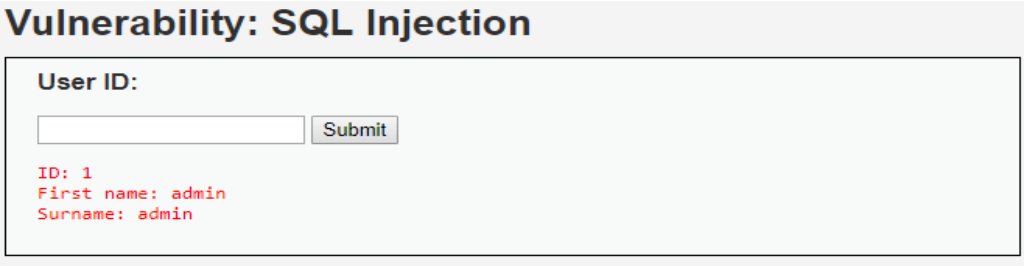
Injection ranjvosti prema OWASP Top 10 2017. zauzimaju prvo mjesto na ljestvici sigurnosnih ranjivosti web aplikacija. Jedan od najjednostavnijih primjera injection ranjivosti je SQL injection.

[64]U većini modernih web aplikacija implementirana je neka vrsta baze podataka, bilo lokalno ili udaljeno. Najčešće se koriste SQL baze podataka. U SQLi napadu, maliciozni korisnik pokušava zloupotrijebiti komunikaciju između aplikacije i baze podataka tako što navodi web aplikaciju na slanje izmijenjenih upita bazi podataka, a to postiže ubrizgavanjem SQL naredbe u obrasce za unos podataka ili bilo koji drugi parametar u zahtjevu koji se koristi za konstruiranje SQL upita na serveru.

Za potrebe testiranja SQLi ranjivosti koristit će se OWASP DVWA web aplikacija.

Prvi korak je identičan kao i kod testiranja XSS ranjivosti tj. prvo je potrebno promatrati odgovor aplikacije na normalni/regularni upit.

[64] Kali Linux Web Penetration Testing Cookbook, Gilberto Najera-Gutierrez, 2016: 93

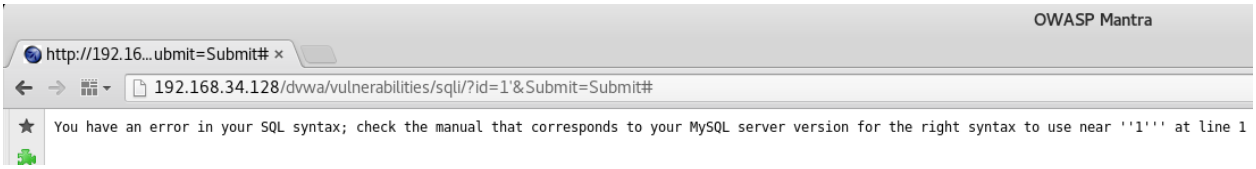


Slika 6.16 Odgovor aplikacije na normalan upit

Iz slike 6.16 može se vidjeti da je web aplikacije najprije poslala upit bazi podataka o tome postoji li korisnik sa ID-om 1 te je zatim vratila odgovor na upit korisnika.

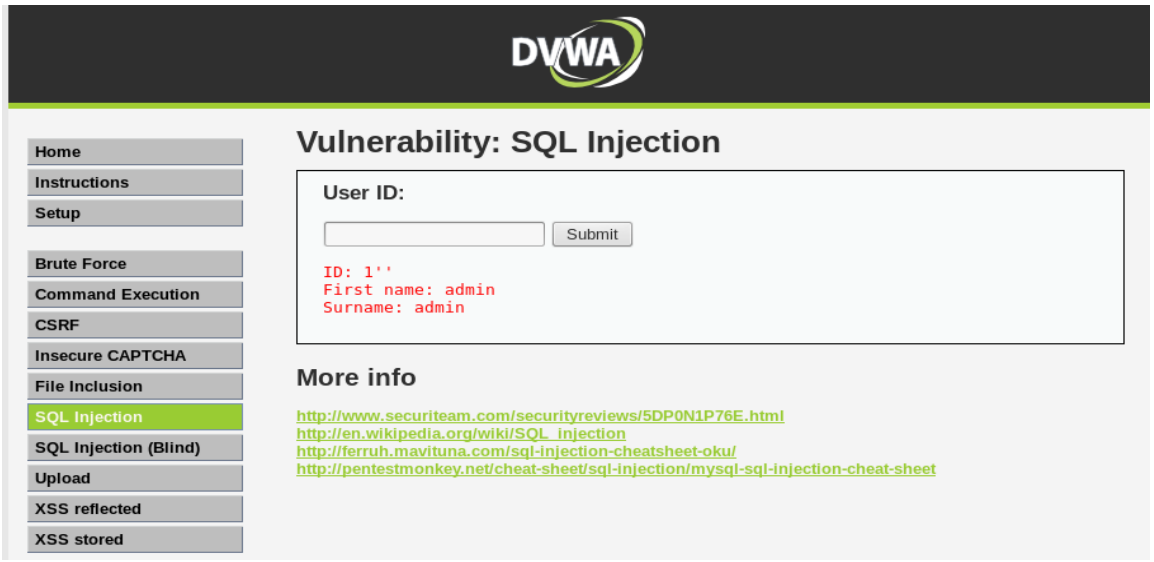
Međutim što ako korisnik pošalje neispravan upit, npr. U polje User ID: unese vrijednost 1'

.



Slika 6.17 Odgovor aplikacije na neispravan upit

Ova poruka o pogrešci govori korisniku da je izmijenio dobro oblikovani upit. Međutim to još uvijek ne znači da u web aplikaciji postoji SQLi. Kako bi se potvrdilo postojanje SQLi ranjivosti potrebno je web aplikaciji poslati novi upit , ali ovog puta će se koristiti dva apostrofa tj. upit će glasit 1" .



Slika 6.18 Identificirana SQLi ranjivost

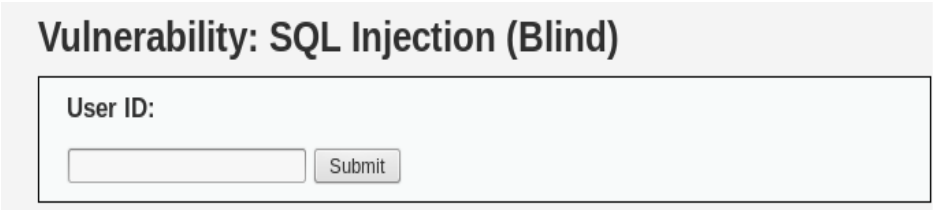
Web aplikacija je procesirala korisnikov upit što znači da u toj web aplikaciji postoji SQL Injection, a to se može potvrditi unosom ' or 1'='1 u polje User ID, rezultat je sljedeći:



Slika 6.19 Popis korisnika koji postoje u bazi podataka

6.6. Identificiranje blind SQL injection ranjivosti

Blind SQLi je gotovo identična ranjivost kao i SQLi. Razlika između ove dvije ranjivosti je u tome što kod blind SQLi-a web aplikacije neće prikazivati poruke o pogreškama. Na primjer ukoliko korisnik u polje User ID unese vrijednost 1' kao u prethodnom primjeru kod identifikacije SLQi ranjivosti web aplikacije mu neće vratiti poruku o neispravnom unosu.



Slika 6.20 Odgovor aplikacije na neispravan upit.

Iz slike 6.20 može se vidjeti da nema poruke o pogrešci , ali ni odgovora na korisnikov upit što može značiti da se nešto zanimljivo odvija u pozadini web aplikacije. Slijedi drugi test tj. slanje upita koji sadrži dva apostrofa (1") .

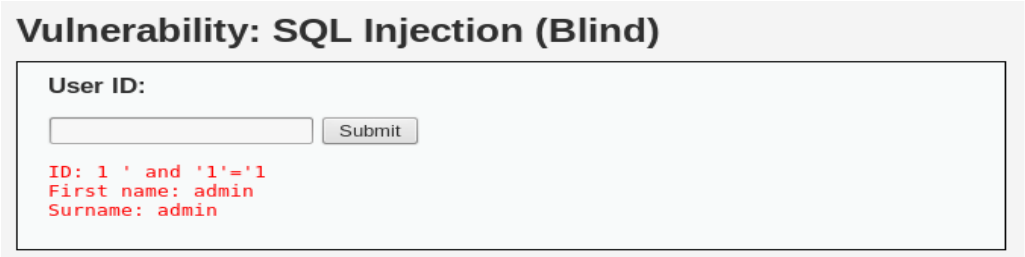


Slika 6.21 Odgovor aplikacije na „ispravan“ upit.

Prikazan je rezultat za ID = 1, što znači da je prethodni test (1 ') rezultirao pogreškom koju je web aplikacija procesirala. To znači da u ovoj web aplikaciji vrlo vjerojatno postoji blind SQLi ranjivost, međutim nema informacija o bazi podataka, pa se identificiranje ove ranjivosti temelji na pogađanju.

[65]Idući korak je pokušati utvrditi što se događa kada korisnik pošalje upit koji je uvijek netočan npr. 1 'and' 1 '=' 2 u polje User ID. '1' nikad nije jednako '2', što znači da niti jedan zapis u bazi podataka neće zadovoljiti kriterije upita i web aplikacija će vratiti prazan odgovor.

Međutim ako korisnik pošalje upit koji će uvijek biti istinit npr. 1 'and' 1 '=' 1, ukoliko u bazi postoji zapis koji ima ID=1 web aplikacija će vratiti odgovor.



Slika 6.22 Identificirana blind SQLi ranjivost

Ovime je potvrđeno da u web aplikaciji postoji blind SQLi ranjivost. Ako web aplikacija daje različite odgovore na SQL upite koji su uvijek neistiniti naspram upita koji su uvijek

[65] Kali Linux Web Penetration Testing Cookbook, Gilberto Najera-Gutierrez, 2016: 96

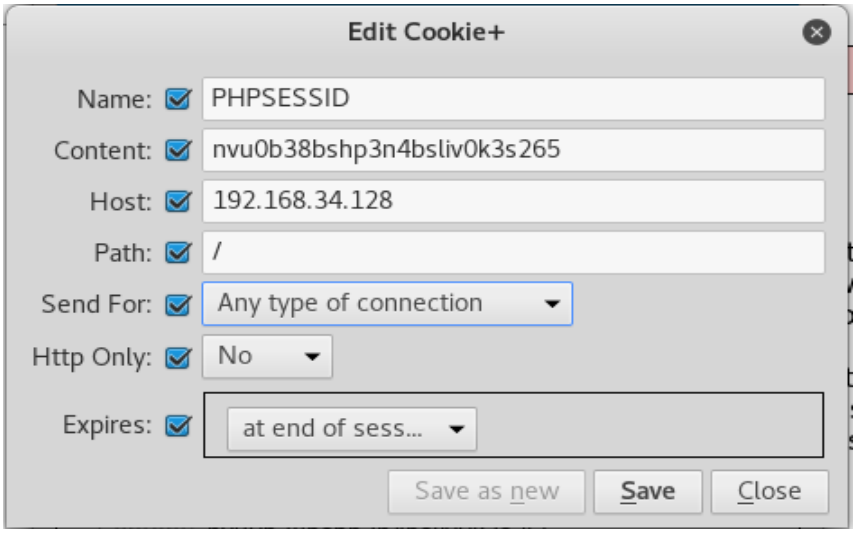
istiniti, postoji ranjivost, jer server izvršava SQL kod čak i ako to ne pokazuje eksplicitno u odgovoru.

