Automatiziranje penetracijskog testiranja u složenim mrežnim okruženjima

Radojčić, Borna

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 429

AUTOMATIZIRANJE PENETRACIJSKOG TESTIRANJA U SLOŽENIM MREŽNIM OKRUŽENJIMA

Borna Radojčić

Zagreb, lipanj 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 429

AUTOMATIZIRANJE PENETRACIJSKOG TESTIRANJA U SLOŽENIM MREŽNIM OKRUŽENJIMA

Borna Radojčić

Zagreb, lipanj 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Zagreb, 4. ožujka 2024.

DIPLOMSKI ZADATAK br. 429

Pristupnik:	**Borna Radojčić (0036525432)**
Studij:	Računarstvo
Profil:	Računarska znanost
Mentor:	izv. prof. dr. sc. Miljenko Mikuc

Zadatak: **Automatiziranje penetracijskog testiranja u složenim mrežnim okruženjima**

Opis zadatka:

Penetracijsko testiranje (engl. penetration test, pentest ili ethical hacking) je tehnika procjene sigurnosti računalnog sustava ili mreže koja se temelji na oponašanju stvarnog napada. Prilikom testiranja, ovlašteni ispitivač izvodi različite vrste napada jednakim tehnikama koje bi koristio i da je stvarni napadač. Cilj testiranja je pronalaženje ranjivost koje bi se mogle iskoristiti za ostvarenje neovlaštenog pristupa. U početnim fazama testiranja u pravilu se koriste automatizirani alati koji pojednostavnjuje proces procjene sigurnosti te povećavaju učinkovitost i preciznost testiranja što je posebno značajno u velikim i složena mrežnim okruženjima. Vaš je zadatak implementirati sustav za automatizirano penetracijsko testiranje temeljen na Pythonu i alatima distribucije Kali Linux, koji će omogućiti učinkovitu procjenu i izvještavanje o sigurnosti u složenim mrežnim okruženjima. Točnost, učinkovitost i skalabilnost rješenja provjerite u kontroliranim laboratorijskim uvjetima.

Rok za predaju rada: 28. lipnja 2024.

Prije svega, izražavam duboku zahvalnost svojem mentoru, Miljenku Mikucu, na savjetima i velikodušnoj podršci tijekom izrade diplomskog rada, kao i prethodnih seminara i završnog rada. Vaša kompetencija i strpljenje bili su od neprocjenjive vrijednosti za razvoj mojih intelektualnih i profesionalnih sposobnosti.

Također, zahvaljujem svojim roditeljima, Saši i Sanji, na bezuvjetnoj ljubavi, motivaciji i podršci tijekom ovih pet godina studiranja. Bez vaše pomoći, ovaj rad ne bi bio moguć.

Posebno zahvaljujem svim voljenima, bližnjima, prijateljima, kolegama na studiju, kao i kolegama u radu, na razgovorima, raspravama te svim trenucima podrške i razumijevanja.

Na kraju, zahvaljujem svima koji su na bilo koji način doprinijeli mom akademskom putovanju i izradi ovog diplomskog rada. Svaka riječ podrške i svaki savjet imali su veliki utjecaj na moje obrazovanje i razvoj.

Sadržaj

Uvod

Penetracijsko testiranje (engl. penetration test, pentest ili ethical hacking) je tehnika procjene sigurnosti računalnog sustava ili mreže koja se temelji na oponašanju stvarnog napada. Prilikom testiranja, ovlašteni ispitivač izvodi različite vrste napada jednakim tehnikama koje bi koristio i da je stvarni napadač.

Cilj testiranja je pronalaženje ranjivosti koje bi se mogle iskoristiti za ostvarenje neovlaštenog pristupa. U početnim fazama testiranja u pravilu se koriste automatizirani alati koji pojednostavnjuje proces procjene sigurnosti te povećavaju učinkovitost i preciznost testiranja što je posebno značajno u velikim i složena mrežnim okruženjima.

Tema ovog diplomskog rada bilo je automatiziranje penetracijskog testiranja u složenim mrežnim okruženjima. Postignuto je s pomoću skripti u programskom jeziku Python, programa unutar Kali Linux operacijskog sustava, nekoliko programa otvorenog koda i par programa koji se plaćaju, ali su neizostavni dio svake tvrtke koja se bavi penetracijskim testiranjem, to su profesionalna verzija programa Burp Suite i Tenable Vulnerability Management uz njemu pripadni Nessus.

U prvom poglavlju, „Penetracijsko testiranje“, opisan je i objašnjen pojam penetracijskog testiranja. Navedene su vrste s obzirom na poznavanje sustava i s obzirom na ciljeve te vektore napada.

U drugom poglavlju, „Složena mrežna okruženja: fokus na penetracijsko testiranje“, objašnjeno je što su složena mrežna okruženja, kako izgledaju te kako se u njima izvode penetracijska testiranja.

Treće poglavlje, „Automatiziranje penetracijskog testiranja“, objašnjava što je automatsko penetracijsko testiranje, najčešće alate korištene u svrhu automatskog testiranja i usporednu analizu istih.

Zadnje poglavlje, „Automatiziranje penetracijskog testiranja u složenim mrežnim okruženjima“, opisuje program razvijen u sklopu ovog diplomskog rada s pomoću kojeg se izvodi automatsko penetracijsko testiranje u složenim mrežnim okruženjima. Navedeni su svi moduli, odnosno programi u Kali Linuxu i ostali koji su korišteni u izradi konačnog programa.

1. Penetracijsko testiranje

Penetracijsko testiranje ima korijene u ranim fazama razvoja računalne sigurnosti, kada su stručnjaci počeli oponašati napade kako bi otkrili slabosti u sustavima. Tijekom godina, tehnike i alati za penetracijsko testiranje su se razvili, a danas su integralni dio procjene sigurnosti u mnogim organizacijama. Ključna uloga penetracijskog testiranja je omogućiti organizacijama da identificiraju i isprave sigurnosne propuste prije nego što ih zlonamjerni napadači iskoriste. [1]

Penetracijsko testiranje je uže područje računalne sigurnosti, ali svejedno se može podijeliti u nekoliko kategorija na temelju pristupa i opsega informacija koje ispitivač ima o sustavu koji se testira. Svaka vrsta testiranja ima svoje prednosti i izazove, a izbor prave vrste ovisi o ciljevima testiranja i specifičnostima testiranog sustava.

1.1. Vrste s obzirom na poznavanje sustava

Postoje tri glavne vrste penetracijskog testiranja koje se razlikuju prema razini poznavanja testiranog sustava: „Black Box“, „White Box“ i „Gray Box“ testiranja. Izbor vrste testiranja obično određuje klijent, a svaka od njih ima svoje prednosti i nedostatke. [2]

1.1.1. Black Box testiranje

„Black Box“ testiranje podrazumijeva da ispitivač nema nikakve informacije o unutarnjoj strukturi sustava koji se testira. Ovaj pristup simulira vanjski napad, gdje napadač nema nikakvog unutarnjeg uvida u sustav. Cilj je identificirati ranjivosti koje su vidljive izvana.

Jedna od glavnih prednosti „Black Box” testiranja je procjena utjecaja ljudskog faktora na sigurnost. Budući da testeri nemaju nikakve unutarnje informacije, moraju se oslanjati na iste tehnike koje bi koristili stvarni napadači, što uključuje socijalni inženjering i druge metode manipulacije. Ovakvo testiranje može otkriti slabosti u sigurnosnoj svijesti zaposlenika, kao i provjeriti učinkovitost sigurnosnih mjera koje su implementirane kako bi se zaštitile vanjske komponente sustava.

Osim toga, „Black Box” testiranje omogućuje identifikaciju vanjskih ranjivosti koje bi mogle biti iskorištene od strane napadača. Testiranjem se može otkriti koje informacije su dostupne napadačima i kako te informacije mogu biti iskorištene za ulazak u sustav.

Međutim, „Black Box" testiranje također ima svoje nedostatke. Može biti etički i pravno osjetljivo, posebno ako uključuje metode poput socijalnog inženjeringa koje mogu dovesti do neugodnosti ili narušavanja povjerenja unutar organizacije.

Također, „Black Box" testiranje ne pokriva tehničke ranjivosti unutar sustava koje bi bile otkrivene pregledom izvornog koda ili unutarnje arhitekture. Stoga je važno kombinirati „Black Box" testiranje s drugim metodama, kao što su „White Box" i „Gray Box" testiranja, kako bi se dobila sveobuhvatna slika sigurnosnog stanja sustava.

1.1.2. White Box testiranje

„White Box" testiranje uključuje potpuno poznavanje sustava, uključujući izvorni kod, mrežnu arhitekturu i interne sigurnosne kontrole. Ovaj pristup omogućava detaljnu analizu i otkrivanje dubljih ranjivosti.

Jedna od glavnih prednosti „White Box" testiranja je njegova sposobnost da omogući sveobuhvatan pregled sigurnosnih mjera. Budući da tester ima pristup svim unutarnjim informacijama, može detaljno ispitati svaki aspekt sustava i identificirati ranjivosti koje možda ne bi bile otkrivene kroz druge metode testiranja. Ovaj pristup omogućava dubinsku analizu sigurnosnih mehanizama i kontrola, osiguravajući da su sve potencijalne slabosti prepoznate i adresirane.

Također, „White Box" testiranje je vrlo učinkovito u otkrivanju unutarnjih ranjivosti. Testiranjem se može analizirati i provjeriti sigurnost izvornog koda, što je ključno za identificiranje sigurnosnih propusta koje bi zlonamjerni napadači mogli iskoristiti. Detaljan pregled mrežne arhitekture i unutarnjih kontrola omogućava testerima da otkriju potencijalne probleme koji bi mogli ugroziti sigurnost sustava.

Međutim, „White Box" testiranje također ima svoje nedostatke. Može biti vrlo zahtjevno i dugotrajno, jer zahtijeva detaljnu analizu velikih količina informacija i koda. Potrebno je puno vremena i resursa da bi se provelo sveobuhvatno testiranje na ovaj način.

Osim toga, „White Box" testiranje ne simulira stvarne uvjete vanjskog napada, jer vanjski napadači obično nemaju pristup svim unutarnjim informacijama sustava.

Stoga, iako je „White Box" testiranje izuzetno korisno za otkrivanje unutarnjih ranjivosti, ne može zamijeniti potrebu za testiranjem koje simulira stvarne vanjske prijetnje.

1.1.3. Gray Box testiranje

„Gray Box“ testiranje je kombinacija „Black Box“ i „White Box“ pristupa, gdje ispitivač ima ograničene informacije o sustavu. Ovaj pristup simulira napad korisnika s određenim privilegijama ili napadača koji je uspio dobiti djelomične informacije o sustavu.

Jedna od glavnih prednosti „Gray Box“ testiranja je ta što kombinira prednosti oba pristupa. Omogućava dubinsku analizu sličnu „White Box“ testiranju, ali s realističnijim scenarijem napada koji nalikuje situacijama iz stvarnog svijeta, gdje napadači često imaju barem djelomične informacije o sustavu. To omogućava otkrivanje ranjivosti koje bi mogle biti propuštene ako se koristi samo jedan od pristupa.

„Gray Box“ testiranje također pruža realističniji scenarij napada. Simulira situaciju u kojoj napadač ima određene privilegije ili je uspio doći do djelomičnih informacija o sustavu, što je često slučaj u stvarnim napadima. Ovaj pristup može otkriti kako se sustav ponaša pod napadom iznutra, te može identificirati ranjivosti koje nisu vidljive vanjskim napadačima, ali mogu biti iskorištene od strane upućenih osoba ili napadača s određenim razinama pristupa.

Međutim, „Gray Box“ testiranje također ima svoje nedostatke. Može biti manje detaljno od „White Box“ testiranja jer tester nema potpuni uvid u izvorni kod i unutarnju arhitekturu sustava. Osim toga, učinkovitost „Gray Box“ testiranja ovisi o dostupnim informacijama; ako tester ima ograničen ili netočan uvid u sustav, određene ranjivosti mogu ostati neotkrivene.

Kombiniranjem različitih vrsta penetracijskog testiranja s obzirom na poznavanje sustava, odnosno „White Box“, „Black Box” i „Gray Box“ testiranja, organizacije mogu osigurati sveobuhvatnu procjenu svoje sigurnosti. Svaki pristup pruža jedinstvene uvide i pomaže u identifikaciji širokog spektra ranjivosti. Integriranim pristupom, organizacije mogu bolje razumjeti svoje sigurnosne slabosti i implementirati mjere za njihovo otklanjanje, čime značajno poboljšavaju svoju ukupnu sigurnost.

1.2. Vrste s obzirom na ciljeve i vektore napada

Ove vrste penetracijskog testiranja temelje se na različitim aspektima mrežne i sistemske infrastrukture, kao i na metodama napada. One se ne temelje na poznavanju sustava kao prethodna testiranja, već se fokusiraju na specifične ciljeve i vektore napada.

1.2.1. Vanjsko penetracijsko testiranje

Vanjsko penetracijsko testiranje fokusira se na testiranje vanjske mreže i infrastrukture organizacije, kao što su web poslužitelji, vanjski vatrozidi i drugi sustavi povezani na internet. Cilj je identificirati ranjivosti koje napadači mogu iskoristiti s vanjske strane mreže. [1]

Jedna od glavnih prednosti vanjskog penetracijskog testiranja je njegov fokus na vanjske prijetnje. Ovakvo testiranje omogućuje organizacijama da prepoznaju slabosti koje bi mogle biti iskorištene od strane vanjskih napadača, uključujući kibernetičke kriminalce i druge zlonamjerne aktere koji pokušavaju pristupiti mreži izvana. Identifikacijom ovih ranjivosti, organizacije mogu poduzeti potrebne mjere i zaštititi svoje sustave.

Osim toga, vanjsko penetracijsko testiranje simulira stvarni vanjski napad, pružajući organizacijama realističan prikaz kako bi se njihovi sustavi ponašali pri napadu iz vanjskog okruženja. To omogućuje testiranje učinkovitosti postojećih sigurnosnih kontrola, poput vanjskih vatrozida, IDS/IPS sustava (Intrusion Detection/Prevention Systems) i sigurnosnih politika koje su implementirane kako bi zaštitile vanjske resurse.

Međutim, vanjsko penetracijsko testiranje također ima svoje nedostatke. Ne pokriva interne sigurnosne prijetnje, što znači da organizacije moraju kombinirati vanjsko testiranje s internim penetracijskim testiranjem kako bi dobile sveobuhvatnu sliku svog sigurnosnog stanja.