6.7. Identificiranje ranjivosti u web kolačićima(Cookies)

[66]Kolačići su mali dijelovi podataka koje web stranica šalje ,a pohranjuju se u korisničkom web pregledniku. U modernim web aplikacijama kolačići se koriste za praćenje sesije korisnika. Spremanjem identifikatora sesije na serveru i na korisničkom računalu, server je u mogućnosti razlikovati zahtjeva od različitih klijenata istovremeno. Kada se zahtjev šalje serveru web preglednik dodaje kolačić, a zatim šalje zahtjev serveru. Na taj način server može utvrditi o sesiju pojedinog korisnika na temelju tog kolačića.

Za potrebe testiranja koristit će se OWASP Mutillidae II web aplikacija i Cookie Manager+ alat za analizu kolačića. Prvi korak je posjetiti početnu stranicu web aplikacije (http://192.168.34.128/ mutillidae) i u glavnom izborniku otići na OWASP Top 10→A3 Broken Authentication and Session managment→ Cookies.

U Cookie Manager+ pojavit će se novi kolačić s nazivom PHPSESSID . Nakon toga je potrebno označiti navedeni kolačić te sa desnim klikom otvoriti meni sa opcijama i odabrati opciju Edit Cookie (uređivanje kolačića) kako bi se dobio uvid u parametre kolačića.



Slika 6.23 Parametri kolačića PHPSESSID

[66] Kali Linux Web Penetration Testing Cookbook, Gilberto Najera-Gutierrez, 2016: 98

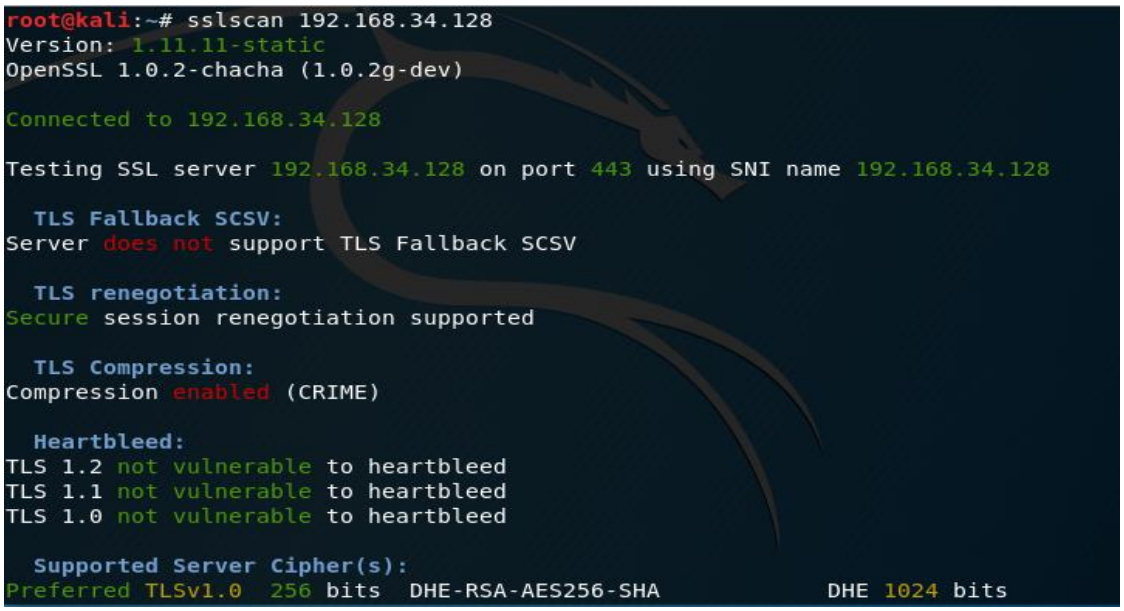
PHPSESSID je zadani naziv kolačića koji se koristi za sesije u web aplikacijama baziranim na PHP-u. Ako se promotre vrijednosti parametra u ovom kolačiću moguće je vidjeti da se ovaj kolačić može slati nesigurnim i sigurnim kanalima (HTTP i HTTPS). Uz to može se pročitati od strane servera i klijenta putem skriptnog koda jer su Secure i HTTPOnly opcije onemogućene. To znači da je ovoj web aplikaciji moguće preuzeti sesije legitimnih korisnika.

6.8. SSLScan alat za dohvaćanje informacija o SSL i TLS protokolu

Revizor mora pretpostaviti da će se tokom penetracijskog testiranja web aplikacija prije ili kasnije susresti s zaštićenom konekcijom tj. HTTPS sa SSL ili TLS enkripcijom što znači ako i presretne pakete koji putuju kroz takav transportni kanal oni će samo sadržavati niz beznačajnih brojeva. Međutim to nije u potpunosti točno tj. HTTPS server mora biti pravilno konfiguriran kako bi pružio dovoljno snažnu razinu enkripcije i zaštitio korisnike od MiTM napada. S obzirom na to da su u implementaciji i dizajnu SSL protokola tijekom proteklih godina otkrivene razne ranjivosti, provedba temeljitog testiranje „sigurnih“ konekcija trebala bi biti dio svakog penetracijskog testiranja web aplikacija.

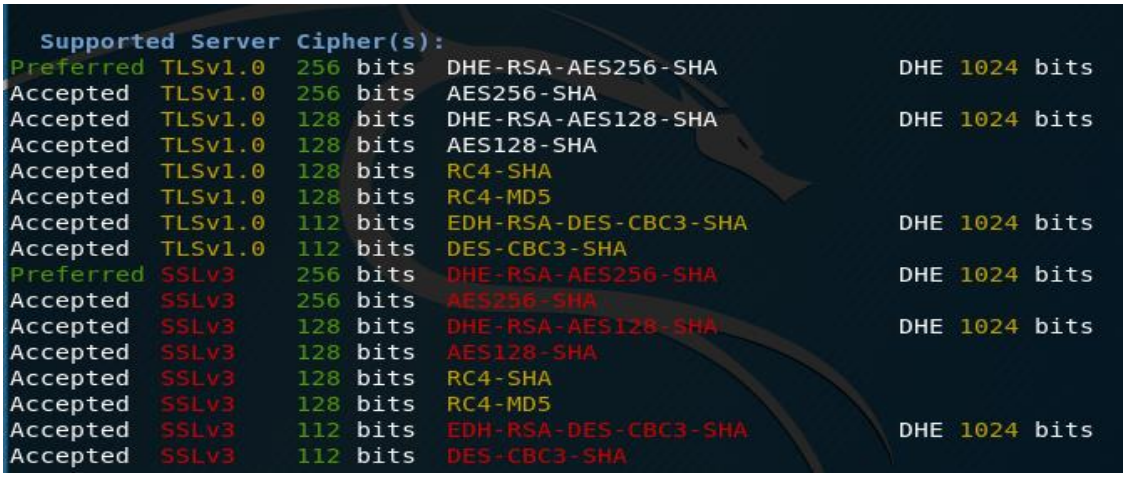
Za potrebe ovog testiranja koristit će se SSLScan alat za analizu SSL i TLS konfiguracije (iz perspektive klijenta) servera.

Osnovna naredba sslscan <ip adresa/ime servera> daje dovoljno informacija o serveru. U ovom slučaju ip adresa servera je 192.168.34.128 .



Slika 6.24 SSL i TLS informacije o serveru

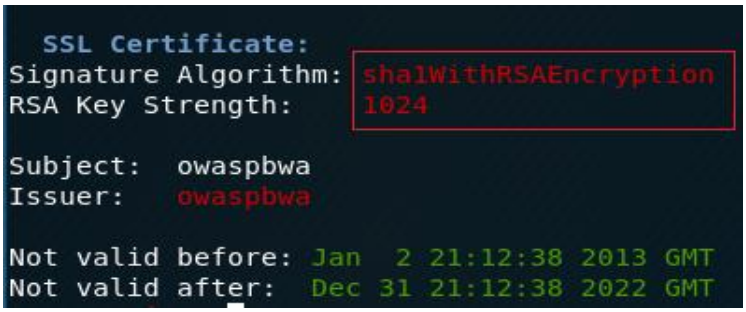
Na slici 6.24 moguće je vidjeti konfiguraciju servera u smislu uobičajenih pogrešaka koje mogu ugroziti sigurnost kao što su : [67]TLS renegotiation, [68]kompresija i [69]Heartbleed.



Slika 6.25 Informacije o podržanim algoritmima za enkripciju komunikacije

Osim TLS/SSL konfiguracije servera SSLScan alat također daje popis algoritama za enkripciju komunikacije koje server prihvaća. Iz slike 6.25 može se vidjeti da server podržava SSLv3 i algoritme kao što je DES, koji se u današnje vrijeme smatraju nesigurnim (nesigurni algoritmi prikazani su u crvenoj boji).

[67] https://www.digicert.com/news/2011-06-03-ssl-renego/ , 21.8.2018. 14:39:24
[68] https://www.acunetix.com/vulnerabilities/web/crime-ssl-tls-attack, 21.8.2018. 14:45:23
[69] https://www.us-cert.gov/ncas/alerts/TA14-098A , 21.8.2018. 15:15:17



Slika 6.26 Informacije o SSL certifikatu

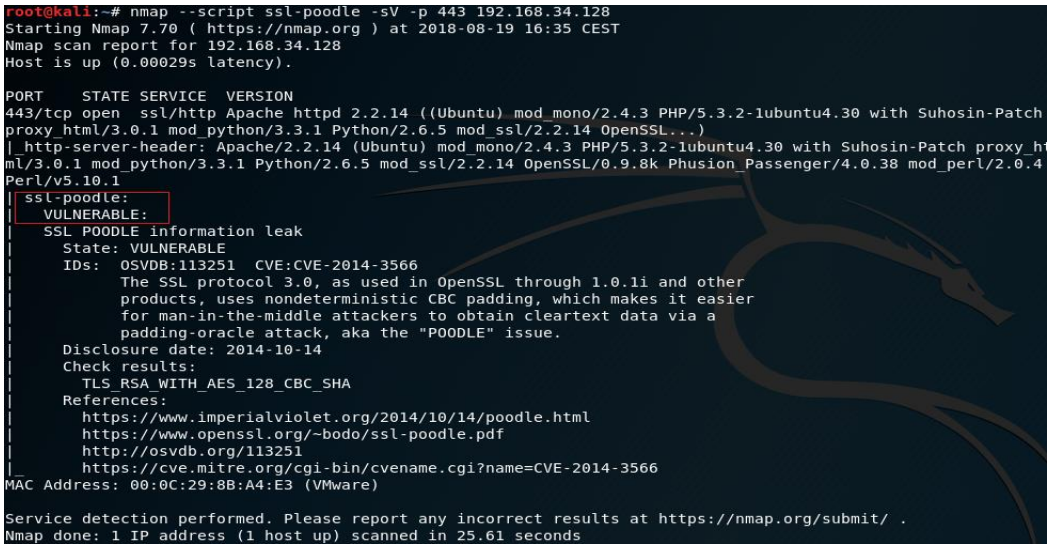
I na kraju moguće je vidjeti informacije o certifikatu kojeg server koristi. Iz slike 6.26 može se vidjeti da server koristi sha1 algoritam za potpis i 1024-bitni RSA ključ za enkripciju koji se u današnje vrijeme smatra „nedovoljno“ sigurnim tj. sigurnosni standardi preporučuju RSA ključ minimalne duljine 2048 bita.

6.9. Identificiranje POODLE ranjivosti

U prethodnom primjeru prilikom analize informacija o SSL i TLS protokolu moguće je bilo vidjeti da server podržava više od jednog algoritma za enkripciju komunikacije. [70]Padding Oracle On Downgraded Legacy Encryption (POODLE) ranjivost to iskorištava na način da navodi server na komunikaciju putem SSLv3 protokola u kombinaciji s Cipher Block Chaining (CBC) algoritmom za enkripciju komunikacije kojega je moguće zaobići tj. maliciozni korisnik može dekriptirati komunikaciju što mu otvara vrata za MiTM napad.

Za testiranje ove ranjivosti koristit će se nmap skripta ssl-poodle na web serveru (192.168.34.128). Prvi korak je preuzeti skriptu ssl-poodle.nse iz [71]nmap repozitorija, a zatim je istu potrebno izvršiti naredbom nmap --script ssl-poodle -sV -p 443 192.168.34.128.

[70]https://blog.cloudflare.com/padding-oracles-and-the-decline-of-cbc-mode-ciphersuites/,22.8.2018. 18:28:16
[71] http://nmap.org/nsedoc/scripts/ssl-poodle.html , 22.8.2018. 18:33:58



Slika 6.27 Identificirana POODLE ranjivost

Nmapa će skenirati port 443(HTTPS) na ip adresi 192.168.34.128 (server) te će izvršiti skriptu ssl-poodle. Iz rezultata skeniranje može se zaključiti da je server ranjiv jer omogućuje komunikaciju putem SSLv3 u kombinaciji s TLS_RSA_WITH_AES_128_CBC_SHA algoritmima .

7. Alati za automatizirano skeniranje ranjivosti

Gotovo svaki projekt penetracijskog testiranja ima strogo definiran raspored, uglavnom prema zahtjevima klijenata. Stoga je za penetracijskog testera korisno da ima na raspolaganju alat koji u kratkom vremenskom može provesti razne testova nad web aplikacijama kako bi se identificiralo što je više ranjivosti moguće. Alati za automatizirano skeniranje ranjivosti su idealni za taj zadatak. Penetracijski testeri ih također mogu koristit kako bi bili sigurni da im nije promaklo nešto očito.

7.1. Nessus

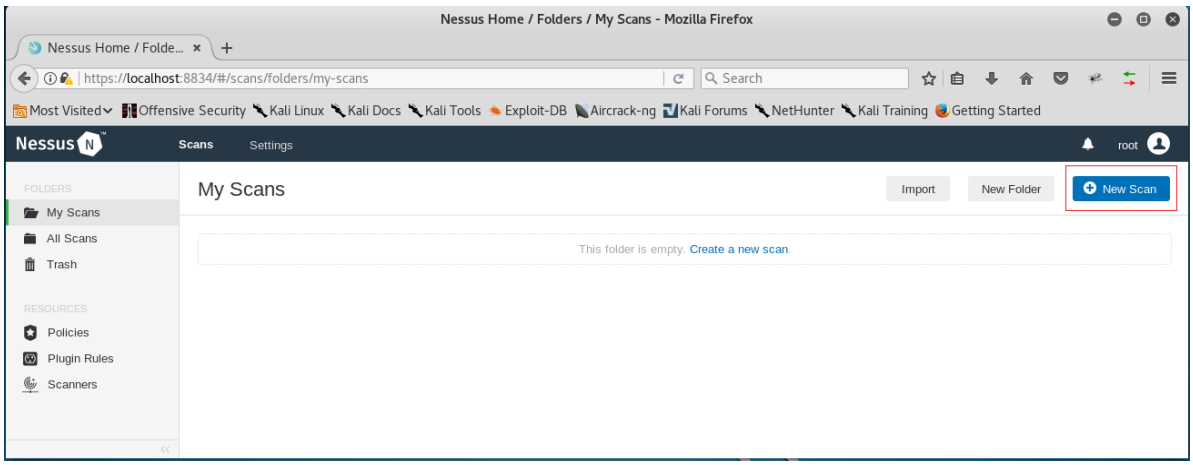
[72]Nessus je jedan od najpopularnijih i najpouzdanijih alata za automatizirano skeniranje ranjivosti. Omogućuje korisniku pronalaženje ranjivosti koje mogu omogućiti malicioznom korisniku udaljeni pristup sustavu i osjetljivim podacima, testiranje slabih i nesigurnih

[72] Kali Linux 2: Assuring Security by Penetration Testing Third Edition, Lee Allen, 2016: 213

lozinki koje se lako mogu pogoditi, greške u konfiguraciji web aplikacija, DDoS ranjivosti...

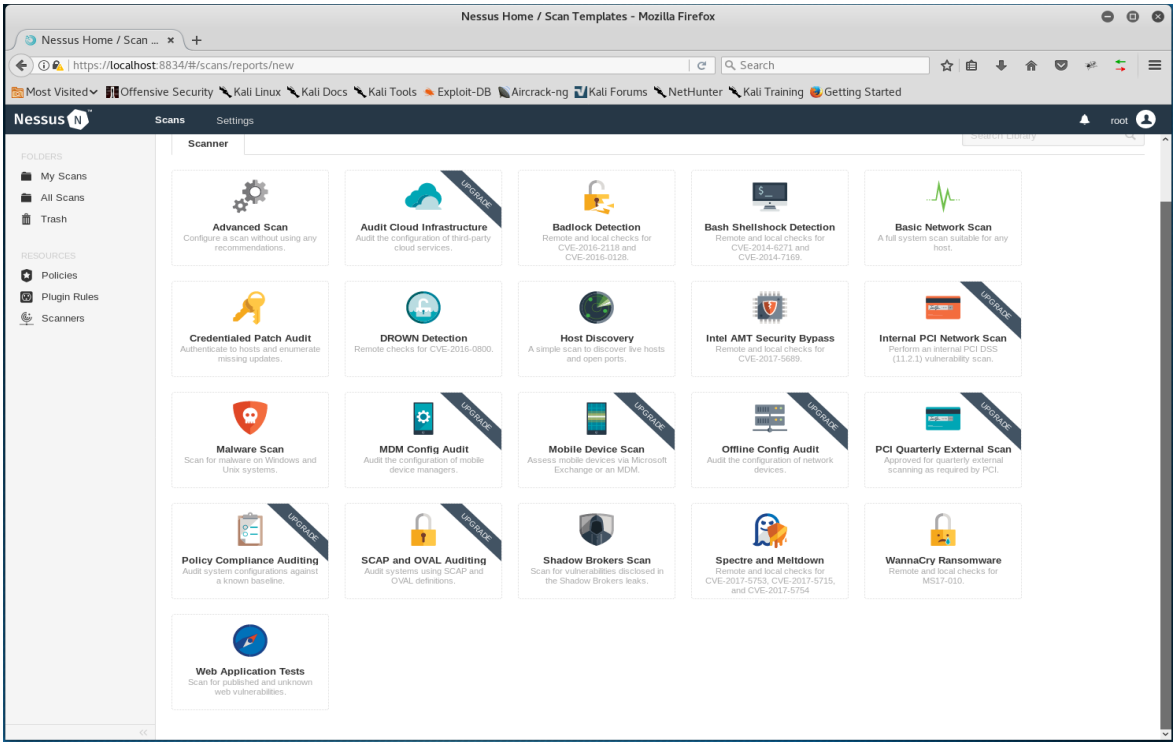
.

Za pristupanje grafičkom sučelju alata potrebo je u web pregledniku otvoriti poveznicu https:// localhost:8834 i prijaviti se sa svojim Nessus korisničkim računom.



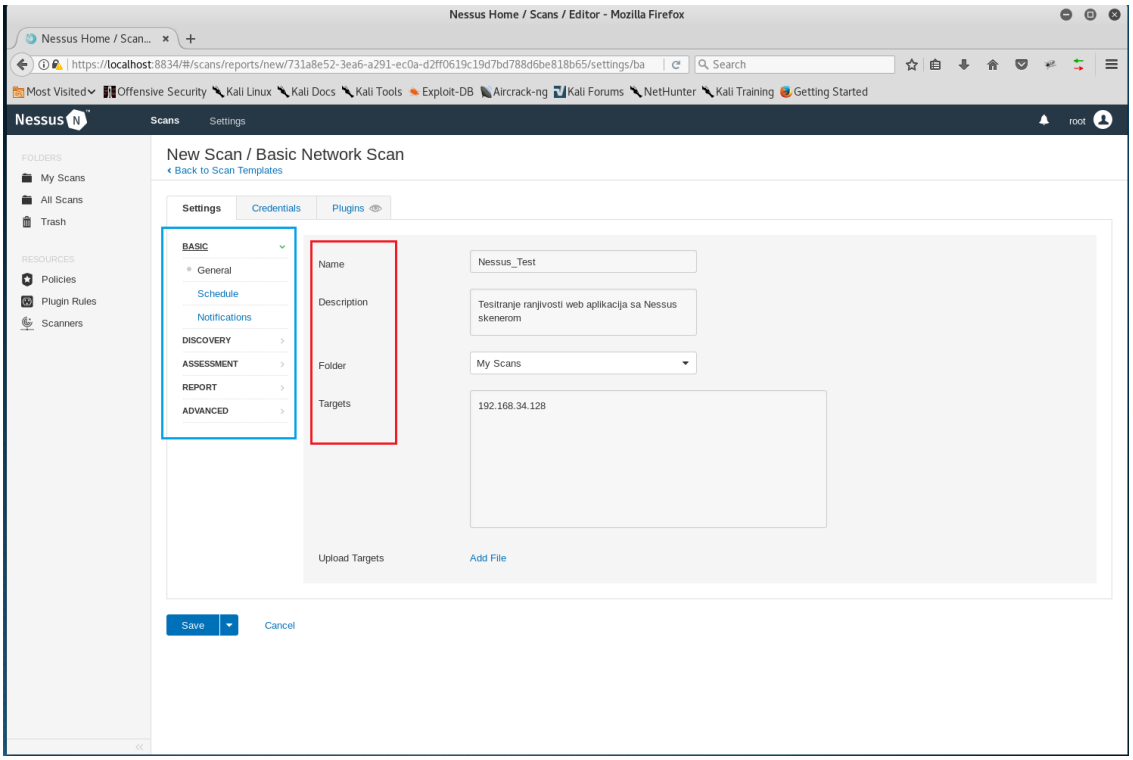
Slika 7.1 Nessus početni zaslon

Prvi korak je odabrati opciju New Scan. Nakon toga alat će ponuditi niz gotovih predložaka za skeniranje ranjivosti, a korisnik ukoliko mu gotovi predlošci ne odgovaraju može definirati vlastiti predložak.



Slika 7.2 Mogućnosti skeniranja ranjivosti

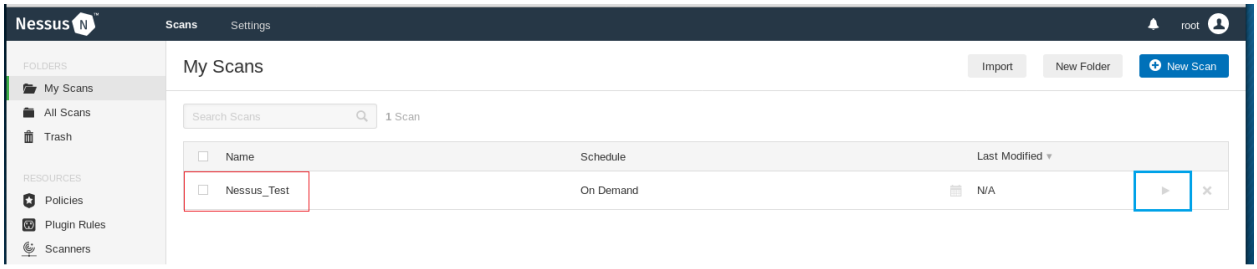
Kako što je moguće vidjeti postoji niz gotovih predložaka za skeniranje sustava. Za potrebe demonstracije alata koristit će se Basic Network Scan (osnovno skeniranje mreže). Nakon odabira predloška otvorit će se novi prozor u kojem je potrebno definirati osnovne postavke skeniranja.