1.2.2. Interno penetracijsko testiranje

Interno penetracijsko testiranje provodi se iz perspektive unutarnjeg korisnika ili napadača koji je već dobio pristup mreži. Cilj ovog testiranja je identificirati ranjivosti unutar mreže koje bi mogle biti iskorištene od strane zlonamjerne upućene osobe ili napadača s pristupom.

Jedna od glavnih prednosti internog penetracijskog testiranja je njegova sposobnost otkrivanja unutarnjih prijetnji i ranjivosti. Ovo testiranje omogućuje organizacijama da prepoznaju slabosti koje bi mogle biti iskorištene od strane zaposlenika, vanjskih suradnika ili napadača koji su već uspjeli probiti vanjske obrambene mjere. Na taj način, organizacije mogu bolje razumjeti potencijalne prijetnje koje dolaze iznutra i poduzeti korake za poboljšanje unutarnje sigurnosti.

Osim toga, interno testiranje pomaže u procjeni sigurnosnih mjera unutar mreže. Omogućuje procjenu učinkovitosti postojećih sigurnosnih kontrola, poput segmentacije mreže, pristupnih prava i sustava za detekciju prijetnji. Identifikacijom slabosti unutar mreže, organizacije mogu poduzeti mjere za jačanje svojih sigurnosnih mjera i smanjenje rizika od unutarnjih napada.

Međutim, interno penetracijsko testiranje također ima svoje nedostatke, slično kao u ranijem odlomku, u ovom slučaju vanjske prijetnje nisu pokrivene što znači da bi organizacije trebale kombinirati ta dva testa.

1.2.3. Testiranje web aplikacija

Testiranje web aplikacija fokusira se na sigurnosne aspekte web aplikacija. Ovo testiranje pronalazi ranjivosti specifične za web aplikacije, kao što su SQL injekcija, XSS (Cross-Site Scripting), CSRF (Cross-Site Request Forgery) i druge. [1]

Jedna od glavnih prednosti testiranja web aplikacija je njegova upućenost na web aplikacije. Ovakvo testiranje omogućuje identifikaciju specifičnih ranjivosti koje se često javljaju u web okruženjima. Na primjer, SQL injekcija može omogućiti napadačima da izvrše zlonamjerne SQL naredbe unutar baze podataka aplikacije, dok XSS omogućava ubacivanje zlonamjernog koda u web stranice koje pregledavaju korisnici. CSRF napadi, s druge strane, omogućuju napadačima da izvrše neovlaštene radnje u ime korisnika. Identifikacijom i otklanjanjem ovih ranjivosti, organizacije mogu značajno poboljšati sigurnost svojih web aplikacija.

Međutim, testiranje web aplikacija također ima svoje nedostatke. Ne pokriva druge aspekte mrežne sigurnosti, kao što su sigurnost mrežne infrastrukture ili sigurnost drugih aplikacija unutar iste mreže. Zbog toga je važno da testiranje web aplikacija bude dio šire strategije sigurnosne procjene koja uključuje i druge vrste testiranja.

Integriranjem različitih pristupa, organizacije mogu osigurati sveobuhvatnu zaštitu svojih digitalnih resursa, pokrivajući sve potencijalne ranjivosti i prijetnje.

1.2.4. Testiranje mobilnih aplikacija

Testiranje mobilnih aplikacija uključuje procjenu sigurnosti kako klijentske tako i serverske strane aplikacija. Ovaj tip testiranja fokusira se na logiku aplikacije i potencijalne ranjivosti koje bi mogle biti iskorištene.

Testiranjem se mogu otkriti ranjivosti unutar mobilnog okruženja, uključujući probleme s autentifikacijom, autorizacijom, pohranom podataka i sigurnošću komunikacija između klijentske i serverske strane aplikacije. Na taj način, organizacije mogu osigurati da su njihove mobilne aplikacije zaštićene od potencijalnih napada i da ispravno upravljaju povjerljivim informacijama korisnika.

Međutim, testiranje mobilnih aplikacija također ima svoje nedostatke. Ne pokriva druge aspekte mrežne sigurnosti, poput sigurnosti infrastrukture mreže ili sigurnosti drugih povezanih sustava. Zbog toga je važno kombinirati testiranje mobilnih aplikacija s drugim vrstama sigurnosnih procjena kako bi se dobila sveobuhvatna slika sigurnosti organizacije. Kroz integrirani pristup, organizacije mogu identificirati i otkloniti širok spektar ranjivosti, osiguravajući visoku razinu zaštite za sve svoje digitalne resurse.

1.2.5. Testiranje bežičnih mreža

Testiranje bežičnih mreža fokusira se na sigurnost bežičnih mreža. Ovaj tip testiranja procjenjuje sigurnost Wi-Fi mreža, uključujući enkripcijske protokole poput WEP, WPA i WPA2, te sigurnosne postavke pristupnih točaka i drugih bežičnih uređaja. [8]

Jedna od glavnih prednosti testiranja bežičnih mreža je njegova specifičnost za bežične mreže. Ovakvo testiranje omogućuje detaljno ispitivanje ranjivosti specifičnih za bežične komunikacije, kao što su napadi na enkripcijske protokole i neadekvatno osigurane pristupne točke. Testiranjem se mogu identificirati potencijalne slabosti koje bi napadači mogli iskoristiti za neovlašteni pristup mreži ili presretanje komunikacija.

Međutim, testiranje bežičnih mreža također ima svoje nedostatke. Ne pokriva žičane mreže i druge aspekte sigurnosti unutar IT infrastrukture.

To znači da, iako može pružiti vrijedne uvide u sigurnost bežičnih komunikacija, mora biti dopunjeno testiranjem žičanih mreža i drugih komponenti sustava kako bi se dobila sveobuhvatna procjena sigurnosnog stanja organizacije.

1.2.6. Socijalni inženjering

Socijalni inženjering testira sposobnost organizacije da se obrani od napada koji koriste manipulaciju i ljudsku interakciju kako bi došli do povjerljivih informacija. Ovi napadi mogu uključivati *phishing* napade, telefonske prevare i druge tehnike socijalnog inženjeringa. [4]

Cilj je procijeniti koliko su zaposlenici svjesni sigurnosnih prijetnji i kako reagiraju na pokušaje manipulacije.

Jedna od glavnih prednosti socijalnog inženjeringa u penetracijskom testiranju je utjecaj ljudskog faktora na sigurnost. Ovakvo testiranje otkriva slabosti u sigurnosnoj svijesti zaposlenika, pomažući organizacijama da identificiraju područja gdje je potrebno dodatno obrazovanje i obuka. Kroz simulacije stvarnih napada, organizacije mogu vidjeti koliko su njihovi zaposlenici spremni prepoznati i pravilno reagirati na pokušaje socijalnog inženjeringa.

Međutim, socijalni inženjering u testiranju sigurnosti ima i svoje nedostatke. Može biti etički i pravno osjetljivo, jer uključuje manipulaciju stvarnih ljudi, što može dovesti do neugodnosti ili narušavanja povjerenja unutar organizacije. Također, ovakvo testiranje ne pokriva tehničke ranjivosti sustava, što znači da mora biti dopunjeno tehničkim testiranjem kako bi se dobila cjelovita slika sigurnosnog stanja.

Različite vrste penetracijskog testiranja omogućavaju sveobuhvatan pristup procjeni sigurnosti sustava i mreža. Izbor prave vrste testiranja ovisi o specifičnim potrebama i ciljevima organizacije, a često je korisno koristiti kombinaciju više vrsta testiranja kako bi se dobila cjelovita slika sigurnosnog stanja.

1.3. Faze penetracijskog testiranja

Penetracijsko testiranje je proces koji se sastoji od četiri faze. Svaka faza ima ključnu ulogu u identificiranju i iskorištavanju sigurnosnih ranjivosti unutar sustava ili mreže, a te faze su izviđanje, skeniranje, iskorištavanje i izvještavanje.

1.3.1. Izviđanje

Izviđanje (engl. reconnaissance), također poznato kao faza prikupljanja informacija, prva je i jedna od najvažnijih faza penetracijskog testiranja. Cilj ove faze je prikupiti što više informacija o ciljanom sustavu ili mreži kako bi se identificirali potencijalni napadi. [4]

Postoje dvije vrste prikupljanja informacija:

- **Pasivno izviđanje**
    - Metode koje ne uključuju direktnu interakciju s ciljem, poput pretraživanja javno dostupnih informacija (npr. putem Google pretraživanja, društvenih mreža, WHOIS upita) kako bi se identificirali domena, IP adrese, i druge bitne informacije.
- **Aktivno izviđanje**
    - Uključuje metode koje uključuju direktnu interakciju s ciljem, poput pinganja IP adresa, pretraživanja otvorenih pristupa, i prikupljanja informacija o verzijama softvera koji se koristi.

1.3.2. Skeniranje

Nakon faze prikupljanja informacija, dolazi faza skeniranja. U ovoj fazi, penetracijski tester koristi alate za skeniranje kako bi dobio detaljnije informacije o ciljanom sustavu ili mreži. Skeniranje pomaže u identifikaciji otvorenih pristupa, aktivnih servisa, operacijskih sustava i topologije mreže. [5]

Najčešća skeniranja su:

- **Skeniranje pristupa**
    - Otkriva koji su pristupi otvoreni i koje usluge slušaju na njima.
- **Skeniranje ranjivosti**
    - Identificira poznate ranjivosti u uslugama koje su otkrivene tijekom skeniranja pristupa.

- **Detekcija verzija**
    - Utvrđuje verzije operacijskog sustava i softvera koji se koristi kako bi se identificirale specifične ranjivosti.

1.3.3. Iskorištavanje

Faza iskorištavanja uključuje aktivno iskorištavanje identificiranih ranjivosti kako bi se dobio neovlašteni pristup sustavu. Cilj ove faze je demonstrirati stvarne prijetnje koje bi mogle ugroziti sigurnost sustava. U ovom koraku, penetracijski tester koristi specifične programe koji iskorištavaju ranjivosti (engl. exploit) kako bi dobio pristup povjerljivim informacijama ili kontrolu nad sustavom.

Nakon što tester dobije inicijalni pristup sustavu, sljedeći korak je povećanje privilegija. U ovoj fazi, tester pokušava povećati svoje privilegije unutar sustava kako bi dobio potpunu kontrolu. Ovo može uključivati različite tehnike, kao što su iskorištavanje dodatnih ranjivosti ili korištenje ukradenih vjerodajnica, kako bi se proširio opseg pristupa i omogućilo daljnje iskorištavanje sustava.

Posljednja komponenta ove faze je eksfiltracija podataka. Tester prikuplja i izvlači osjetljive podatke iz ciljanog sustava. Ovo može uključivati osobne podatke, financijske informacije, poslovne tajne ili bilo koje druge vrste povjerljivih informacija koje mogu biti vrijedne napadačima. Cilj je demonstrirati kako bi stvarni napadači mogli pristupiti i ukrasti osjetljive podatke, što predstavlja ozbiljan rizik za organizaciju.

Ova faza iskorištavanja pokazuje koliko su stvarne prijetnje opasne i naglašava potrebu za sigurnosnim mjerama koje mogu spriječiti ili ublažiti takve napade.

1.3.4. Izvještavanje

Završna faza penetracijskog testiranja je izvještavanje (engl. reporting). U ovoj fazi, penetracijski tester sastavlja detaljan izvještaj o pronađenim ranjivostima, metodama korištenim za njihovo otkrivanje i iskorištavanje, te preporukama za njihovo otklanjanje. Izvještaj treba biti jasan, precizan i koristan za tehničku i menadžersku publiku.

Prvi dio izvještaja uključuje opis ranjivosti. Tester pruža detaljan opis svake identificirane ranjivosti, uključujući njen utjecaj i rizik.

Ovaj dio izvještaja treba jasno pokazati potencijalne posljedice ako se ranjivost ne otkloni, čime se menadžerima pomaže u razumijevanju ozbiljnosti problema.

Drugi dio izvještaja odnosi se na metodologiju testiranja. Ovdje se opisuje koje su metode i alati korišteni tijekom testiranja. Detaljan prikaz koraka koje je tester poduzeo omogućuje tehničkom timu da razumije kako su ranjivosti otkrivene, te može poslužiti kao vodič za buduće testiranje ili interne provjere sigurnosti.

Treći dio izvještaja pokriva iskorištavanje. Ovdje tester pruža detalje o tome kako su ranjivosti iskorištene. Opisuje se način na koji su napadi izvedeni, što je postignuto iskorištavanjem i kako bi stvarni napadači mogli iskoristiti te slabosti. Ovaj dio je ključan za tehničko osoblje koje treba razumjeti praktične aspekte napada kako bi učinkovito otklonilo ranjivosti.

Četvrti dio izvještaja sadrži preporuke. Tester daje praktične preporuke za otklanjanje ranjivosti i poboljšanje sigurnosti sustava.

Preporuke su često specifične za svaku ranjivost i mogu uključivati prijedloge za ažuriranja softvera, promjene konfiguracije ili dodatne sigurnosne kontrole. Ovaj dio izvještaja je posebno koristan menadžerima i tehničkim timovima koji su odgovorni za implementaciju sigurnosnih mjera.

Posljednji dio izvještaja uključuje dodatnu dokumentaciju. Ova dokumentacija može sadržavati snimke zaslona, zapise skeniranja i druge relevantne podatke koji podržavaju nalaze testera. Dokumentacija pomaže u potvrđivanju otkrića i pruža dodatne informacije koje mogu biti korisne za daljnju analizu i praćenje sigurnosnih mjera.

Izvještavanje je ključna komponenta penetracijskog testiranja jer osigurava da svi relevantni dionici razumiju nalaze testiranja i poduzmu potrebne korake za poboljšanje sigurnosti sustava. Kvalitetan izvještaj pomaže organizacijama da bolje zaštite svoje podatke i infrastrukturu od potencijalnih napada.

2. Složena mrežna okruženja: fokus na penetracijsko testiranje

Složena mrežna okruženja u računalnim znanostima predstavljaju sofisticiran i međusobno povezan sustav uređaja. Ove mreže imaju karakteristike poput dinamične topologije, heterogenih čvorova i različitih protokola, što ih čini izazovnima za osiguranje i testiranje. [7]

Penetracijsko testiranje u takvim okruženjima zahtijeva razumijevanje njihovih jedinstvenih karakteristika kako bi se identificirale ranjivosti i ojačala sigurnost.