Slika 7.3 Basic Network Scan

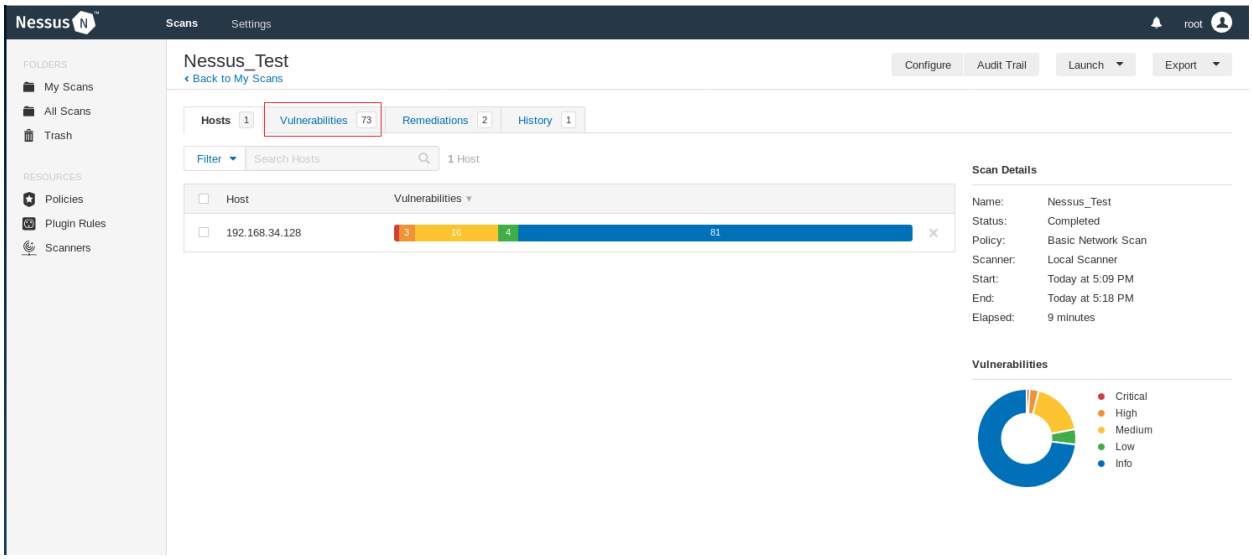
Prije nego li je moguće pokrenuti skeniranje potrebno je definirati naziv, opis i cilj/metu skeniranja(crveni okvir). Međutim kao što je moguće vidjeti na slici 7.3 u lijevom stupcu (plavi okvir), postoji još niz dodatnih postavki. Svaki od tih postavki testeru omogućuje prilagodbu skeniranja kako bi odgovaralo njegovim specifičnim zahtjevima.

Jednom kada su postavke skeniranja konfigurirane potrebno je spremiti predložak klikom na tipku save. Nakon toga će se na početnoj stranici alata pojaviti definirani predložak.



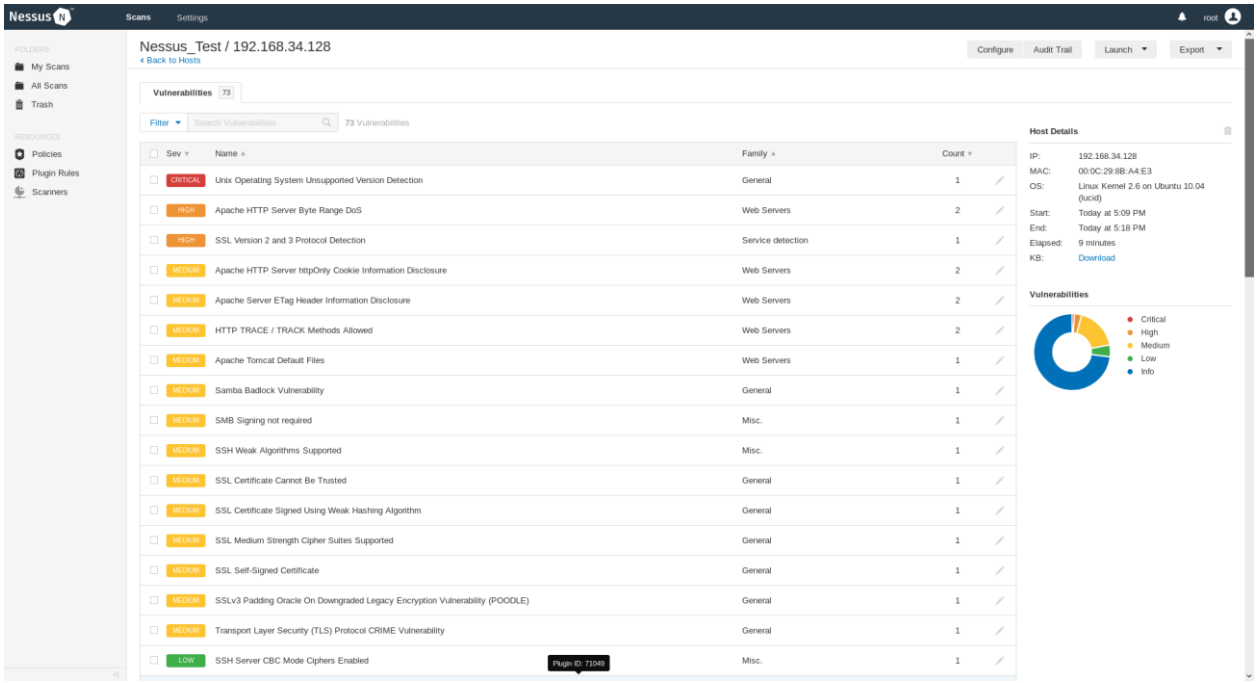
Slika 7.4 Definirani predložak

Klikom na ikonu play (plavi okvir) započinje se skeniranje ciljanog servera.



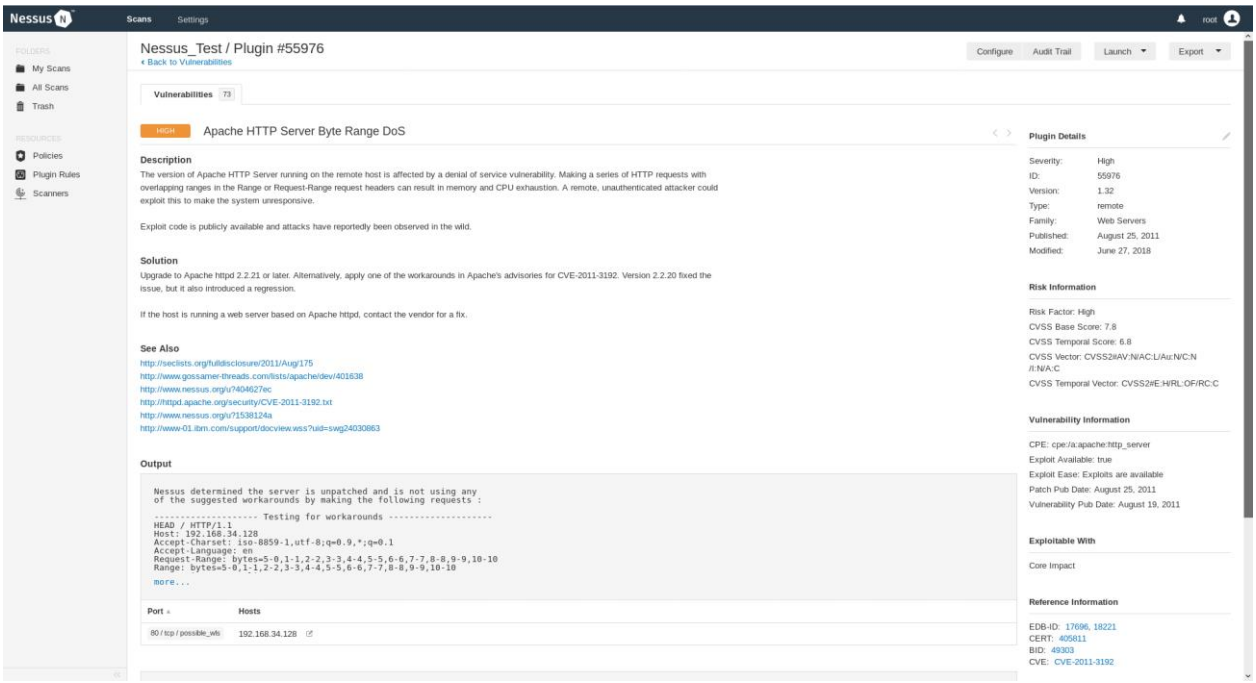
Slika 7.5 Rezultat skeniranja

Iz slike 7.5 vidljivo je da je pronađeno ukupno 73 ranjivosti. Klikom na ip adresu servera (Host kartica) moguće je dobiti detaljniji popis otkrivenih ranjivosti.



Slika 7.6 Popis i sortiranje pronađenih ranjivosti prema stupnju rizika

Klikom na pojedinu ranjivost otvara se novi prozor na kojemu je moguće vidjeti detaljne informacije o ranjivosti.



Slika 7.7 Detaljne informacije o ranjivosti

Ove informacije ne uključuju samo informacije o ranjivosti već i informacije o tome postoji li za njih dostupni exploit-i što penetracijskom testeru uvelike olakšava iskorištavanje ranjivosti .

7.2. Wapiti

[73]Wapiti je komandno linijski alat za automatizirano testiranje ranjivosti web aplikacija. Radi na „black box“ principu što znači da ne proučava izvorni kod web aplikacije već indeksira web stranice koje su dio web aplikacije koja se testira i pritom traži skripte i obrasce unutar kojih je moguće ubaciti podatke.

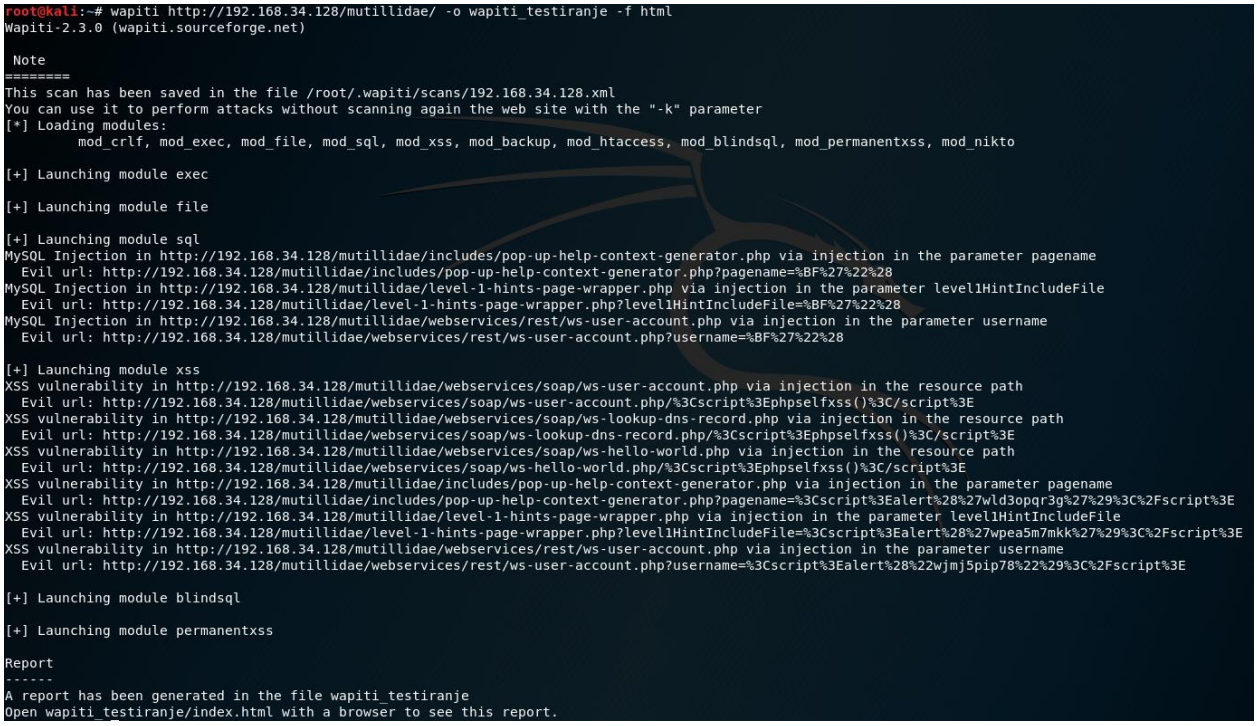
Wapiti može detektirati sljedeće ranjivosti:

- File disclosure (Local and remote include/require, fopen, readfile...)
- Database Injection (PHP/JSP/ASP SQL Injections and XPath Injections)
- XSS (Cross Site Scripting) injection (reflected and permanent)
- Command Execution detection (eval(), system(), passtru()...)
- CRLF Injection (HTTP Response Splitting, session fixation...)
- XXE (XML External Entity) injection

[73] http://wapiti.sourceforge.net/ , 24.8.2018. 19:51:34

- SSRF (Server Side Request Forgery)
- Potencijalno opasne datoteke
- Slabe .htaccess konfiguracijske datoteke koje je moguće zaobići.
- Backup datoteke koje sadrže povjerljive informacije (source code disclosure)
- Shellshock (aka Bash bug)

Za potrebe demonstriranja alata koristi će se OWASP Mutillidae 2 web aplikacija. Prvi korak je započeti skeniranje naredbom wapiti http://192.168.34.128/mutillidae/ -o wapiti_testiranje -f html. Wapiti će skenirati web aplikaciju OWASP Mutillidae 2, a izvještaj o pronađenim ranjivostima će pohraniti u html formatu (-f html opcija) unutar direktorija wapiti_testiranje(-o wapiti_testiranje opcija) pod nazivom index.html.



Slika 7.8 Skeniranje web aplikacije OWASP Mutillidae 2

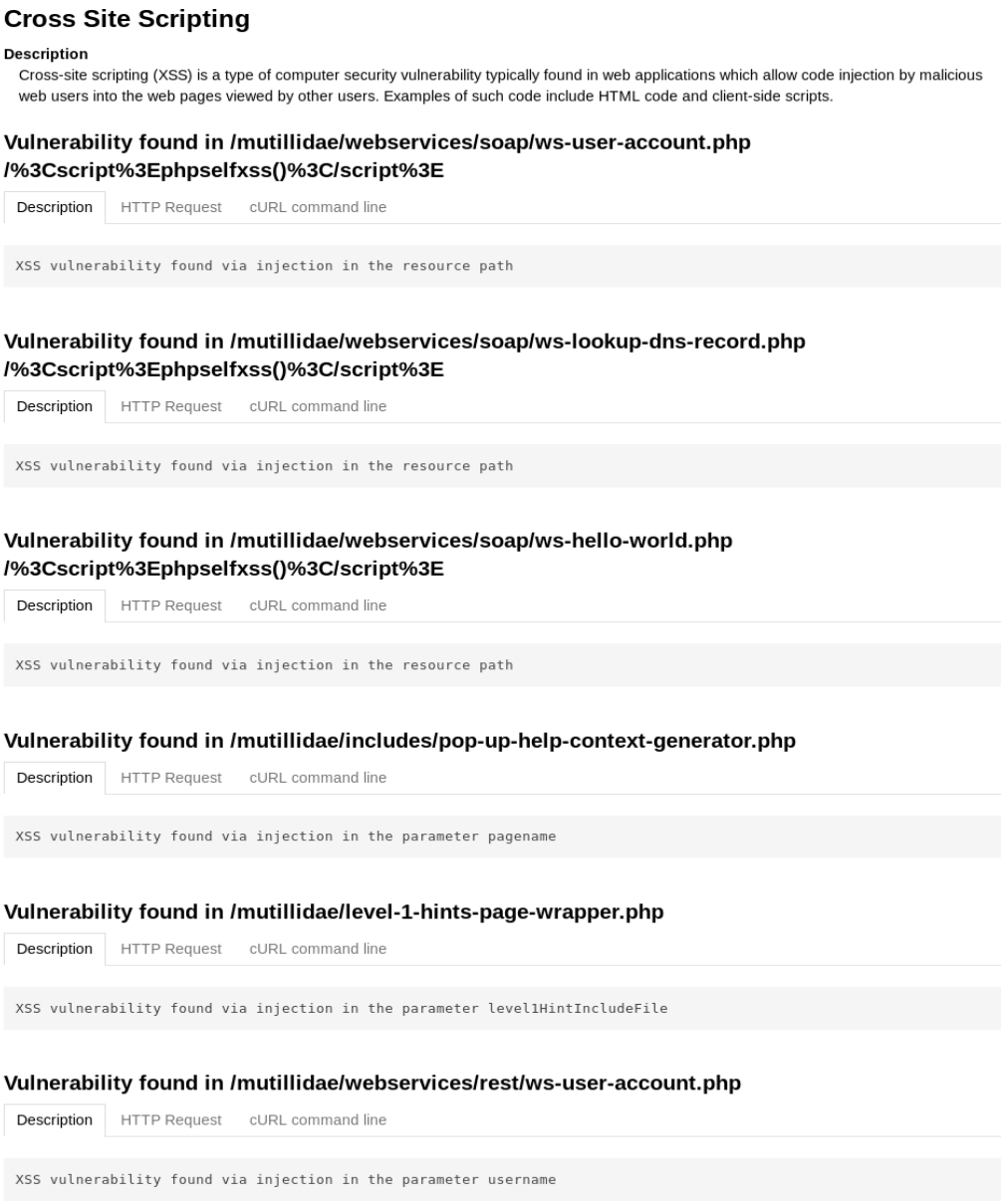
Kada testiranje završi moguće je preko web preglednika otvoriti izvještaj i analizirati pronađene ranjivosti.



Slika 7.9 Izvještaj o pronađenim ranjivostima

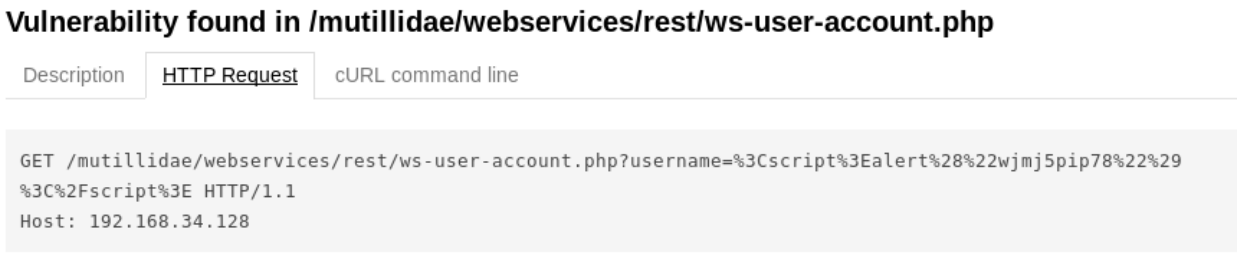
Iz slike 7.9 može se vidjeti da je Wapiti pronašao 6 cross-site scripting (XSS) i 3 SQL injection ranjivosti.

Klikom na pojedinu kategoriju ranjivosti otvara se lista ranjivosti.



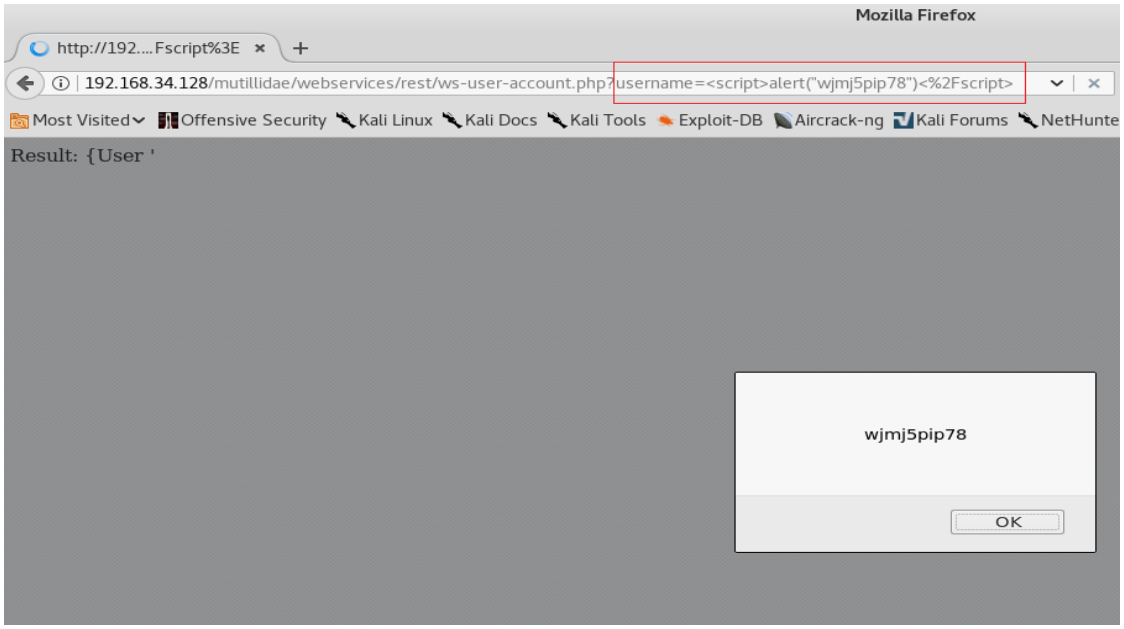
Slika 7.10 Popis pronađenih XSS ranjivosti

Odabirom ranjivosti i klikom na karticu HTTP Request otvara se i zahtjev koji je doveo do otkrivanja ranjivosti.