Jedna od ključnih karakteristika složenih mrežnih okruženja je njihova dinamična topologija. Čvorovi i veze unutar složenih mreža često se mijenjaju zbog različitih faktora poput mobilnosti čvorova, dinamičkog dodjeljivanja IP adresa i promjene mrežnih konfiguracija. Ova dinamika komplicira tradicionalne sigurnosne mjere i zahtjeva adaptivne tehnike penetracijskog testiranja za identifikaciju ranjivosti kako se pojavljuju.

Složene mreže također obuhvaćaju širok raspon uređaja, uključujući tradicionalna računala, mobilne uređaje, IoT uređaje i resurse u oblaku. Ovi uređaji mogu koristiti različite komunikacijske protokole, što dodatno komplicira strukturu i sigurnost mreže. Osim dinamične prirode i heterogenih čvorova, složene mreže karakterizira i visoka povezanost. Čvorovi su visoko međusobno povezani i moguće je da većina čvorova ima malo veza, a neki čvorovi (čvorišta) vrlo velik broj veza.

Visoko povezani čvorovi su kritične točke kvara i visoko vrijedne mete za napadače, što zahtijeva fokusirane sigurnosne mjere.

Penetracijsko testiranje u složenim mrežnim okruženjima uključuje simulaciju kibernetičkih napada za identifikaciju i iskorištavanje ranjivosti unutar mreže. Ovaj proces je ključan za procjenu sigurnosnog stanja i otpornosti mreže. Penetracijsko testiranje mora pokriti cijelu mrežu, uključujući sve vrste uređaja, komunikacijskih protokola i tokova podataka. Primarni ciljevi uključuju identificiranje sigurnosnih rupa, testiranje učinkovitosti postojećih sigurnosnih kontrola i pružanje konkretnih preporuka za smanjenje rizika. [2]

Tehnike i alati za penetracijsko testiranje uključuju automatizirane skenere poput Nmapa i Nessusa, koji mogu automatizirati otkrivanje mrežnih uređaja i poznatih ranjivosti. [1]

Iskusni testeri ručno istražuju mrežu kako bi identificirali suptilne ranjivosti koje mogu promaknuti automatiziranim alatima. Alati poput Metasploita olakšavaju iskorištavanje otkrivenih ranjivosti i procjenu njihovog potencijalnog utjecaja.

Izazovi u složenim mrežama uključuju stalno mijenjanje topologije koje zahtijeva kontinuirano praćenje i testiranje kako bi se pratili novi problemi. Testiranje mora biti skalabilno kako bi učinkovito obradilo veliki broj čvorova i veza.

Različiti uređaji i protokoli zahtijevaju specijalizirano znanje i alate za učinkovito testiranje. [1]

Ključne strategije uključuju izradu detaljne karte mrežne topologije kako bi se razumjela struktura i identificirali kritični čvorovi. Testiranje učinkovitosti mrežne segmentacije za ograničavanje proboja i smanjenje lateralnog kretanja napadača također je važno. Implementacija kontinuiranih praksi testiranja, uključujući automatizirane i planirane penetracijske testove, ključna je za održavanje sigurnosti tijekom vremena.

Penetracijsko testiranje u složenim mrežnim okruženjima ključna je komponenta suvremenih strategija kibernetičke sigurnosti.

Dinamična, heterogena i visoko povezana priroda ovih mreža predstavlja jedinstvene izazove koji zahtijevaju specijalizirane tehnike i alate. Razumijevanjem karakteristika ovih okruženja i primjenom sveobuhvatnih, adaptivnih metoda penetracijskog testiranja, organizacije mogu bolje zaštititi svoje mreže od kibernetičkih prijetnji.

Stalna evolucija mrežnih tehnologija i sve sofisticiraniji kibernetički napadi zahtijevaju proaktivan pristup penetracijskom testiranju i mrežnoj sigurnosti.

3. Automatiziranje penetracijskog testiranja

Tradicionalno, penetracijsko testiranje bio je ručni i dugotrajan proces koji zahtijeva vješte sigurnosne stručnjake. Međutim, s brzim porastom cyber prijetnji i složenosti IT okruženja, automatsko penetracijsko testiranje pojavilo se kao održivo rješenje za poboljšanje učinkovitosti i efektivnosti. Ovaj rad pruža detaljnu analizu automatskog penetracijskog testiranja, uključujući metodologije, alate, prednosti i ograničenja, na temelju različitih istraživačkih studija i praktičnih implementacija.

Ručno penetracijsko testiranje uključuje testere koji simuliraju napade na sustav. Ovaj pristup je vrlo prilagodljiv i može se primijeniti na složene scenarije, ali ima značajne nedostatke. Ručno testiranje može trajati znatno vrijeme, osobito za velike sustave, te zahtijeva visoko kvalificirane stručnjake, što može biti skupo. Osim toga, rezultati mogu varirati ovisno o iskustvu i stručnosti testera, što dovodi do nekonzistentnosti u procjenama sigurnosti.

Automatsko penetracijsko testiranje koristi softverske alate za izvođenje testova, nudeći nekoliko prednosti u odnosu na ručno testiranje. Automatski alati mogu brzo skenirati sustave i identificirati ranjivosti, čime se značajno povećava učinkovitost procesa. Pruža standardizirane rezultate, smanjujući varijabilnost koja postoji u ručnom testiranju, te smanjuje potrebu za velikim ljudskim resursima, čime se smanjuju troškovi.

Postoji nekoliko alata koji se često koriste u automatskom penetracijskom testiranju, svaki s posebnim mogućnostima. Na primjer, Nmap se koristi za skeniranje otvorenih pristupa na uređaju i pronalaženje raznih informacija o njima, Nessus nudi skeniranje i nalaženje svih ranjivosti vezanih uz uređaj, slično radi i Burp Suite, ali s fokusom na web aplikacije. Metasploit je široko korišten alat i okvir (framework) koji olakšava otkrivanje ranjivosti i njihovo iskorištavanje, a još jedan popularan alat za iskorištavanje je SQLMap, koji automatizira napade SQL injekcije.

Kao primjer implementacije uzeo sam Net-Nirishak, Net-Nirikshak 1.0 je automatski alat razvijen za olakšavanje procjene ranjivosti i penetracijskog testiranja specifično za indijske banke. Ovaj alat radi kroz pet faza: prikupljanje informacija, skeniranje, otkrivanje ranjivosti, iskorištavanje i izvještavanje.

Na taj način omogućuje visoku učinkovitost u detekciji ranjivosti i generiranju akcijskih izvješća, čineći ga pouzdanim izborom za redovne sigurnosne revizije u bankama. [4]

Usporedna analiza različitih automatskih penetracijskih alata ističe njihove prednosti i nedostatke. Na primjer, alati poput Net-Nirikshak 1.0 i Metasploit nude brze skenove i mogućnosti iskorištavanja, dok Nessus pruža sveobuhvatnu detekciju ranjivosti preko različitih protokola i usluga. Alati s korisničkim sučeljem i automatiziranim mogućnostima, kao što su Burp Suite i Nessus, lakši su za korištenje od strane ne-stručnjaka.

Unatoč prednostima, automatsko penetracijsko testiranje ima neka ograničenja. Automatski alati mogu proizvesti lažne pozitivne ili propustiti određene ranjivosti. Također, automatski alati možda neće u potpunosti razumjeti kontekst sustava, što može dovesti do nepotpunih procjena. Osim toga, alati se oslanjaju na ažurirane baze ranjivosti; zastarjele baze mogu rezultirati propuštenim ranjivostima.

Automatsko penetracijsko testiranje nudi snažan način za poboljšanje sigurnosnih procjena pružajući konzistentnu, učinkovitu i ekonomičnu detekciju ranjivosti. Iako ne može u potpunosti zamijeniti ručno testiranje, značajno poboljšava sposobnost organizacije da održava robustan sigurnosni položaj. Kontinuirana poboljšanja u alatima i metodologijama automatiziranja dodatno će učvrstiti njihovu ulogu u kibernetičkoj sigurnosti.

4. Automatiziranje Penetracijskog Testiranja u Složenim Mrežnim Okruženjima

U ovom poglavlju detaljno ću opisati implementaciju automatiziranog penetracijskog testiranja u složenim mrežnim okruženjima. Fokusirat ću se na metodologiju, korištene alate, izazove s kojima sam se susreo te rezultate testiranja. Cilj ovog poglavlja je prikazati kako se koriste različiti alati i tehnike za automatiziranje procesa penetracijskog testiranja te kako se svi ti elementi sklapaju u jedan cjelovit sustav.

Program je razvijen u programskom jeziku Python i koristi razne alate iz Kali Linux distribucije kako bi automatizirao sve ključne faze penetracijskog testiranja. Ovo automatiziranje značajno olakšava posao penetracijskom testeru, čineći proces bržim i jednostavnijim.

Ime razvijenog programa je APTuSMO (Automatizacija Penetracijskog Testiranja u Složenim Mrežnim Okruženjima), on udružuje puno različitih programa unutar Kali Linuxa, ali i druge programe. Programi navedeni u poglavljima 4.2 i 4.3. čine module glavnog programa APTuSMO.

```
Program/
├── APTuSMO.py
├── Report/
│   ├── Report.py
├── Exploiting/
│   ├── sqlmap.py
│   ├── metasploit.py
│   ├── Exploting.py
│   ├── hydra.py
│   ├── etc…
├── Scanning/
│   ├── Scanning.py
│   ├── InfoGather/
│   ├── Endpoint_Security/
│   │   ├── endpoint.py
│   │   ├── permissions.py
│   │   ├── malware.py
│   │   ├── etc…
│   ├── Scanners/
│   │   ├── burp_scan.py
│   │   ├── nessus_scan.py
│   │   ├── nikto.py
│   │   ├── nmap_scan.py
│   │   ├── etc…
```

Rezultat 1. Struktura programa APTuSMO razvijenog u sklopu diplomskog rada

Program će biti dostupan na GitHubu, gdje će svi zainteresirani moći preuzeti i koristiti najnovije verzije. Poveznica do repozitorija je ovdje.

4.1. Pokretanje programa

Prvi korak penetracijskog testiranja je pokretanje programa, odnosno prikupljanje svih potrebnih informacija o kriterijima skeniranja, koje će biti zapisane u konfiguracijskoj datoteci, ali u ovom slučaju su navedene ispod kako bi se čitatelj upoznao s programom.

```
python3 APTuSMO.py
Do you wish to do a network or a domain penetration test? For
network, type 1, for web type 2: 1
Put in your IP address or network range, e.g. (192.168.130.0/24):
192.168.100.56
Would you like an internal or external network scan? Put 1 for
internal, 2 for external: 1
Which interface are you testing, needed for Wireshark (e.g. eth1):
eth0
```

Rezultat 2. Pokretanje glavnog programa, primjer mrežnog testa

```
python3 APTuSMO.py
Do you wish to do a network or a domain penetration test? For
network, type 1, for web type 2: 2
Put in your domain, i.e. (mywebsite.com): https://google-
gruyere.appspot.com/534573053448919143269586844777698226645
```

Rezultat 3. Pokretanje glavnog programa, primjer testa web aplikacije

4.2. Izviđanje i skeniranje

Prvi korak penetracijskog testiranja je izviđanje, nakon kojeg slijedi skeniranje, u ovom poglavlju su opisani točno ti procesi unutar programa, rezultati svih skeniranja spremljeni su u direktorij *scan_results*, ti rezultati se mogu koristiti u iskorištavanju, ali i mogu biti korisni sigurnosnim stručnjacima kod upoznavanja testiranog sustava.

U slučaju da korisnik odabere mrežni test, bit će upitan želi li interni ili vanjski test.

Razlika između ova dva tipa testa je u modulima koji će se pokrenuti; primjerice, Wireshark informacije se neće prikupljati za vanjski sken, dok se Shodan ili Whois neće pokretati za interni sken.

U nastavku ću prvo navesti zajedničke alate internog i vanjskog testa, a kasnije one u kojima se razlikuju.

4.2.1. Nmap

Nmap [9] (Network Mapper) je moćan i fleksibilan alat za mrežno skeniranje i sigurnosne revizije, kojeg je razvio Gordon Lyon, poznat i kao Fyodor. Ovaj alat se široko koristi za otkrivanje uređaja i usluga na računalnim mrežama, kao i za identifikaciju različitih sigurnosnih ranjivosti.

Jedna od ključnih funkcionalnosti Nmapa je otkrivanje poslužitelja (Host Discovery). Nmap može identificirati aktivne uređaje na mreži koristeći razne metode, uključujući ICMP ping, TCP ping i ARP ping. Ova funkcionalnost pomaže u mapiranju mrežne topologije i identifikaciji uređaja koji su spojeni na mrežu.

Druga važna funkcionalnost je skeniranje pristupa (Port Scanning). Nmap može skenirati otvorene pristupe na ciljnim uređajima, što je ključno za razumijevanje koje su mrežne usluge dostupne na određenom uređaju. Nmap podržava različite tehnike skeniranja pristupa, uključujući „TCP SYN“ skeniranje, „TCP Connect“ skeniranje, UDP skeniranje i mnoge druge.

Nadalje, Nmap omogućava detekciju verzija servisa (Service Version Detection). Pored identifikacije otvorenih pristupa, Nmap može detektirati koje su usluge aktivne na tim pristupima, kao i njihove verzije. Ova mogućnost je korisna za identifikaciju potencijalnih ranjivosti specifičnih za određene verzije softvera.

Detekcija operacijskog sustava (OS Detection) je još jedna bitna funkcionalnost. Nmap može identificirati operativni sustav ciljnog uređaja analizom mrežnog prometa, što može biti ključno za planiranje daljnjih koraka penetracijskog testiranja.

Osim toga, Nmap podržava snažan sustav skripti (Nmap Scripting Engine, NSE), koji omogućava izvršavanje prilagođenih skripti tijekom skeniranja. Ove skripte mogu biti korištene za identifikaciju ranjivosti, izvođenje naprednih otkrivanja i prikupljanje dodatnih informacija o cilju. [6]

U ovom projektu, Nmap se koristi za izvođenje nekoliko ključnih koraka penetracijskog testiranja. Prvo se koristi za identifikaciju aktivnih uređaja unutar zadanog mrežnog raspona. Ovo omogućava testeru da identificira sve potencijalne ciljeve za daljnje skeniranje.

Nakon toga, izvršava detaljno skeniranje otkrivenih poslužitelja. To uključuje skeniranje „pristupa“, detekciju verzija servisa i operacijskih sustava, te prikupljanje detaljnih izvještaja o stanju mreže.