Slika 7.11 Maliciozni zahtjev

Na primjer ukoliko se kopira url dio tog zahtjeva i zatim zalijepi taj URL u preglednik kao što je prikazano na slici 7.12 rezultat će biti sljedeći:



Slika 7.12 XSS ranjivost

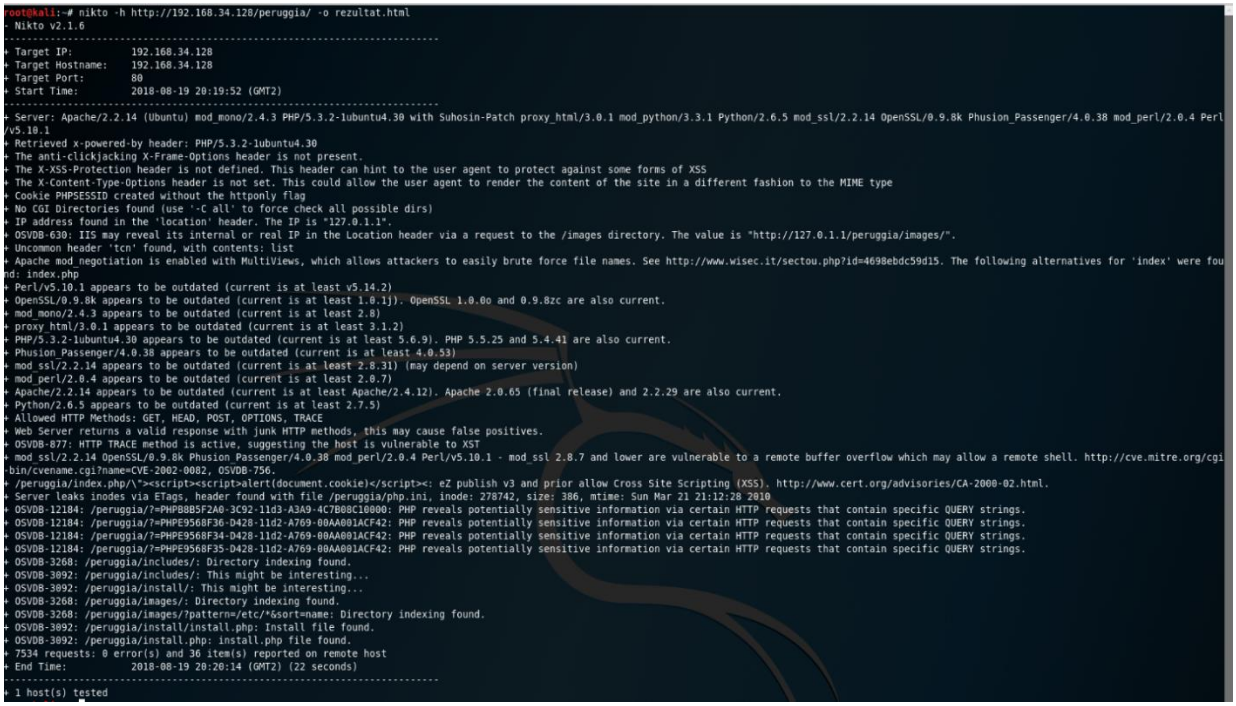
Na slici 7.12 moguće je vidjeti da je web aplikacija ranjiva na XSS ranjivost.

7.3. Nikto

[74]Nikto je sveobuhvatni alat otvorenog koda za automatizirano skeniranje web servera Radi na principu slanja GET i POST zahtjeva serveru, analizira odgovore servera te pritom traži greške u konfiguraciji servera i web aplikacija, nesigurne datoteke, zastarjele aplikacije... .

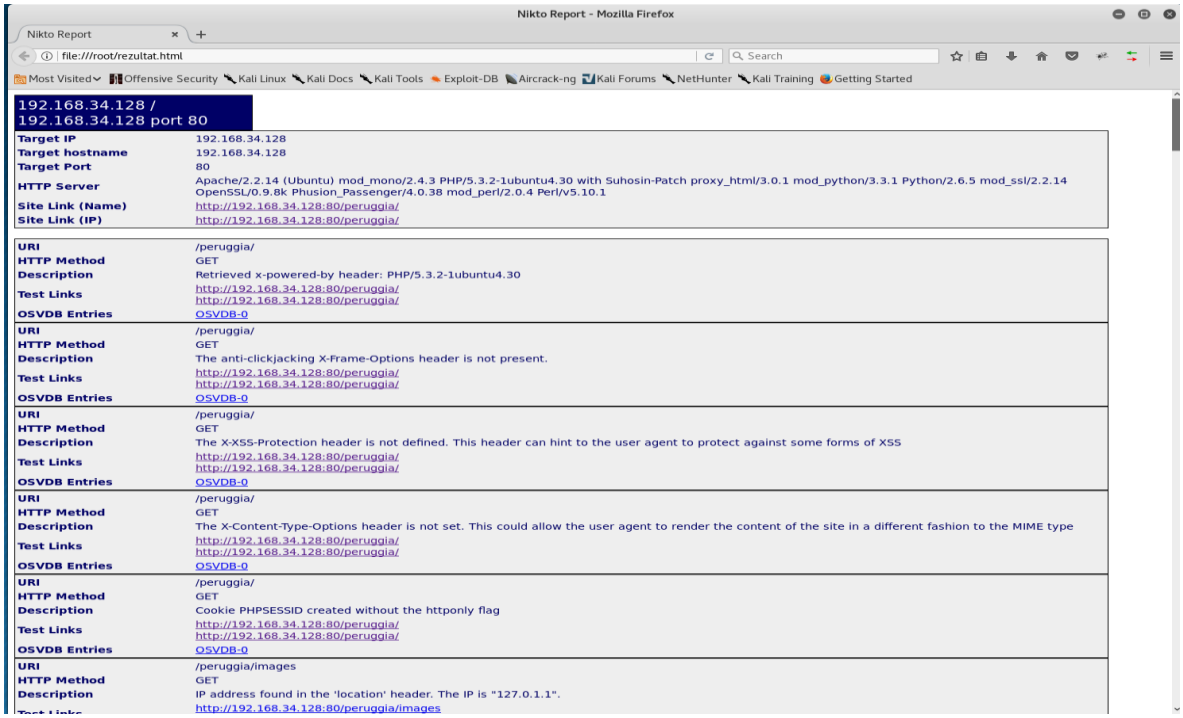
Za potrebe demonstriranja alata koristit će se OWASP Peruggia web aplikacija dok će se rezultati skeniranja tj. Izvještaj pohraniti u html formatu. Naredba kojem se navedeno postiže je nikto -h http://192.168.34.128/peruggia/ -o result.html

[74] https://cirt.net/Nikto2 , 25.8.2018. 13:34:11



Slika 7.13 Skeniranje web aplikacije Peruggia s alatom nikto

Opcija -h definira metu skeniranja dok -o opcija definira lokaciju gdje i u kojem formatu će se pohraniti izvještaj (u ovom primjeru .html). Kada skeniranje završi moguće je otvoriti i analizirati nastali izvještaj.



Slika 7.14 Nikto izvještaj

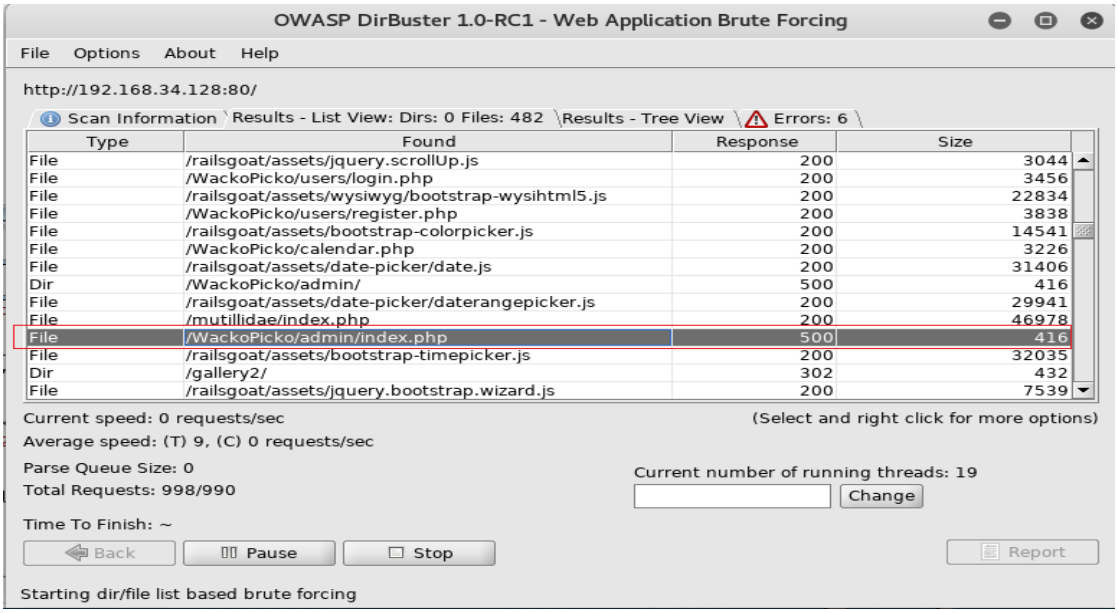
8. Iskorištavanje ranjivosti

U fazi iskorištavanja ranjivosti tester preuzima ulogu malicioznog korisnika tj. napadača i pokušava iskoristiti ranjivosti otkrivene u prethodnoj fazi (identifikacija ranjivosti) kako bi kompromitirao sustav, dobio pristup internoj mreži, osjetljivim podatcima itd. dok u isto vrijeme mora paziti da ne utječu na raspoloživost sustava ili da slučajno ne ostavi sigurnosne rupe koje bi pravi napadač mogao iskoristiti.

8.1. Brute force napad na login stranice s Burp-om

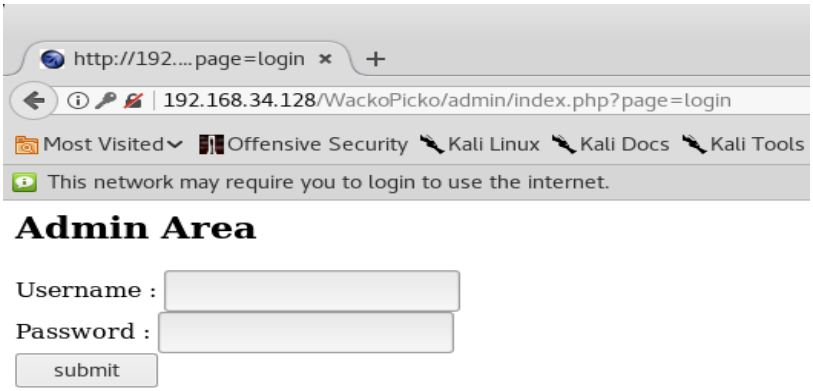
Burp Suite Intruder modul omogućuje korisniku da izvrši fuzzing i bruteforce napade na bilo koji parametar HTTP zahtjeva što je osobito korisno pri izvođenju napada na stranice za prijavu korisnika koristeći rječnik. Prije nego li se krene s demonstracijom napada potrebno je pokrenuti Burp Suite i podesiti u web pregledniku da koristi Burp kao proxy.

Za potrebe demonstriranja ovog napada koristit će se web stranica za prijavu administratora web aplikacije WackoPicko koja je identificirana ranije u fazi izviđanja pomoću alata DirBuster dok će se za brute force napad (pogađanje korisničkog imena i lozinke) koristiti Burp Suite Intruder u kombinaciji s rječnikom koji je također generiran ranije u fazi izviđanja pomoću alata CeWL i JohnTheRipper.



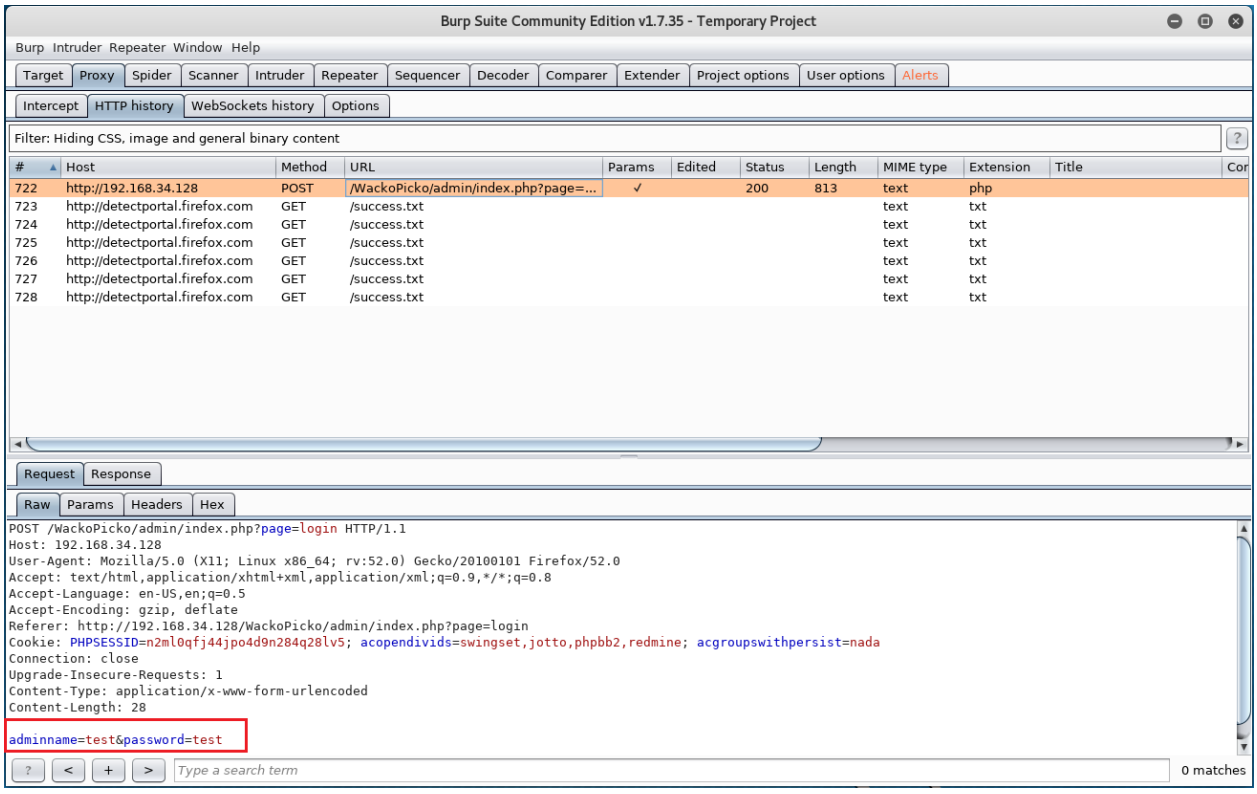
Slika 8.1 Identifikacija portala za prijavu administratora

Ukoliko se navedena poveznica (http://192.168.34.128/WackoPicko/admin/index.php) otvori u web pregledniku web aplikacija će preusmjeriti korisnika na web stranicu za prijavu.



Slika 8.2 Obrazac za prijavu administratora

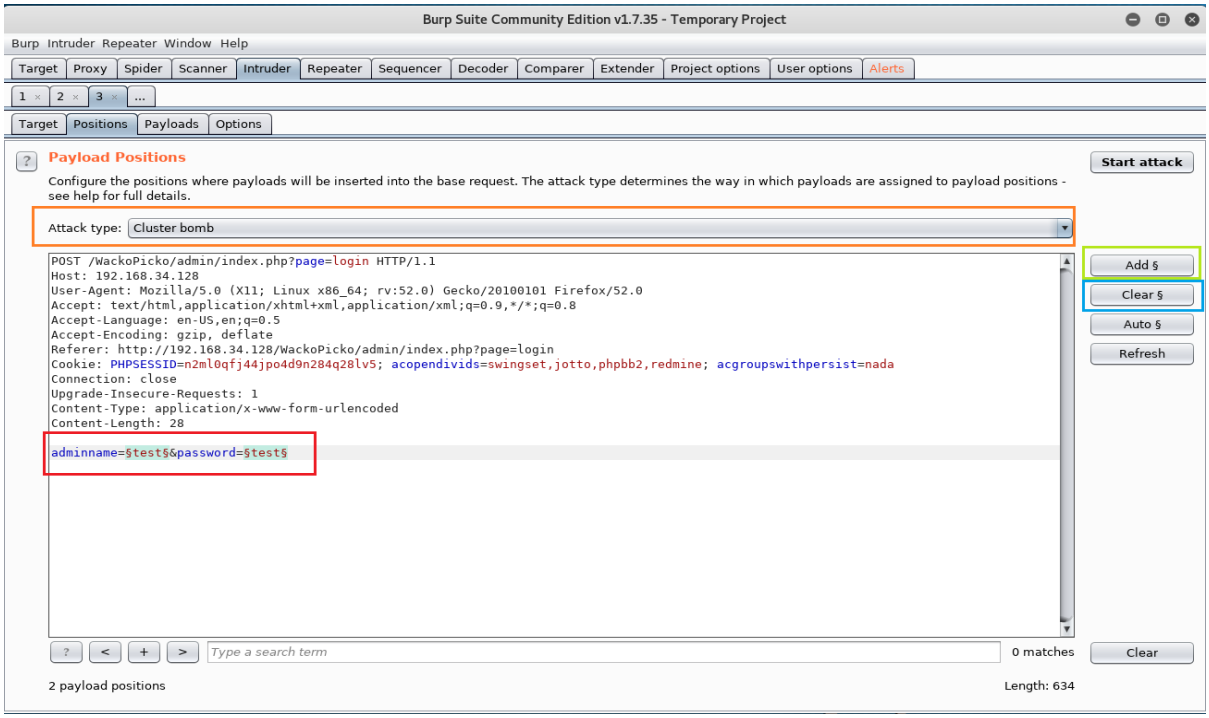
Za početak u oba polja (Username i Password) unijeti će se vrijednost test. Nakon toga potrebno je otići na Burp i na kartici Proxy pa HTTP History pronaći POST zahtjev koji je upravo poslan serveru.



Slika 8.3 POST zahtjev

Nakon toga je navedeni zahtjev potrebno proslijediti Intruder-u što se postiže desnim klikom na zahtjev i odabirom opcije Send to Intruder. Idući korak je otići na karticu intruder, a zatim na karticu Positions (pozicije) kako bi definirali koji parametri u zahtjevu će biti testirani.

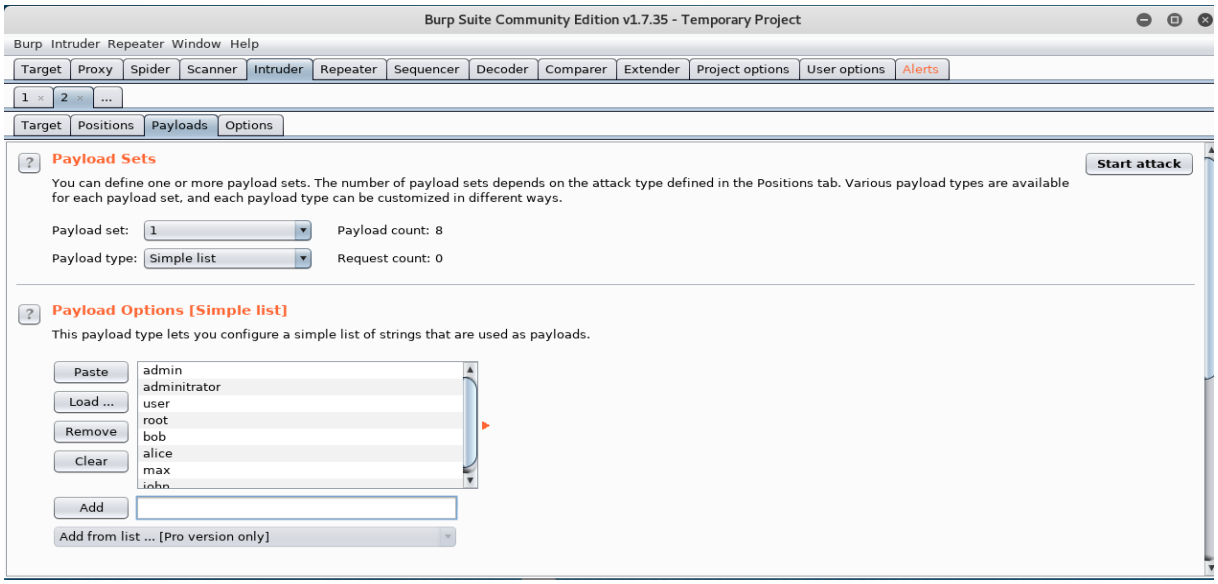
Po zadanim postavkama Burp-a svi parametri koji se mogu testirati će biti označeni međutim kako se u ovom slučaju testiraju samo parametri korisničko ime i lozinka (slika 8.4 crveni okvir) potrebno je prvo sve odznačiti klikom na tipku Clear § (slika 8.4. plavi okvir) te zatim ručno označiti željene parametre i potvrditi selekciju tipkom Add § (slika 8.4 zeleni okvir). [75]Zatim je potrebno odabrati jedan od mogućih tipova napada. U ovom primjeru koristit će se Cluster Bomb napad (slika 8.4 Narančasti okvir) iz razloga jer omogućuje korištenje različitih setova vrijednosti za različita polja (username i password) te testira sve moguće kombinacije vrijednosti zadanih u kartici Payload.



Slika 8.4 Testni parametri

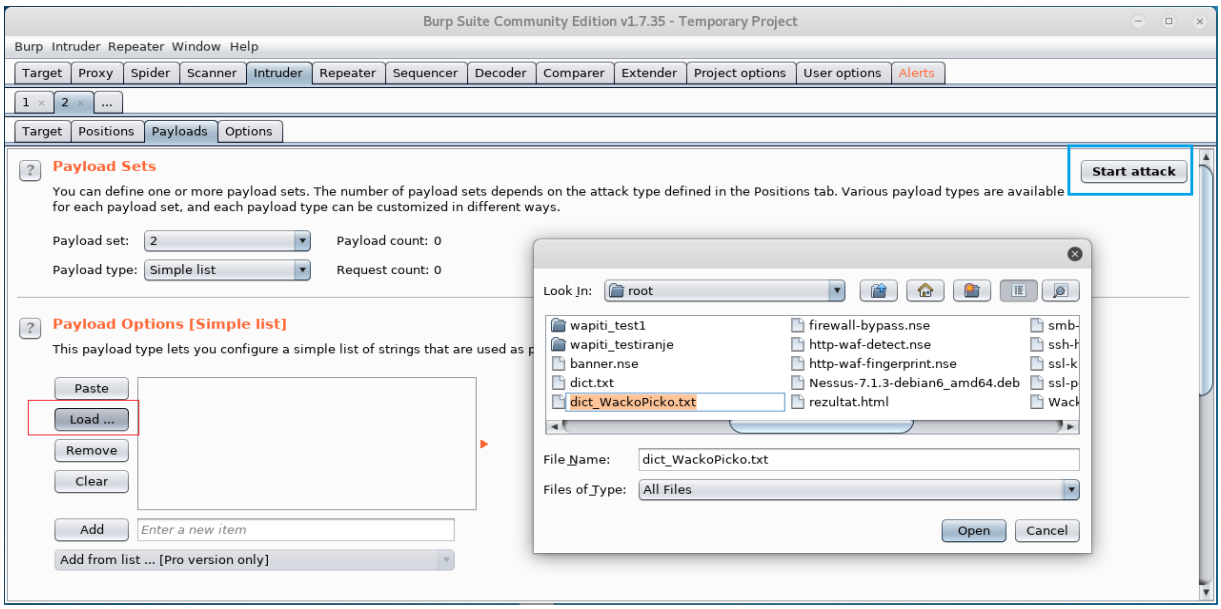
Sljedeći korak je definirati vrijednosti koje će Intruder koristiti za testiranje odabranih parametara za što je potrebno otići na karticu Payload.