Dodatno, izvodi niz različitih skeniranja, uključujući ping skeniranje, brzo skeniranje, detekciju servisa, detekciju operativnog sustava, agresivno skeniranje i mnoge druge. Svako od ovih skeniranja pruža specifične informacije korisne za analizu sigurnosnog stanja mreže.

IoT uređaji često predstavljaju sigurnosne rizike zbog svojih ograničenih sigurnosnih značajki i nedovoljno ažuriranog softvera. Nmap se može koristiti za identifikaciju IoT uređaja na određenoj mreži. Nmap to izvodi tako što pokreće skripte i provjerava pristupe koji su često povezani s IoT uređajima kako bi otkrila prisutnost i identifikaciju tih uređaja. Na primjer, koriste se *banner, http-title, upnp-info, snmp-info, i ftp-anon* skripte, kao i skeniranje uobičajenih „pristupa“ kao što su 21, 22, 23, 80, 161, 443, 5000, i 8000-8100.

Osim toga, dodatne naredbe kao što su *upnp-info, mqtt-subscribe, dns-service-discovery,* i *-sP* se koriste za detaljniju analizu specifičnih protokola i servisa povezanih s IoT uređajima.

Kombinacijom ovih funkcionalnosti, Nmap omogućava detaljnu i sveobuhvatnu analizu mrežnog okruženja. Ovo pomaže penetracijskim testerima da identificiraju sigurnosne ranjivosti te predlože odgovarajuće mjere za njihovo otklanjanje. Na taj način, Nmap ne služi samo kao alat za otkrivanje problema, već i kao ključni resurs u procesu unapređenja mrežne sigurnosti.

U nastavku su prikazani rezultati izvođenja brzog skeniranja pristupa i otkrivanja ranjivosti.

Pokretanje brzog skena:

```
nmap -T4 -F 192.168.100.56

Starting Nmap 7.92 ( https://nmap.org ) at 2024-06-06 11:34 EDT

Nmap scan report for 192.168.100.56
Host is up (0.00032s latency).
Not shown: 97 filtered tcp ports (no-response)
PORT    STATE  SERVICE
22/tcp  closed ssh
80/tcp  open   http
443/tcp open   https
MAC Address: 08:00:27:F7:09:7B (Oracle VirtualBox virtual NIC)
Nmap done: 1 IP address (1 host up) scanned in 2.04 seconds
```

Rezultat 4. Nmap brzi sken

Pokretanje otkrivanja ranjivosti:

```
nmap 192.168.100.56 -sV –script=vulscan/vulscan.nse

Starting Nmap 7.92 ( https://nmap.org ) at 2024-06-06 11:35 EDT
Nmap scan report for 192.168.100.56
Host is up (0.00031s latency).
Not shown: 997 filtered tcp ports (no-response)
PORT    STATE  SERVICE  VERSION
22/tcp  closed ssh
80/tcp  open   http     Apache httpd
| vulscan: VulDB - https://vuldb.com:
| [224717] Apache James up to 3.7.3 JMX Management Service missing
authentication
| [202336] Amazon AWS Apache Log4j Hotpatch Package up to 1.3.5
race condition
| [194540] Northern.tech CFEngine Enterprise up to 3.15.4/3.18.0
Apache/Logs permission
Nmap done: 1 IP address (1 host up) scanned in 26.15 seconds
```

Rezultat 5. Nmap sken ranjivosti

4.2.2. TCP Traceroute

TCP Traceroute [10] je mrežni dijagnostički alat koji koristi TCP (Transmission Control Protocol) pakete za mapiranje puta koji mrežni paketi prelaze od izvornog do odredišnog uređaja. Ovaj alat je posebno koristan u situacijama gdje tradicionalni *traceroute* koji koristi ICMP ili UDP pakete nije učinkovit zbog blokiranja ili filtriranja prometa od strane vatrozida. [4]

TCP Traceroute koristi TCP SYN pakete za slanje zahtjeva na specifične pristupe ciljnog uređaja.

Glavni koraci uključuju postavljanje početne TTL vrijednosti u IP zaglavlju na 1, nakon čega se šalje TCP SYN paket s tom vrijednošću, što rezultira time da TTL vrijednost dođe na 0 kod prvog usmjerivača na putu.

Kada TTL vrijednost dođe na 0, usmjerivač šalje ICMP Time Exceeded poruku natrag izvornom uređaju. Zatim se TTL vrijednost povećava za 1 i šalje se novi paket, omogućujući mu da dosegne drugi usmjerivač prije nego što TTL vrijednost dođe na 0.

Ovaj se proces ponavlja sve dok paket ne stigne do cilja ili dok se ne dosegne maksimalna TTL vrijednost.

Ova metoda omogućava identifikaciju svih usmjerivača na putu do cilja, mjerenje vremena potrebno za prelazak svakog koraka i pružanje detaljne slike mrežnog puta.

Jedna od ključnih prednosti TCP Traceroutea je njegova sposobnost da zaobiđe ograničenja koja nameću vatrozidi, to postiže korištenjem TCP SYN paketa koji su obično dopušteni na uobičajenim „pristupima“ kao što su 80 (HTTP) i 443 (HTTPS). Ova karakteristika omogućava da TCP Traceroute prođe kroz zaštitne slojeve mreže i pruži jasnu sliku mrežnog puta čak i u okruženjima sa strogim sigurnosnim politikama.

```
tcptraceroute completed successfully for 192.168.1.56 on port 22.
traceroute to 192.168.1.56 (192.168.1.56), 30 hops max, 60 byte packets
 1  192.168.1.56 (192.168.1.56) <rst,ack>  0.234 ms  0.181 ms  0.125 ms
```

Rezultat 6. Isječak TCP traceroute skena

4.2.3. WhatWeb

WhatWeb [11] je alat za identifikaciju i analizu web aplikacija, može identificirati web tehnologije, otkriti verzije softvera, prikupljati metapodatke i prepoznati potencijalne sigurnosne ranjivosti na ciljanom web poslužitelju. WhatWeb koristi različite metode otkrivanja, uključujući analizu HTTP zaglavlja, HTML sadržaja, JavaScript-a i drugih dijelova web aplikacije, što ga čini izuzetno korisnim za penetracijsko testiranje i sigurnosne revizije.

Jedna od tehnika koje WhatWeb koristi je analiza HTTP zaglavlja. HTTP zaglavlja koja web poslužitelj vraća često sadrže informacije o softveru i verzijama koje se koriste. Analizom tih zaglavlja, WhatWeb može identificirati osnovne tehnologije koje stoje iza web stranice.

Druga tehnika je analiza HTML sadržaja. HTML kod može otkriti mnogo informacija o korištenim tehnologijama, kao što su specifični tagovi, atributi i komentari koji ukazuju na određene okvire ili CMS platforme. WhatWeb pretražuje HTML kod u potrazi za karakterističnim znakovima specifičnih tehnologija i aplikacija.

JavaScript analiza je također važna komponenta WhatWeba. Analizirajući JavaScript datoteke i varijable, WhatWeb može otkriti informacije o korištenim okvirima i knjižnicama. Imena varijabli, funkcija i objekata često ukazuju na specifične JavaScript knjižnice ili okvire.

Uz ove tehnike, WhatWeb koristi opsežnu bazu podataka s potpisima za prepoznavanje specifičnih verzija softvera i tehnologija. Ova baza podataka sadrži karakteristične uzorke za različite tehnologije, omogućujući WhatWeb-u da precizno identificira širok spektar tehnologija na temelju prepoznatljivih uzoraka.

```
whatweb --color=never -a 1 -v --user-agent 'Mozilla/5.0
(compatible; WhatWeb/0.5.5;
+https://www.morningstarsecurity.com/research/whatweb)'
192.168.100.56

WhatWeb report for http://192.168.100.56
Status    : 200 OK
Title     : <None>
IP        : 192.168.100.56
Country   : RESERVED, ZZ

Summary   : Apache, HTML5, HTTPServer[Apache], Script,
UncommonHeaders[x-mod-pagespeed], X-Frame-Options[SAMEORIGIN]
HTTP Headers:
      HTTP/1.1 200 OK
      Date: Thu, 06 Jun 2024 15:50:09 GMT
      Server: Apache
      X-Frame-Options: SAMEORIGIN
      Accept-Ranges: bytes
      Vary: Accept-Encoding
      X-Mod-Pagespeed: 1.9.32.3-4523
      Content-Encoding: gzip
      Cache-Control: max-age=0, no-cache
```

```
Content-Length: 649
Connection: close
Content-Type: text/html
```

Rezultat 7. Isječak WhatWeb Skena

4.2.4. Dirsearch

Dirsearch [12] je alat za brzo i učinkovito pretraživanje direktorija i datoteka na web poslužiteljima. Razvijen je u Pythonu i koristi *brute-force* tehniku kako bi otkrio skrivene direktorije i datoteke na web poslužiteljima. Dirsearch se koristi za testiranje sigurnosti web aplikacija, pomažući u identifikaciji neovlaštenih pristupnih točaka, administracijskih panela, sigurnosnih kopija, konfiguracijskih datoteka i drugih potencijalno osjetljivih resursa koji nisu namijenjeni za javnu dostupnost.

Dirsearch može koristiti različite ekstenzije datoteka i omogućava korisnicima da specificiraju prilagođene popise direktorija i datoteka koje treba tražiti. Također, može generirati izvještaje u različitim formatima, što olakšava analizu rezultata.

U sklopu penetracijskog testiranja, dirsearch se koristi za otkrivanje skrivenih putanja koje bi mogle biti ranjive na napade. Na primjer, može identificirati neosigurane administracijske panele ili stara sigurnosna kopiranja koja sadrže osjetljive podatke.

Iako je primarno napravljen za web aplikacije, koristan je i za internu i vanjsku upotrebu, pomažući sigurnosnim stručnjacima kako bi identificirali potencijalne sigurnosne propuste i poduzeli odgovarajuće mjere za njihovo otklanjanje.

```
Results for http://192.168.100.56:

http://192.168.100.56/%2e%2e//google.com
http://192.168.100.56/.ht_wsr.txt
http://192.168.100.56/.htaccess.bak1
http://192.168.100.56/.htaccess.orig
http://192.168.100.56/.htaccess.save
http://192.168.100.56/.htaccess_extra
http://192.168.100.56/.htaccess_orig
http://192.168.100.56/.htaccess_sc
http://192.168.100.56/.htaccess.sample
http://192.168.100.56/.htaccessOLD
http://192.168.100.56/.htpasswd_test
http://192.168.100.56/.htm
http://192.168.100.56/.htaccessBAK
```

Rezultat 8. Isječak dirsearch skena

4.2.5. Nikto

Nikto [13] je alat otvorenog koda koji skenira web poslužitelje te otkriva sigurnosne propuste i ranjivosti. Razvijen je s ciljem pružanja sveobuhvatne analize web aplikacija i poslužitelja, pomažući sigurnosnim stručnjacima u identifikaciji i otklanjanju potencijalnih sigurnosnih rizika. Nikto traži razne vrste ranjivosti, uključujući zastarjele verzije softvera, nesigurne konfiguracije, poznate ranjivosti i druge sigurnosne probleme.

Nikto je izuzetno koristan alat za skeniranje web poslužitelja, a njegova primjena je važna i za interna i za vanjska testiranja unutar konteksta sigurnosnih provjera.

Penetracijski testeri koriste Nikto za prepoznavanje skrivenih ranjivosti i nesigurnih konfiguracija koje bi mogle biti iskorištene od strane unutarnjih prijetnji. Na primjer, Nikto može identificirati zastarjele verzije softvera, nesigurne konfiguracije, kao i neautorizirane administracijske sučelja koja nisu pravilno zaštićena. Korištenjem Nikta, stručnjaci mogu skenirati razvojne i testne servere koji možda koriste zastarjele verzije softvera, omogućujući organizacijama da pravovremeno isprave ove sigurnosne propuste i time poboljšaju ukupnu sigurnost svojih internih mreža.

Za vanjska testiranja, Nikto je neprocjenjiv alat za skeniranje javno dostupnih web poslužitelja i aplikacija.

Sigurnosni stručnjaci koriste Nikto kako bi otkrili sigurnosne probleme koji su izloženi internetu i koji bi mogli biti iskorišteni od strane vanjskih napadača. Primjenom Nikta, stručnjaci mogu brzo i efikasno analizirati javne web stranice, administracijske panele i druge resurse dostupne s interneta.

Na primjer, Nikto može otkriti osjetljive datoteke i direktorije koji su nenamjerno izloženi internetu, kao i sigurnosne propuste u vanjskim administracijskim panelima koji omogućavaju neautorizirani pristup. Identifikacija ovih problema pomaže organizacijama da poboljšaju sigurnost svojih javno dostupnih resursa i zaštite se od potencijalnih napada.

Izvođenje Nikto skena za adresu 192.168.100.56:

```
nikto -h 192.168.100.56 -o nikto.txt -maxtime 20s

Target Port: 80
+ GET /: The X-Content-Type-Options header is not set. This could
allow the user agent to render the content of the site in a
different fashion to the MIME type. See:
https://www.netsparker.com/web-vulnerability-
scanner/vulnerabilities/missing-content-type-header/:
+ GET /IB6gzjFW.py: Retrieved x-powered-by header: PHP/5.5.29.
+ GET /index: Uncommon header 'tcn' found, with contents: list.
+ GET /index: Apache mod_negotiation is enabled with MultiViews,
which allows attackers to easily brute force file names. The
following alternatives for 'index' were found: index.html,
index.php. See:
http://www.wisec.it/sectou.php?id=4698ebdc59d15,https://exchange.x
force.ibmcloud.com/vulnerabilities/8275:
+ GET /admin/: This might be interesting.
+ GET /image/: Drupal Link header found with value:
<http://192.168.100.56/?p=23>; rel=shortlink. See:
https://www.drupal.org/:
```

Rezultat 9. Isječak Nikto Skena

4.2.6. Tenable Nessus i Vulnerability Management

Tenable Nessus [14] jedan je od najpoznatijih alata za skeniranje ranjivosti, koristi se za detaljno skeniranje mreža i sustava kako bi se identificirale sigurnosne ranjivosti.

Nessus pruža sveobuhvatne mogućnosti skeniranja, uključujući otkrivanje neispravnih konfiguracija, zastarjelog softvera, otvorenih „pristupa“ i drugih sigurnosnih propusta. [2]

Tenable Vulnerability Management je platforma koja integrira Nessus s dodatnim mogućnostima za upravljanje ranjivostima, izvještavanje, praćenje sigurnosnog statusa i omogućava API pristup prema Nessusu.