[75] https://portswigger.net/burp/documentation/desktop/tools/intruder/positions , 29.8.2018. 19:04:32



Slika 8.5 Definiranje jednostavne liste za polje username (korisničko ime)

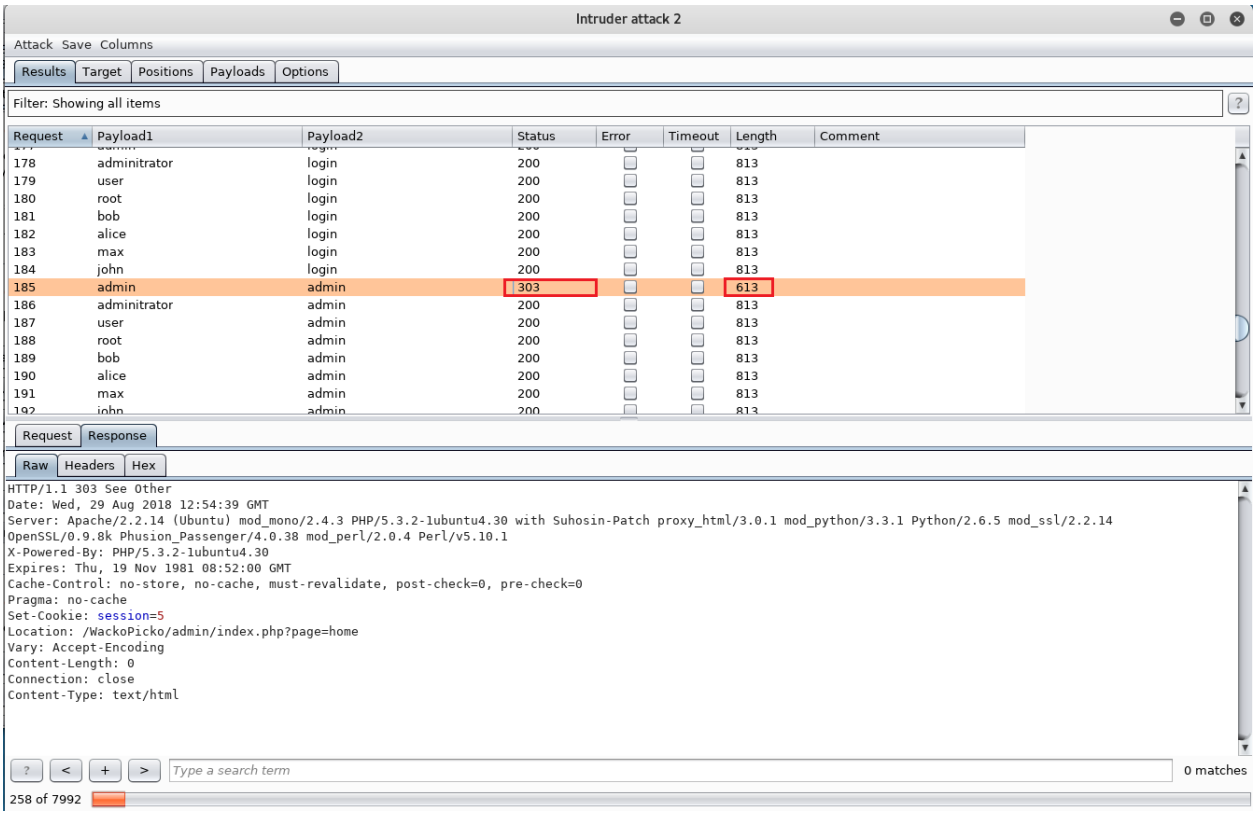
Nakon toga je potrebno odabirati listu 2 iz kartice Payload Set kako bi se definirala lista vrijednosti koje će se koristit za polje password.



Slika 8.6 Definiranje liste za polje password (lozinka)

Za listu 2 koristiti će se ranije generirani rječnik pomoću alata JohnTheRipper. U meniju Payload options potrebno je kliknuti na tipku load i odabrati datoteku tj. rječnik. Sada je sve spremno za napad na stranicu za prijavu. Klikom na tipku Start Attack u gornjem desnom kutu započinje napad.

Otvorit će se novi prozor koji pokazuje napredak napada. Kako bi se identificirala uspješna prijavu, potrebno je pratiti duljinu odgovora web servera.



Slika 8.7 Identificirani administratorski korisnički račun tj. uspješna prijava

Na slici 8.7 može se vidjeti da zahtjev broj 185. ima drugačiju dužinu od ostalih i status 303 koji indicira da je došlo do preusmjeravanja korisnika. Daljnjom analizom odgovora web servera moguće je vidjeti da je korisnik preusmjeren na početnu stranicu za administratore web aplikacije WackoPicko.

9. Zaključak

Sigurnost web aplikacija predstavlja središnju komponentu bilo kojeg poslovanja koje se odvija preko web-a. Neovisno o vrsti djelatnosti kojom se organizacija bavi, ako ta djelatnost uključuje novčano poslovanje i/ili rukovanje s osjetljivim podatcima korisnika potrebno je poduzeti mjere za zaštitu resursa koji su izloženi internetu i čije bi kompromitiranje imalo značajan negativan utjecaj na poslovanje organizacije kako bi se opravdalo povjerenje korisnika tj. klijenata. Penetracijsko testiranje web aplikacija ima značajnu ulogu u tom segmentu iz razloga jer izvještaj koji se generira na kraju pomaže razvojnom timu u poboljšanju sigurnosti web aplikacije. Uz to prilikom samog razvoja web aplikacije dobra je praksa pridržavati se OWASP-ovih se smjernica i dobrih praksi za razvojne programere kako bi web aplikacija u startu bila otporna na „uobičajene“ ranjivosti poput SQLi-a. Isto tako vrlo je važno redovito ažurirati web aplikacije kada su ažuriranja dostupna. Međutim kako penetracijsko testiranje nije jeftin proces na kraju sve se svodi na to koliko organizacije žele riskirati po pitanju sigurnosti web aplikacija s obzirom na to da klijenti izgovor za gubitak važnih ili osobnih podataka klijenata jer sigurnosni standardi nisu ažurirani na web-stranici neće prihvatiti kao valjani izgovor.

Popis slika

Dijagram metodologije penetracijskog testiranja 8

Testiranje statusa servera 27

Identifikacija otvorenih portova na serveru 28

Identifikacija verzije servisa i operacijskog sustava 28

primjer korištenja whois-ip skripte za dobivanje informacija o domeni algebra.hr 29

Testiranje prisutnosti vatrozida, IDS/IPS sustava na serveru koristeći nmap skriptu http-waf-detect 30

Otkrivanje prisutnosti vatrozida, IDS/IPS sustava na serveru 30

Identifikacija vatrozida na serveru 31

Identifikacija vatrozida uz pomoć alata wafw00f 31

Izvorni kod web stranice WackoPicko 32

Analiza polja web stranice koristeći firebug dodatak za preglednike 33

Web stranica nakon izmjene parametra 33

Cookies Manager + ... 34

Primjer izmjene Http only parametra u kolačiću 35

Otkrivanje skrivenih direktorija pomoću robots.txt 35

Sadržaj direktorija cgi-bin 36

Sadržaj direktorija jotto 36

Lista mogućih odgovora za igru jotto 36

DirBuster početni zaslon 37

Pronađene datoteke i direktoriji web servera 38

Lista opcija alata CeWL 39

Lista riječi web aplikacije WackoPicko 39

Lista riječi koja će se koristit za generiranje kompleksnog rječnika 40

Novostvoreni rječnik na temelju zadane liste riječi 41
Analiza zahtjeva sa Burp suite-om 41
Indeksiranje web stranica 42
Otkriveni obrazac za prijavu 43
Zahtjev za prijavu na web stranicu The Bodgeit Store 43
Slanje zahtjeva na Burp Repeater tj. priprema zahtjeva za ponavljanje 44
Ponavljanje zahtjeva 44
Prikaz web stranice u render modu 45
Slanje modificiranog zahtjeva i odgovor servera na zahtjev. 45
Hackbar 47
Izmjena parametra id preko alata Hackbar 48
Greška u aplikaciji 49
Tamper Data 49
Aktivni zahtjevi tj. zahtjevi u tijeku 50
Presretanje i modificiranje zahtjeva 50
Burp Proxy 51
Pogreška o nedopuštenim znakovima u obrascu za unos korisničkog imena i lozinke 51
Presretanje i modificiranje zahtjeva 52
Odgovor web aplikacije na modificirani zahtjev 52
Normalni odgovor web aplikacije 53
Odgovor web aplikacije na unos specijalnih znakova 53
HTML kod web stranice 54
Identificirana XSS ranjivost 54
Analiza nakon testiranja XSS ranjivosti 55
Odgovor aplikacije na normalan upit 56
Odgovor aplikacije na neispravan upit 56

Identificirana SQLi ranjivost 56
Popis korisnika koji postoje u bazi podataka 57
Odgovor aplikacije na neispravan upit. 57
Odgovor aplikacije na „ispravan“ upit. 58
Identificirana blind SQLi ranjivost 58
Parametri kolačića PHPSESSID 59
SSL i TLS informacije o serveru 61
Informacije o podržanim algoritmima za enkripciju komunikacije 61
Informacije o SSL certifikatu 62
Identificirana POODLE ranjivost 63
Nessus početni zaslon 64
Mogućnosti skeniranja ranjivosti 64
Basic Network Scan 65
Definirani predložak 65
Rezultat skeniranja 66
Popis i sortiranje pronađenih ranjivosti prema stupnju rizika 66
Detaljne informacije o ranjivosti 67
Skeniranje web aplikacije OWASP Mutillidae 2 68
Izvještaj o pronađenim ranjivostima 69
Popis pronađenih XSS ranjivosti 70
Maliciozni zahtjev 70
XSS ranjivost 71
Skeniranje web aplikacije Peruggia s alatom nikto 72
Nikto izvještaj 72
Identifikacija portala za prijavu administratora 73
Obrazac za prijavu administratora 74

POST zahtjev 74
Testni parametri 75
Definiranje jednostavne liste za polje username (korisničko ime) 76
Definiranje liste za polje password (lozinka) 76
Identificirani administratorski korisnički račun tj. uspješna prijava 77

Literatura

1. Kali Linux Web Penetration Testing Cookbook, Gilberto Najera-Gutierrez, 2016, ISBN-13: 978-1784392918
2. Kali Linux Network Scanning Cookbook Second Edition, Michael Hixon, Justin Hutchens, 2017, ISBN-13: 978-1787287907
3. Kali Linux 2 – Assuring Security by Penetration Testing Third Edition, Gerard Johansen, Lee Allen , Tedi Heriyanto,, Shakeel Ali , 2016, ISBN-13: 978-1785888427
4. Web Penetration Testing with Kali Linux Second Edition, Juned Ahmed Ansari, 2015, ISBN-13: 978-1783988525
5. Penetration Testing with Kali Linux v1.0.1, Offensive Security, 2014
6. url:https://www.softwaretestinghelp.com/getting-started-with-web-application-penetration-testing/

„Pod punom odgovornošću pismeno potvrđujem da je ovo moj autorski rad čiji niti jedan dio nije nastao kopiranjem ili plagiranjem tuđeg sadržaja. Prilikom izrade rada koristio sam *tuđe materijale navedene u popisu literature ali nisam kopirao niti jedan njihov dio, osim citata za koje sam naveo autora i izvor te ih jasno označio znakovima navodnika. U slučaju da se u bilo kojem trenutku dokaže suprotno, spreman sam snositi sve posljedice uključivo i poništenje javne isprave stečene dijelom i na temelju ovoga rada".*