U internim mrežnim testovima, cilj je identificirati ranjivosti unutar mreže organizacije. Nessus je izuzetno koristan za detaljno skeniranje svih uređaja unutar interne mreže. Koristeći Nessus, sigurnosni stručnjaci mogu identificirati neispravne konfiguracije koje otkrivaju pogrešno konfigurirane uređaje, čineći ih potencijalno ranjivim na napade. Skeniranjem „pristupa“ i servisa, Nessus provjerava usluge koje rade na mrežnim uređajima i može otkriti potencijalne ranjivosti.

Dodatno, Nessus može detektirati zastarjeli softver, identificirajući uređaje koji koriste zastarjele verzije softvera s poznatim sigurnosnim rupama. Korištenjem Tenable Vulnerability Management platforme, moguće je pratiti i upravljati otkrivenim ranjivostima, omogućavajući kontinuirano praćenje i ažuriranje sigurnosnog statusa mreže. Primjena ovih funkcionalnosti pomaže u osiguravanju da svi uređaji unutar interne mreže budu pravilno konfigurirani i ažurirani, smanjujući rizik od potencijalnih sigurnosnih prijetnji.

U vanjskim mrežnim testovima, cilj je otkriti ranjivosti koje su izložene vanjskim prijetnjama. Nessus se koristi za skeniranje javno dostupnih IP adresa, web aplikacija, mrežnih uređaja i drugih resursa kako bi se identificirale sigurnosne prijetnje. Koristeći Nessus, sigurnosni stručnjaci mogu skenirati web aplikacije te tražiti ranjivosti kao što su SQL injekcije, XSS (Cross-Site Scripting) i drugi sigurnosni propusti.

Također, Nessus se koristi za provjeravanje vanjskih servisa skeniranjem servisa koji su dostupni preko interneta, kao što su SSH ili FTP, kako bi se otkrile potencijalne ranjivosti. Osim toga, skeniranjem mrežnih uređaja, poput usmjerivača i vatrozida, Nessus pomaže u identificiranju sigurnosnih problema koji mogu biti iskorišteni za neovlašteni pristup.

Nessus je vrlo dobar u vanjskim testovima zbog svoje sposobnosti da brzo i efikasno skenira velike mrežne prostore i identificira ranjivosti koje su izložene internetu. Korištenjem Tenable Vulnerability Managementa, stručnjaci mogu kontinuirano pratiti sigurnosni status javno dostupnih resursa i brzo reagirati na nove prijetnje.

```
def netscan_create(name, targets, timeout_min=10):

    url = "https://cloud.tenable.com/scans"

    payload = {
        "settings": {
            "enabled": False,
            "name": name,
            "scan_time_window": timeout_min,
            "text_targets": targets,
            "scanner_id": "xxxx"
        },
        "uuid": "xxxx"
    }

    headers = {
        "accept": "application/json",
        "content-type": "application/json",
        "X-ApiKeys": "accessKey=xxxx"
    }

    response = requests.post(url, json=payload, headers=headers)

    resptext = json.loads(response.text)
    resptext = resptext.get('scan')
    return resptext.get('id')
```

Rezultat 10. . Primjer stvaranja mrežnog Nessus testa pomoću Tenable Vulnerability Managementa

```
HTTP Server Type and Version
     Plugin ID: 10107
     Count: 2
     Severity: 0
     Plugin Family: Web Servers
SSL Certificate Cannot Be Trusted
     Plugin ID: 51192
     Count: 1
     Severity: 2
     Plugin Family: General
```

Rezultat 11. Isječak rezultata Nessus Skena

4.2.7. Moduli vezani uz interne penetracijske testove

U ovom poglavlju detaljno su opisani programi, odnosno moduli unutar programa APTuSMO, koji se koriste primarno u kontekstu internih penetracijskih testova.

4.2.7.1. ARP Scan

„ARP Scan“ koristi se za slanje ARP zahtjeva na ciljanom segmentu mreže kako bi dobio odgovore od svih uređaja. To uključuje prikupljanje IP i MAC adresa uređaja, što omogućuje precizno mapiranje mrežne infrastrukture. U internim testovima, „ARP Scan“ je neophodan jer otkriva uređaje koji su aktivni i povezani na mrežu, ali nisu nužno vidljivi putem tradicionalnih metoda poput ICMP pinga. [6]

Za implementaciju ARP skeniranja koristi se Python biblioteka *Scapy*, koja omogućuje jednostavno stvaranje i slanje ARP zahtjeva te obradu odgovora.

Korisnost ARP skeniranja u internim testovima je višestruka. Prvo, omogućuje identifikaciju skrivenih uređaja koji možda ne odgovaraju na ICMP ping, ali su i dalje aktivni na mreži. Ovo je ključno za otkrivanje svih uređaja unutar mreže, čime se povećava točnost inventara mrežne opreme. Precizno mapiranje mreže pomaže u stvaranju točnog popisa svih povezanih uređaja, što je esencijalno za sigurnosnu procjenu.

Drugo, ARP skeniranje je korisno za otkrivanje neovlaštenih uređaja koji su možda priključeni na mrežu bez dozvole. Identifikacija takvih uređaja je kritična za održavanje sigurnosti mreže, jer neovlašteni uređaji mogu predstavljati značajan sigurnosni rizik.

Treće, prikupljanje IP i MAC adresa putem ARP skeniranja pruža osnovu za daljnje sigurnosne testove. Nakon što su uređaji identificirani, mogu se provesti dodatni testovi kako bi se procijenila njihova sigurnost i otkrili potencijalni ranjivosti.

U konačnici, korištenje ARP skeniranja u internim testovima omogućuje sveobuhvatan pregled mrežne infrastrukture i poboljšava sposobnost identificiranja potencijalnih sigurnosnih rizika unutar organizacije. Ova metoda pruža kritične informacije koje su potrebne za zaštitu mreže i osiguranje integriteta svih povezanih uređaja.

```
Target: 192.168.100.56
IP: 192.168.100.56, MAC: 08:00:27:f7:09:7b
```

Rezultat 12. Primjer ARP Skena

4.2.7.2. Wireshark

Wireshark [15] je jedan od najpoznatijih alata za analizu mrežnog prometa, koji omogućava detaljan pregled i ispitivanje paketa unutar mreže. U okviru internog testiranja sigurnosti, Wireshark može biti izuzetno koristan za presretanje i analizu mrežnih komunikacija.

Wireshark u mom programu koristi *PyShark* i *Scapy* biblioteke za automatsko presretanje i spremanje mrežnog prometa, omogućujući kasniju analizu u Wiresharku.

Prva funkcija modula koristi *PyShark* za presretanje paketa na specificiranom mrežnom sučelju. Korisnik na početku bira mrežno sučelje (npr. eth1) i broj paketa koji će biti presretnuti. *PyShark* kreira *LiveCapture* objekt koji snima mrežni promet u stvarnom vremenu i pohranjuje pakete u listu. Ova funkcija omogućuje korisnicima da jednostavno konfiguriraju i pokrenu proces presretanja, osiguravajući da se svi relevantni paketi uhvate za daljnju analizu.

Nakon što su paketi presretnuti, lista presretnutih paketa se sprema u PCAP datoteku. Ova funkcija koristi *Scapy* za konverziju *PyShark* paketa u *Scapy* pakete, omogućujući fleksibilnije upravljanje i spremanje paketa. Nakon konverzije, paketi se spremaju u datoteku pomoću *Scapy* funkcije *wrpcap*, što omogućuje jednostavnu pohranu i kasniju analizu u Wiresharku. Spremanje paketa u PCAP format standardizira prikupljene podatke i omogućuje njihovo dijeljenje s drugim alatima i stručnjacima za mrežnu sigurnost.

Korištenje Wiresharka u internim testovima omogućuje detaljan uvid u mrežni promet, što je ključno za prepoznavanje sigurnosnih problema i optimizaciju mrežne sigurnosti. Automatiziranje procesa presretanja i spremanja paketa putem *PySharka* i *Scapyja* dodatno olakšava posao sigurnosnim stručnjacima, omogućujući im da se fokusiraju na analizu i interpretaciju podataka umjesto na ručno prikupljanje mrežnih paketa. Ovaj program pruža učinkovit i pouzdan način za prikupljanje mrežnog prometa i analizu sigurnosnih prijetnji unutar organizacije.

```
def capture_packets(interface='eth1', packet_count=1000):
"""
Capture packets on the specified network interface.
Args:
interface (str): Network interface to capture packets from.
packet_count (int): Number of packets to capture.
Returns:
list: List of captured packets.
"""
capture = pyshark.LiveCapture(interface=interface, use_json=True,
include_raw=True) print('Capturing packets...')
capture.sniff(packet_count=packet_count)
return capture
```

Rezultat 13. Capture Packets funkcija unutar Wiresharka

4.2.7.3. Sigurnosno skeniranje krajnjih uređaja

Interni testovi sigurnosti često uključuju temeljitu provjeru sigurnosnog stanja krajnjih uređaja i to je ključna uloga ovih testova. [6]

Modul unutar razvijenog programa koji izvršava sigurnosno skeniranje krajnjih uređaja koristi nekoliko specijaliziranih Python funkcija kako bi obavio različite sigurnosne provjere. Prva provjera fokusira se na ažuriranje softvera, što je ključno za održavanje sigurnosti sustava. Funkcija *check_updates* automatski traži zastarjele softverske pakete ili dostupna ažuriranja, vraćajući rezultate o stanju ažuriranja. Redovito ažuriranje softvera pomaže u ispravljanju sigurnosnih ranjivosti koje napadači mogu iskoristiti.

Provjera dozvola datoteka također je od vitalne važnosti. Funkcija *check_file_permissions* provjerava dozvole na kritičnim datotekama, kao što je npr. */etc/shadow*, koja sadrži šifrirane lozinke korisnika. Pogrešno konfigurirane dozvole mogu omogućiti neovlašten pristup osjetljivim informacijama. Ova funkcija osigurava da su postavke dozvola pravilno postavljene, čime se smanjuje rizik od neovlaštenog pristupa i povećava sigurnost sustava.

Analiza mrežnih veza je još jedan bitan aspekt sigurnosnih provjera. Funkcija *list_network_connections* generira popis trenutnih mrežnih veza na sustavu.

Ova analiza pomaže u otkrivanju sumnjivih ili neovlaštenih aktivnosti koje mogu ukazivati na prisutnost malicioznog softvera ili neovlaštenog korisnika. Identifikacija i analiza mrežnog prometa koji prolazi kroz krajnji uređaj omogućuje bolje razumijevanje sigurnosnog stanja mreže.

Konačno, skeniranje direktorija na prisutnost zlćudnih programa dodatno osigurava krajnje uređaje. Funkcija *scan_directory_with_clamav* koristi ClamAV, antivirusni alat, za skeniranje direktorija kako bi se našao maliciozni softver. Zlćudni programi mogu ozbiljno ugroziti sigurnost sustava, a redovito skeniranje pomaže u otkrivanju i uklanjanju takvih prijetnji, osiguravajući da je direktorij pregledan i da su nađene bilo kakve maliciozne datoteke koje bi mogle ugroziti sigurnost sustava.

Automatiziranje ovih provjera postiže nekoliko ključnih prednosti: efikasnost, dosljednost i centralizirano mjesto za rezultate.

Automatizirano skeniranje štedi vrijeme i smanjuje mogućnost ljudske pogreške u usporedbi s ručnim testiranjem, a provjere se izvode na isti način svaki put, osiguravajući dosljednost u rezultatima.

```
Upgradable packages:
apache2-bin
apache2-dana
File permissions for /etc/shadow: 640
Network connections:
Local Address: 0.0.0.0:57973, Remote Address: N/A, Status: NONE
Local Address: 192.168.100.50:57126, Remote Address:
18.211.24.19:80, Status: TIME_WAIT
Scan results for /home:
/home/kali/.java/fonts/11.0.20-ea/fcinfo-1-kali-Linux-5.16.0-
kali7-amd64-en.properties: OK
/home/kali/.java/fonts/11.0.14.1/fcinfo-1-kali-Linux-5.16.0-kali7-
amd64-en.properties: OK
/home/kali/.java/fonts/17.0.11/fcinfo-1-kali-Kali GNU/Linux-
2024.2-en-US.properties: OK
```

Rezultat 14. Isječak sigurnosnog skeniranja krajnjih uređaja

4.2.8. Moduli vezani uz vanjske penetracijske testove

U ovom poglavlju detaljno su opisani programi, odnosno moduli unutar programa APTuSMO, koji se koriste primarno u kontekstu vanjskih penetracijskih testova.

4.2.8.1. Prikupljanje informacija - whois

Whois [16] modul omogućuje prikupljanje javno dostupnih informacija o registraciji domena, što može pružiti dragocjene uvide u ciljni sustav.

Dodatno, uz modul Whois, koristi se i biblioteka *requests* za dohvaćanje sadržaja web stranice te *BeautifulSoup* za parsiranje HTML-a kako bi dobio naslov stranice. Ovo je korisno za prikupljanje osnovnih informacija o ciljanom web mjestu, što može biti od pomoći prilikom daljnje analize. [4]

```
whois 192.168.100.56

Information for 192.168.100.56:
Whois Information:

#
# ARIN WHOIS data and services are subject to the Terms of Use
# available at: https://www.arin.net/resources/registry/whois/tou/
#
# If you see inaccuracies in the results, please report at
#
https://www.arin.net/resources/registry/whois/inaccuracy_reporting
/
#
# Copyright 1997-2024, American Registry for Internet Numbers,
Ltd.
#


NetRange:       192.168.0.0 - 192.168.255.255
CIDR:           192.168.0.0/16
NetName:        PRIVATE-ADDRESS-CBLK-RFC1918-IANA-RESERVED
NetHandle:      NET-192-168-0-0-1
Parent:         NET192 (NET-192-0-0-0-0)
NetType:        IANA Special Use
OriginAS:
Organization:   Internet Assigned Numbers Authority (IANA)
RegDate:        1994-03-15
Updated:        2024-05-24
Comment:        These addresses are in use by many millions of
independently operated networks, which might be as small as a
single computer connected to a home gateway, and are automatically
configured in hundreds of millions of devices.  They are only
intended for use within a private context  and traffic that needs
```

```
to cross the Internet will need to use a different, unique
address.
Comment:
Comment:        These addresses can be used by anyone without any
need to coordinate with IANA or an Internet registry.  The traffic
from these addresses does not come from ICANN or IANA.  We are not
the source of activity you may see on logs or in e-mail records.
Please refer to http://www.iana.org/abuse/answers
Comment:
Comment:        These addresses were assigned by the IETF, the
organization that develops Internet protocols, in the Best Current
Practice document, RFC 1918 which can be found at:
Comment:        http://datatracker.ietf.org/doc/rfc1918
Ref:            https://rdap.arin.net/registry/ip/192.168.0.0



OrgName:        Internet Assigned Numbers Authority
OrgId:          IANA
Address:        12025 Waterfront Drive
Address:        Suite 300
City:           Los Angeles
StateProv:      CA
PostalCode:     90292
Country:        US
RegDate:
Updated:        2024-05-24
Ref:            https://rdap.arin.net/registry/entity/IANA


OrgAbuseHandle: IANA-IP-ARIN
OrgAbuseName:   ICANN
OrgAbusePhone:  +1-310-301-5820
OrgAbuseEmail:  abuse@iana.org
OrgAbuseRef:    https://rdap.arin.net/registry/entity/IANA-IP-ARIN

OrgTechHandle: IANA-IP-ARIN
OrgTechName:   ICANN
OrgTechPhone:  +1-310-301-5820
OrgTechEmail:  abuse@iana.org
OrgTechRef:    https://rdap.arin.net/registry/entity/IANA-IP-ARIN


#
# ARIN WHOIS data and services are subject to the Terms of Use
# available at: https://www.arin.net/resources/registry/whois/tou/
#
# If you see inaccuracies in the results, please report at
#
https://www.arin.net/resources/registry/whois/inaccuracy_reporting
/
#
# Copyright 1997-2024, American Registry for Internet Numbers,
Ltd.
#=============================================
```

Rezultat 15. Whois sken

4.2.8.2. Shodan

Shodan [17] je specijalizirani pretraživač koji se koristi za pronalaženje uređaja povezanih na internet.

Dok tradicionalni pretraživači poput Googlea indeksiraju web stranice, Shodan pretražuje uređaje kao što su serveri, usmjerivači, IoT uređaji i industrijski kontrolni sustavi.

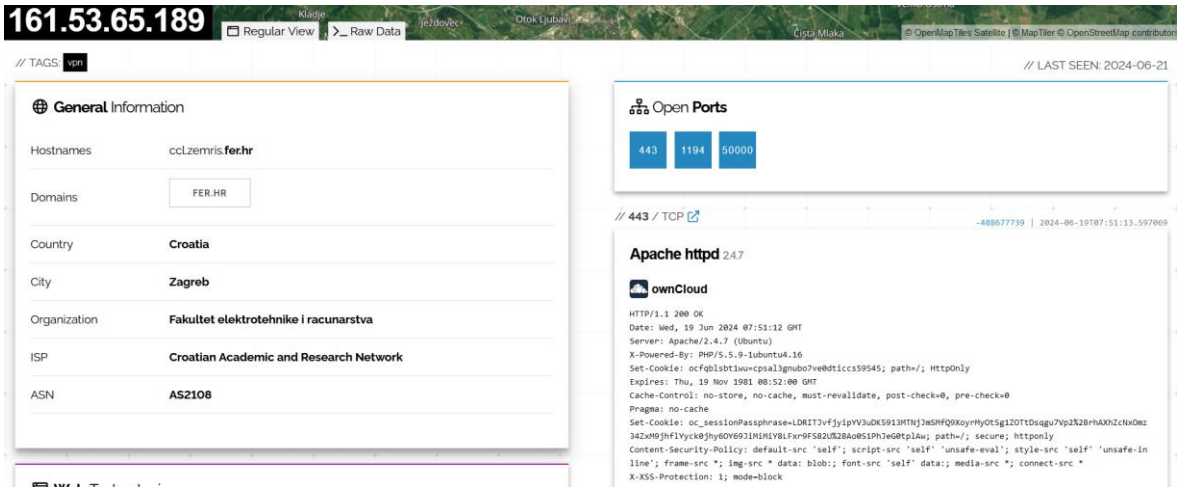
Ovaj alat je iznimno koristan za sigurnosne stručnjake, istraživače i penetracijske testere jer omogućuje otkrivanje potencijalnih ranjivosti u mrežnim sustavima.

Shodan funkcionira tako što kontinuirano skenira internet koristeći različite pristupe i prikuplja podatke o uređajima koji odgovaraju na te zahtjeve. Kada Shodan pronađe uređaj, bilježi informacije poput IP adrese, operativnog sustava i otvorenih „pristupa“. Podaci se pohranjuju u Shodanovu bazu podataka, čineći ih dostupnima za pretraživanje putem web stranice ili API-ja.

Korištenje Shodan API-ja omogućuje programerima integraciju Shodan pretraživanja u različite alate i skripte, moram napomenuti da se Shodan API plaća, ali pretraživanje Shodanove web stranice je besplatno. Na primjer, prilikom penetracijskih testova, sigurnosni stručnjaci mogu koristiti Shodan API za automatizirano prikupljanje informacija o ciljanim uređajima. Skripta može pretraživati Shodanovu bazu podataka za informacije o zadanim IP adresama ili domenama i pohraniti te podatke za kasniju analizu.

Ovo automatiziranje ima nekoliko prednosti. Prvo, štedi vrijeme i smanjuje mogućnost ljudske pogreške u usporedbi s ručnim prikupljanjem podataka. Drugo, omogućuje dosljednost jer se provjere izvode na isti način svaki put, osiguravajući točnost i pouzdanost prikupljenih podataka. Treće, rezultati se pohranjuju na organiziran način, omogućujući jednostavnu analizu, dijeljenje i arhiviranje nalaza.

Korištenje Shodana u vanjskim penetracijskim testovima pruža sveobuhvatan uvid u mrežnu infrastrukturu ciljeva. Informacije dobivene putem Shodana mogu otkriti ranjivosti poput neadekvatno zaštićenih uređaja ili zastarjelog softvera, što omogućuje sigurnosnim stručnjacima da prepoznaju i isprave te ranjivosti prije nego što ih napadači mogu iskoristiti. Na taj način, Shodan je neprocjenjiv alat za poboljšanje sigurnosnog stanja mrežnih sustava i zaštitu od potencijalnih napada.



Slika 1. Primjer Shodan pretrage FERove stranice SPRUT

4.2.9. Testiranje web aplikacija

Druga opcija koju korisnik može odabrati je penetracijsko testiranje web aplikacije, u tom slučaju će biti upitan samo za URL domene koju želi testirati.

Testovi web aplikacija dijele neke module s mrežnim testovima, od prethodno navedenih to su Whois, TCP Traceroute, WhatWeb, Dirsearch ,Nikto i Tenable Vulnerability Management, ali kako je to drugačiji oblik penetracijskog testa, također imaju i puno modula koji se striktno odnose na njih, o njima ću više reći u nastavku.

4.2.9.1. Dig i DNS skenovi

DNS (*Domain Name System*) skenovi su ključna komponenta u sigurnosnim procjenama mreža, omogućujući istraživanje i analizu DNS zapisa povezanih s ciljnim domenama. [1]

Ovi skenovi pružaju uvid u DNS infrastrukturu, pomažući u otkrivanju potencijalnih ranjivosti i prikupljanju bitnih informacija za daljnju analizu. Korištenjem alata poput *dig-a* i drugih DNS alata, moguće je izvesti sveobuhvatne DNS skenove koji otkrivaju kritične podatke o mrežnoj arhitekturi ciljanih sustava.

Ovaj alat omogućuje istraživanje različitih vrsta DNS zapisa, uključujući A, AAAA, MX, NS, TXT zapise i mnoge druge. Kroz DNS upite, *dig* prikuplja informacije o domenama, što uključuje IP adrese povezanih servera, autoritativne DNS poslužitelje, te druge relevantne podatke. Na primjer, upit tipa 'ANY' vraća sve dostupne DNS zapise za određenu domenu, pružajući sveobuhvatan pregled DNS konfiguracije.

Pored *dig* alata, skripta koristi niz drugih DNS naredbi za dublju analizu. Jedna od tih naredbi uključuje pokušaj izvođenja zonalnog transfera (AXFR), koji može otkriti cijelu DNS zonu ako nije pravilno zaštićena. Ovaj postupak može otkriti sve DNS zapise unutar domene, što je kritično za razumijevanje kompletne DNS strukture i potencijalnih ranjivosti.

Dodatne naredbe koriste nmap za skeniranje pristupa povezanih s DNS uslugama, te za identificiranje dodatnih informacija putem specijaliziranih skripti. [1]

Na primjer, Nmap skeniranje može otkriti otvorene pristupe na DNS serverima, što je važno za identificiranje potencijalnih ulaznih točaka za napade. Korištenje nmap-a zajedno s DNS skriptama omogućuje dubinsku analizu i otkrivanje informacija koje nisu vidljive kroz standardne DNS upite.

Izvršavanje DNS skena pomoću nmapa:

```
Running command: nmap -n -v -Pn -sV -p 53 --script dns-nsid
192.168.100.56
Starting Nmap 7.92 ( https://nmap.org ) at 2024-06-06 11:42 EDT
Nmap scan report for 192.168.100.56
Host is up (0.00017s latency).

PORT   STATE    SERVICE VERSION
53/tcp filtered domain
MAC Address: 08:00:27:F7:09:7B (Oracle VirtualBox virtual NIC)

Command e.g. dig 192.168.100.56 MX

; <<>> DiG 9.18.1-1-Debian <<>> NS 192.168.100.56
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 58531
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL:
1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 92c565b2da1e2c17cfc991156661d8fa1fb236d40f601205 (good)
;; AUTHORITY SECTION:
.                10800 IN    SOA   a.root-servers.net.
nstld.verisign-grs.com. 2024060600 1800 900 604800 86400

;; Query time: 8 msec
;; SERVER: 192.168.100.1#53(192.168.100.1) (UDP)
;; WHEN: Thu Jun 06 11:42:49 EDT 2024
;; MSG SIZE  rcvd: 146
```

Rezultat 16. Isječak DNS Skena

4.2.9.2. TheHarvester

TheHarvester [18] je alat za prikupljanje informacija koji se široko koristi u penetracijskim testovima, posebno u fazi inicijalnog prikupljanja podataka.

Ovaj alat omogućuje sigurnosnim stručnjacima da brzo i učinkovito prikupe javno dostupne informacije o cilju iz raznih izvora, uključujući tražilice, društvene mreže i druge javne baze podataka. Osmišljen je da olakša pronalaženje i organizaciju podataka kao što su e-mail adrese, IP adrese, poddomene i drugi relevantni podaci o ciljnim domenama.

TheHarvester radi na principu pretraživanja različitih izvora informacija kako bi prikupio što više podataka o cilju. Među izvorima koje koristi su popularne tražilice poput Googlea, Binga i Baidua, kao i specijalizirane baze podataka kao što su PGP ključni poslužitelji i Shodan. Alat također može pretraživati društvene mreže, uključujući LinkedIn, kako bi prikupio informacije o zaposlenicima i njihovim e-mail adresama. Nakon prikupljanja podataka, TheHarvester ih organizira i prikazuje u preglednom formatu, omogućujući sigurnosnim stručnjacima da ih lako analiziraju i koriste u daljnjim fazama testiranja.

Prikupljeni podaci pružaju dragocjene uvide u strukturu i sigurnosno stanje ciljne domene. Na primjer, identificiranje e-mail adresa može pomoći u provođenju socijalnog inženjeringa ili phishing napada, dok otkrivanje poddomena može otkriti dodatne ulazne točke koje bi mogle biti ranjive.

Prikupljanje IP adresa omogućuje mapiranje mrežne infrastrukture, dok informacije o poslužiteljima i uslugama koje se koriste na ciljnim sustavima pomažu u identifikaciji potencijalnih ranjivosti specifičnih za određene tehnologije i softver.

Korištenje TheHarvester-a donosi nekoliko ključnih prednosti. Prvo, omogućuje brz i učinkovit način prikupljanja širokog spektra informacija iz različitih izvora, čime se smanjuje vrijeme potrebno za ručno pretraživanje. Drugo, organizira prikupljene podatke u preglednom formatu, olakšavajući analizu i korištenje tih podataka u daljnjim testovima. Konačno, omogućuje sigurnosnim stručnjacima da dobiju sveobuhvatan uvid u ciljne sustave, što je ključno za učinkovito i uspješno provođenje penetracijskih testova.

```
theHarvester -d 192.168.100.56 -l 100 -b all

[*] Searching Brave.

[*] ASNS found: 1
--------------------
AS15169

[*] Interesting Urls found: 15
--------------------
http://google-gruyere.appspot.com/
http://google-
gruyere.appspot.com/341973498225144587203316926302118026940/login
http://google-
gruyere.appspot.com/558048196702655120925705138677109065729/
https://google-gruyere.appspot.com/

[*] IPs found: 81
-------------------
142.250.181.244
142.250.185.148
```

Rezultat 17. Isječak TheHarvester Skena

4.2.9.3. Sublist3r

Sublist3r [19] je popularan alat razvijen za nalaženje poddomena web stranica koristeći OSINT (Open Source Intelligence).

Penetracijski testeri koriste Sublist3r kako bi prikupili informacije o poddomenama ciljne domene. Alat koristi različite tražilice i specijalizirane servise za prikupljanje podataka, kao što su:

- **Tražilice**:
  - Google
  - Yahoo
  - Bing
  - Baidu
- **Specijalizirani servisi**:
  - Netcraft
  - Virustotal
  - ThreatCrowd
  - DNSdumpster

Sublist3r integrira rezultate iz ovih izvora kako bi pružio što kompletniji popis poddomena za zadanu domenu. To pomaže stručnjacima da identificiraju potencijalne ulazne točke i ranjivosti koje mogu postojati na manje poznatim poddomenama.

```
www.kali.org
10cake.kali.org
10year.kali.org
aeacus.kali.org
aphrodite.kali.org
archive.kali.org
archive-10.kali.org
archive-11.kali.org
```

Rezultat 18. Isječak sublist3r skena za domenu kali.org

4.2.9.4. WafW00f

WafW00f [20] je alat razvijen za detekciju Web Aplikacijskih Vatrozida (WAF - Web Application Firewall).

Ovaj alat pruža penetracijskim testerima mogućnost da identificiraju i analiziraju prisutnost WAF-ova na ciljnim web stranicama. WAF-ovi igraju ključnu ulogu u zaštiti web aplikacija filtriranjem i praćenjem HTTP prometa između web aplikacije i interneta, čime se sprječavaju razni napadi i neovlašteni pristupi.

Jedna od glavnih karakteristika WafW00fa je njegova sposobnost detekcije širokog spektra WAF-ova. Alat koristi razne tehnike otkrivanja kako bi prepoznao različite vrste i verzije WAF-ova. Ovo je posebno korisno za sigurnosne stručnjake jer im omogućuje da razumiju koji su sigurnosni sustavi postavljeni na ciljnoj web stranici, te da planiraju daljnje korake u svojoj analizi.

WafW00f također omogućuje korisnicima da automatiziraju proces skeniranja i identificiranja WAF-ova na velikom broju web stranica. Ova funkcionalnost je izuzetno vrijedna za penetracijske testere jer im omogućuje da brzo i efikasno skeniraju više ciljeva bez potrebe za ručnim radom, a na kraju pruža rezultate na vrlo čitljiv i razumljiv način.

```
wafw00f -a 192.168.100.56

The Web Application Firewall Fingerprinting Toolkit
[*] Checking https://google-
gruyere.appspot.com/534573053449819143269586484777699826645
[+] Generic Detection results:
[-] No WAF detected by the generic detection
[~] Number of requests: 7
```

Rezultat 19. Isječak WAF Skena

4.2.9.5. WPScan

WPScan [21] je alat namijenjen za sigurnosno skeniranje WordPress web stranica. Razvijen prvenstveno za penetracijske testere i sigurnosne istraživače, WPScan pomaže u identificiranju ranjivosti i sigurnosnih problema specifičnih za WordPress platformu. S obzirom na popularnost WordPress-a kao sustava za upravljanje sadržajem (CMS), WPScan je neophodan alat za održavanje sigurnosti web stranica koje koriste ovu platformu.

WPScan koristi razne metode za otkrivanje ranjivosti, uključujući skeniranje verzije WordPressa, tema i dodataka (plugins), kao i provjeru prisutnosti slabih lozinki i drugih konfiguracijskih problema. Alat također omogućuje enumeraciju korisnika, što može pomoći u otkrivanju potencijalno ugroženih korisničkih računa.

Jedna od ključnih prednosti WPScan-a je njegova mogućnost integracije s API-jem za dodatne informacije o ranjivostima. Korištenjem API tokena, WPScan može pristupiti bazi podataka poznatih ranjivosti i pružiti detaljne informacije o pronađenim sigurnosnim propustima. Ova funkcionalnost značajno povećava učinkovitost skeniranja i točnost rezultata.

```
wpscan --url 192.168.1.56 --enumerate u --format json –output
wpscan_output.json
```


```
{
    "banner": {
        "description": "WordPress Security Scanner by the WPScan
Team",
        "version": "3.8.25",
        "authors": [
            "@_WPScan_",
```

```
            "@ethicalhack3r",
            "@erwan_lr",
            "@firefart"
        ],
        "sponsor": "Sponsored by Automattic -
https://automattic.com/"
    },
    "scan_aborted": "The remote website is up, but does not seem
to be running WordPress.",
    "target_url": "https://google-
gruyere.appspot.com/534573053448919143269586844777698226645/"
}
```


Rezultat 20. Isječak WPScan rezultata, stranica nije rađena u WordPressu

4.2.9.6. Burp Suite

Burp Suite [22] je sveobuhvatan alat za sigurnosno testiranje web aplikacija koji se široko koristi među penetracijskim testerima. Razvijen od strane PortSwiggera, Burp Suite nudi niz funkcionalnosti koje omogućuju identifikaciju, analizu i iskorištavanje sigurnosnih ranjivosti u web aplikacijama. Njegova modularnost i fleksibilnost čine ga ključnim alatom za bilo koji projekt koji se bavi sigurnosnim provjerama web aplikacija. [3]

Burp Suite se sastoji od nekoliko modula, svaki dizajniran za specifične zadatke unutar procesa sigurnosnog testiranja.

Neki od ključnih modula uključuju Burp Proxy, koji omogućuje presretanje i manipulaciju HTTP/HTTPS prometa između preglednika i ciljne aplikacije. Ovaj modul je izuzetno koristan za analizu i testiranje sigurnosti web aplikacija jer omogućuje pregled i izmjenu prometa u stvarnom vremenu.

Burp Scanner je automatizirani alat za skeniranje ranjivosti u web aplikacijama. Koristi se za identificiranje sigurnosnih slabosti u aplikacijama, poput SQL injekcija, XSS napada i drugih uobičajenih ranjivosti, te pruža detaljne izvještaje o pronađenim problemima.

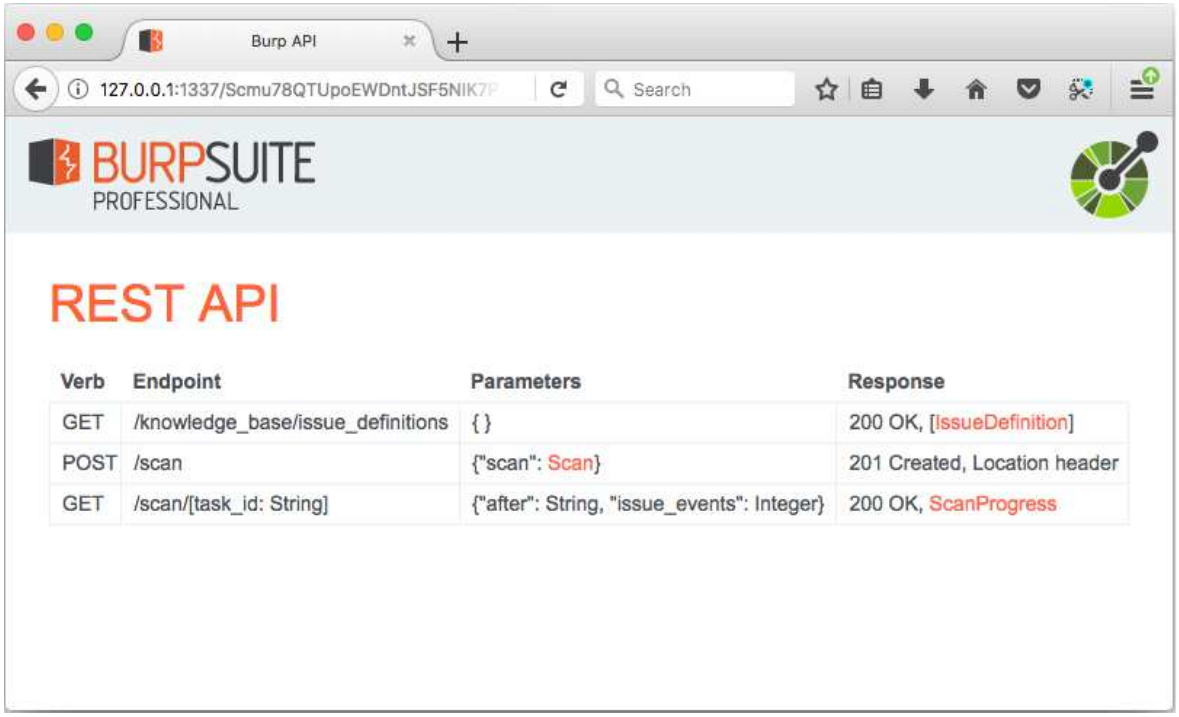
Burp Intruder omogućuje automatizirano ponavljanje HTTP zahtjeva s ciljem iskorištavanja ranjivosti. Ovaj alat je vrlo fleksibilan i može se koristiti za razne vrste napada, uključujući *brute force* napade, *fuzzing*, i testiranje autentifikacije.

Burp Repeater je alat za ponavljanje i izmjenu HTTP zahtjeva. Omogućuje korisnicima da izmijene HTTP zahtjeve te pregledaju odgovore, što je korisno za testiranje specifičnih parametara i provjeru reakcija aplikacije na različite unose.

Burp Sequencer analizira nasumičnost tokena generiranih od strane aplikacije. Ovaj alat pomaže u procjeni sigurnosti tokena koji se koriste za sesije, autentifikaciju, CSRF zaštitu i druge sigurnosne mehanizme, osiguravajući da su dovoljno nasumični i teško predvidljivi.

Burp REST API je alat koji omogućuje programsku kontrolu nad Burp Suiteom, omogućavajući korisnicima da integriraju Burp Suite funkcionalnosti unutar vlastitih skripti i alata. Korištenjem REST API-ja, korisnici mogu pokretati i upravljati Burp Suite zadacima izravno iz Python skripti ili drugih programskih jezika koji podržavaju HTTP zahtjeve.

Ova mogućnost značajno povećava fleksibilnost i učinkovitost automatiziranja sigurnosnih testova.



Slika 2. Burp Suite REST API, izvor: Portswigger

- **Automatiziranje**
    - Burp REST API omogućuje potpuno automatiziranje skeniranja web aplikacija, što je ključno za kontinuiranu integraciju i kontinuirano isporučivanje (CI/CD) procesa.
    - Redovito skeniranje može biti zakazano i automatizirano bez potrebe za ručnom intervencijom, što osigurava dosljednu sigurnosnu provjeru kroz cijeli razvojni ciklus.
- **Integracija**
    - API omogućuje jednostavnu integraciju Burp Suitea s drugim alatima i skriptama, npr. korisnici mogu koristiti Python skripte za pokretanje skeniranja, analizu rezultata i generiranje izvještaja, čime se Burp Suite funkcionalnosti mogu lako uklopiti u postojeće sigurnosne alate i sustave.

- **Udaljeno Upravljanje**
    - Iako API omogućuje programsku kontrolu unutar istog sustava, on također podržava scenarije u kojima je potrebno upravljati Burp Suiteom s udaljenog servera ili unutar distribuiranih sustava.
- **Prilagodljivost**
    - Korisnici mogu pisati skripte koje koriste API za vlastite potrebe, to jest, moguće je napisati skripte koje pokreću specifične vrste skeniranja, prilagođavaju postavke skeniranja, prate napredak i automatski preuzimaju i analiziraju rezultate.

```
INFO - Initiating unauthenticated scan...
INFO - https://google-
gruyere.appspot.com/534573053449819143269586484777699826645 Added
to the scan queue, ID 3
INFO - Scan started
INFO - Scan status: crawling
INFO - Scan status: auditing
INFO - Timeout reached. Stopping scan(s).
INFO - Scan completed
INFO - Scan metrics for https://google-
gruyere.appspot.com/534573053449819143269586484777699826645 :
INFO - CURRENT_URL =
INFO - CRAWL_REQUESTS_MADE = 54
INFO - CRAWL_NETWORK_ERRORS = 0
```

```
INFO - CRAWL_UNIQUE_LOCATIONS_VISITED = 8
INFO - CRAWL_REQUESTS_QUEUED = 0
INFO - AUDIT_QUEUE_ITEMS_COMPLETED = 0
INFO - AUDIT_QUEUE_ITEMS_WAITING = 12
INFO - AUDIT_REQUESTS_MADE = 1432
INFO - AUDIT_NETWORK_ERRORS = 0
INFO - ISSUE_EVENTS = 23
INFO - CRAWL_AND_AUDIT_CAPTION = Auditing. 2h 12m estimated time
remaining.
INFO - CRAWL_AND_AUDIT_PROGRESS = 12
INFO - TOTAL_ELAPSED_TIME = 56
INFO - Scan issues for https://google-
gruyere.appspot.com/534573053449819143269586484777699826645 :
INFO - Issue: Strict transport security not enforced, Severity:
Low (Certain)
INFO - Issue: Password field with autocomplete enabled, Severity:
Low (Certain)
INFO - Issue: Browser cross-site scripting filter disabled,
Severity: Information (Certain)
[REDACTED]
INFO - Issue: Password submitted using GET method, Severity: Low
(Certain)
INFO - Downloading HTML/XML report for https://google-
gruyere.appspot.com/534573053449819143269586484777699826645
INFO - Shutting down Burp Suite ...
INFO - Burp is stopped

ScanRecord(task_id='3', target_url='https://google-
gruyere.appspot.com/534573053449819143269586484777699826645',
date_time='2024-06-22T00:01:06', status='auditing',
metrics={'current_url': '', 'crawl_requests_made': 54,
'crawl_network_errors': 0, 'crawl_unique_locations_visited': 8,
'crawl_requests_queued': 0, 'audit_queue_items_completed': 0,
'audit_queue_items_waiting': 12, 'audit_requests_made': 1432,
'audit_network_errors': 0, 'issue_events': 23,
'crawl_and_audit_caption': 'Auditing. 2h 12m estimated time
remaining.', 'crawl_and_audit_progress': 12, 'total_elapsed_time':
56}, report_files=['/tmp/burp-report_20240622-0002_httpsgoogle-
gruyere.appspot.com534573053449819143269586484777699826645.htm'])
```

Rezultat 21. Burp Suite automatizirani sken

Mrežni testovi i testovi web aplikacija pružaju sveobuhvatan pristup procjeni sigurnosnog stanja sustava, posebno u složenim mrežnim okruženjima. Kroz poglavlja su detaljno obrađeni različiti alati i moduli poput Nmapa, Dirsearcha, Nikta, Tenable Nessusa, Wiresharka, Whoisa, Shodana i Burp Suitea. Ovi alati predstavljaju raznovrstan set metoda i tehnika za procjenu sigurnosti mrežnih sustava i web aplikacija.

Kombinacija internog i vanjskog testiranja omogućava ciljani pristup specifičnim sigurnosnim potrebama i prijetnjama u složenim mrežnim okruženjima. Ovaj sveobuhvatni pristup omogućuje prepoznavanje potencijalnih prijetnji i implementaciju učinkovitih mjera za njihovo otklanjanje.

Korištenje naprednih alata i automatiziranih skripti olakšava proces prikupljanja i analize podataka, smanjujući mogućnost ljudske pogreške te osiguravajući dosljednost i točnost rezultata. Integracija različitih modula u mrežno i web testiranje pruža dubinski uvid u mrežnu infrastrukturu i funkcionalnost web aplikacija, što je ključno za učinkovitu sigurnosnu procjenu i unapređenje sigurnosnog stanja složenih mrežnih sustava.

U složenim mrežnim okruženjima, redovito provođenje mrežnih testova i testova web aplikacija ključni je dio sigurnosne strategije svake organizacije. Ovi testovi pružaju nužne informacije za zaštitu podataka i resursa i omogućuju organizacijama da budu korak ispred potencijalnih prijetnji.

Kontinuirano poboljšanje sigurnosnog statusa i otpornosti na napade osigurava dugoročnu zaštitu i integritet sustava. Kombinacijom različitih modula i testova, sigurnosni stručnjaci mogu pravovremeno identificirati i otkloniti ranjivosti, osiguravajući visoku razinu zaštite i sigurnosti u složenim mrežnim okruženjima.

4.3. Iskorištavanje

Iskorištavanje je ključna faza penetracijskog testiranja koja slijedi nakon izviđanja i skeniranja. U ovoj fazi, fokus se prebacuje s identifikacije ranjivosti na njihovo aktivno iskorištavanje kako bi se demonstrirala stvarna mogućnost napada. Cilj iskorištavanja je prodrijeti u sustav ili aplikaciju, dobiti neovlašteni pristup, preuzeti kontrolu nad ciljnim resursima, ili eksfiltrirati osjetljive podatke.

U složenim mrežnim okruženjima, faza iskorištavanja je posebno izazovna zbog višeslojnih sigurnosnih mehanizama i naprednih zaštitnih mjera koje su implementirane kako bi zaštitile mrežu i njene komponente. Zbog toga se koriste napredni alati i tehnike kako bi se što učinkovitije iskoristile otkrivene ranjivosti.

Svi rezultati vezani uz iskorištavanje spremljeni su u direktorij *attack_reports*.

4.3.1. Metasploit Framework

Metasploit [23] je jedan od najpoznatijih i najmoćnijih alata za penetracijsko testiranje, koji omogućuje sigurnosnim stručnjacima da identificiraju, testiraju i iskorištavaju ranjivosti u sustavima.

Razvio ga je H. D. Moore 2003. godine, a kasnije ga je preuzela kompanija Rapid7. Metasploit je postao standardni alat svakog penetracijskog testera zahvaljujući svojoj fleksibilnosti, velikom broju dostupnih modula za iskorištavanje ranjivosti (engl. exploit).

Metasploit omogućuje korisnicima da brzo i jednostavno koriste gotove module za iskorištavanje ranjivosti, ali i da kreiraju vlastite. Osim modula, Metasploit također nudi širok spektar *payload* unosa, *post-exploit* modula i mnogih drugih alata koji olakšavaju penetracijsko testiranje.

Korištenje Metasploita uključuje nekoliko ključnih koraka:

- **Pokretanje Metasploita**
    - Metasploit se može pokrenuti putem konzole (msfconsole) ili web sučelja. Najčešće se koristi msfconsole zbog fleksibilnosti.
- **Odabir modula za iskorištavanje ranjivosti**
    - Korisnik može pretraživati bazu modula koristeći naredbu *search*, a zatim odabrati željeni pomoću naredbe *use*.

- **Konfiguracija modula**
    - Nakon odabira modula, korisnik mora postaviti parametre kao što su ciljna IP adresa (RHOST), lokalna IP adresa (LHOST), ciljni pristup (RPORT) i drugi specifični parametri za odabrani modul.
- **Odabir *payload* unosa**
    - *Payload* predstavlja kod koji će biti izvršen na ciljnom sustavu nakon uspješnog iskorištavanja.
    - Korisnik može birati između različitih unosa ovisno o cilju napada i željenom rezultatu.
- **Pokretanje modula**
    - Nakon konfiguriranja modula i odabira unosa, korisnik pokreće modul koristeći naredbu *run* ili *exploit*.
    - Metasploit tada pokušava iskoristiti ranjivost i izvršiti unos na ciljnom sustavu.
- ***Post-exploiting***
    - Ako je modul uspješan, Metasploit omogućuje daljnje radnje poput prikupljanja informacija, instalacije trajnih stražnjih vrata (engl. backdoor), premještanja u druge sustave unutar mreže i slično.

4.3.1.1. Korištenje metasploita u programu

U ovom projektu, Metasploit je korišten za iskorištavanje različitih ranjivosti identificiranih u fazama izviđanja i skeniranja. Alat je korišten kako za mrežne testove (interni i vanjski), tako i za testiranje web aplikacija.

Pronalaženje ranjivosti temeljilo se na rezultatima skeniranja pomoću alata kao što su Nmap i Nessus za mrežne testove, te Burp Suite, Nikto, WafW00f, WhatWeb i WPScan za testove web aplikacija. Ovi alati omogućili su identifikaciju specifičnih ranjivosti i otvorenih pristupa na ciljanim sustavima, čime su postavljeni temelji za daljnju analizu i iskorištavanje.

Odabir odgovarajućih modula za iskorištavanje ranjivosti proveden je korištenjem Metasploit okvira. Pretraženi su moduli koji su relevantni za ranjivosti i servise identificirane tijekom početne faze skeniranja. Ovaj korak osigurava da su korišteni alati i metode optimalno prilagođeni specifičnim sigurnosnim propustima ciljanih sustava.

Iskorištavanje je uključivalo konfiguriranje parametara za odabrane module, koji su potom pokrenuti s ciljem dobivanja neovlaštenog pristupa ciljanim sustavima. Ovaj korak predstavlja praktičnu primjenu prethodno odabranih tehnika i alata kako bi se ostvario stvarni pristup ili kontrola nad ranjivim sustavima, testirajući time njihovu otpornost i sigurnost u stvarnim uvjetima.

4.3.1.2. Primjeri iskorištavanja

Nmap skeniranje otkrilo je otvorene pristupe i ranjive servise, što je omogućilo pretragu odgovarajućih modula u Metasploit bazi podataka. Identifikacija tih otvorenih pristupa i ranjivih servisa bila je ključna za određivanje potencijalnih vektora napada i odabir specifičnih modula koji su kasnije korišteni za pokušaj proboja sustava. Korištenjem Metasploita, možemo brzo pronaći relevantne module i konfigurirati ih za ciljanje otkrivenih ranjivosti.

Nessus izvještaji pružili su detaljne informacije o specifičnim ranjivostima, uključujući CVE oznake, koje su korištene za nalaženje modula za iskorištavanje ranjivosti i njihovu primjenu na ciljne sustave.

PCAP analize omogućile su prepoznavanje specifičnih komunikacija unutar mreže, koje su potom korištene za ciljanje određenih „pristupa“ i servisa.

Analizom mrežnog prometa dobili smo uvid u aktivnosti koje se odvijaju unutar mreže, što je omogućilo preciznije ciljanje modula na temelju stvarnih podataka o mrežnom prometu i ponašanju sustava.

Burp Suite skeniranja otkrila su web ranjivosti koje su iskorištene korištenjem odgovarajućih Metasploit modula. Identifikacija ovih ranjivosti bila je ključna za razumijevanje sigurnosnih propusta u web aplikacijama, što je omogućilo ciljanje specifičnih web ranjivosti, demonstrirajući stvarne mogućnosti napada na web aplikacije.

```
2024-06-21 21:18:28,171 - INFO - Parsed results:
2024-06-21 21:18:28,171 - INFO - Nmap open ports: [('123', 'ntp'),
('10000', 'snet-sensor-mgmt'…]

2024-06-21 21:18:28,171 - INFO - Nmap vulnerabilities: ['CVE-2003-
1172', 'CVE-2002-2009', 'CVE-2012-0393', 'CVE-2008-3666'...]
```

```
2024-06-21 21:18:28,171 - INFO - Nessus results: [{'count': 2,
'plugin_id': 10107, 'plugin_name': 'HTTP Server Type and Version',
'severity': 0, 'plugin_family': 'Web Servers', 'vuln_index': 1},
{'count': 2, 'plugin_id': 22964, 'plugin_name': 'Service
Detection', 'severity': 0, 'plugin_family': 'Service detection',
'vuln_index': 2}...]

2024-06-21 21:18:28,173 - INFO - PCAP results: [['192.168.100.45',
'192.168.100.255', '', '', '137', '137']…]

2024-06-21 21:18:28,173 - INFO - Nikto results: ['License file
found may identify site software', 'This might be interesting',
'Drupal Link header found with value'…]

2024-06-21 21:20:15,714 - INFO - Found exploits for term '137':
['exploit/linux/local/abrt_sosreport_priv_esc',
'exploit/multi/http/solr_velocity_rce'…]
2024-06-21 21:20:15,714 - INFO - Searching for exploits related
to: 137
2024-06-21 21:20:35,623 - INFO - Search output for term '137':

Matching Modules
================

   #   Name
Disclosure Date  Rank       Check  Description
   -   ----
---------------  ----       -----  -----------
   0   exploit/linux/local/abrt_sosreport_priv_esc
2015-11-23       excellent  Yes    ABRT sosreport Privilege
Escalation
   1
exploit/windows/scada/advantech_webaccess_dashboard_file_upload
2016-02-05       excellent  Yes    Advantech WebAccess Dashboard
Viewer uploadImageCommon Arbitrary File Upload
   2   exploit/multi/http/solr_velocity_rce
2019-10-29       excellent  Yes    Apache Solr Remote Code
Execution via Velocity Template


2024-06-21 21:24:46,966 - INFO - Found exploits for term 'A
Wordpress installation was found': []
2024-06-21 21:24:46,966 - INFO - Found exploits:
['exploit/multi/http/jenkins_metaprogramming',
'exploit/linux/local/cve_2021_3490_ebpf_alu32_bounds_check_lpe',
'exploit/android/browser/stagefright_mp4_tx3g_64bit'…]

2024-06-21 21:24:46,966 - INFO - Executing exploit:
exploit/multi/http/jenkins_metaprogramming against target:
192.168.100.56
2024-06-21 21:25:40,810 - INFO - Result of exploit
exploit/multi/http/jenkins_metaprogramming:
[*] Using configured payload java/meterpreter/reverse_https
RHOSTS => 192.168.100.56
LHOST => 127.0.0.1
```

```
[!] You are binding to a loopback address by setting LHOST to
127.0.0.1. Did you want ReverseListenerBindAddress?
[*] Started HTTPS reverse handler on https://127.0.0.1:8443
[*] Running automatic check ("set AutoCheck false" to disable)
[-] Exploit aborted due to failure: unknown: Cannot reliably check
exploitability. "set ForceExploit true" to override check result.
[*] Exploit completed, but no session was created.


2024-06-21 21:26:27,984 - INFO - Executing exploit:
exploit/unix/webapp/wp_pixabay_images_upload against target:
192.168.100.56
2024-06-21 21:26:56,173 - INFO - Result of exploit
exploit/unix/webapp/wp_pixabay_images_upload:
[*] No payload configured, defaulting to
php/meterpreter/reverse_tcp
RHOSTS => 192.168.100.56
LHOST => 127.0.0.1
[!] You are binding to a loopback address by setting LHOST to
127.0.0.1. Did you want ReverseListenerBindAddress?
[*] Started reverse TCP handler on 127.0.0.1:4444
[-] Exploit aborted due to failure: no-target: 192.168.100.56:80 -
/ does not seeem to be WordPress site
[*] Exploit completed, but no session was created.
```

Rezultat 22. Isječak Metasploit iskorištavanja

Svi moduli koje Metasploit pronađe preusmjereni su u *ms_exploits* datoteku, a rezultati iznad zapisani su u datoteku *exploits_log*.

4.3.2. Hydra

Hydra [24] je popularni alat za probijanje lozinki koji se koristi za testiranje sigurnosti sustava. Omogućuje napade *brute-force* i *dictionary* na različite mrežne servise kako bi se otkrile slabe lozinke. Hydra podržava veliki broj protokola poput SSH, FTP, HTTP, i mnoge druge, što ga čini iznimno fleksibilnim alatom za penetracijsko testiranje.

Program prihvaća listu korisničkih imena i lozinki te provjerava svaku kombinaciju protiv specificiranih servisa i poslužitelja koristeći Hydra alat.

Modul priprema listu korisničkih imena i lozinki, koja se može dobiti iz datoteka ili izravno iz proslijeđenih unosa. Potom, za svaki servis, poslužitelj, korisničko ime i lozinku, program koristi Hydra alat za pokušaj autentifikacije.

```
Hydra encountered an error: Command '['hydra', '-l', 'Elliot', '-
p', 'ER28-0652', '-t 4', 'ssh://192.168.100.56']' returned non-
zero exit status 255.
Command: hydra -l robot -p abcdefghijklmnopqrstuvwxyz -t 4
ssh://192.168.100.56
[WARNING] Restorefile (you have 10 seconds to abort... (use option
-I to skip waiting)) from a previous session found, to prevent
overwriting, ./hydra.restore
[ERROR] could not connect to ssh://192.168.100.56:22 - Connection
refused

Trying Elliot:ER28-0652 on http-form-post://192.168.0.56
[DATA] attacking http-post-form://192.168.0.56:80/wp-
login.php:log=^USER^&pwd=^PASS^:F=Invalid username
[80][http-post-form] host: 192.168.0.113   login: Elliot
password: ER28-0652
1 of 1 target successfully completed, 1 valid password found
```

Rezultat 23. Isječak Hydra napada

4.3.3. SQLMap

SQLMap [25] je alat otvorenog koda koji se koristi za otkrivanje i iskorištavanje SQL injekcija u web aplikacijama. Razvijen je od strane Miroslava Štampara i drugih suradnika. Alat automatizira proces otkrivanja i eksploatacije SQL injekcija, omogućujući penetracijskim testerima da brzo i učinkovito procijene sigurnost web aplikacija. [2]

SQLMap funkcionira tako što analizira web aplikacije i pronalazi potencijalne SQL injekcije. Korisnik unosi ciljanu URL adresu koja može biti ranjiva, a SQLMap testira tu adresu pomoću različitih tehnika kako bi identificirao ranjive parametre. Nakon što potvrdi postojanje ranjivosti, SQLMap automatizira proces iskorištavanja, omogućujući korisniku da dohvati podatke iz baze podataka, manipulira datotekama na poslužitelju i izvršava naredbe na operativnom sustavu.

U modulu vezan uz moj program, SQLMap se koristi za testiranje niza URL adresa na ranjivosti SQL injekcija. Prvo, skripta prima URL adrese iz liste ili datoteke. Nakon toga, za svaku adresu, SQLMap se pokreće u *batch* načinu rada, što znači da se izvršava bez potrebe za interakcijom s korisnikom. Rezultati se pohranjuju u datoteku, omogućujući pregled svih otkrivenih ranjivosti.

```
[21:12:07] [INFO] testing connection to the target URL
got a 301 redirect to
'http://192.168.100.56/sugarcrm/?module=Accounts&action=ShowDuplic
ates'. Do you want to follow? [Y/n] Y
[21:12:07] [INFO] testing if the target URL content is stable
[21:12:07] [WARNING] GET parameter 'module' does not appear to be
dynamic
[21:12:08] [INFO] testing 'AND boolean-based blind - WHERE or
HAVING clause'
[21:12:09] [INFO] testing 'Boolean-based blind - Parameter replace
(original value)'
[REDACTED]
[21:12:19] [WARNING] GET parameter 'action' does not seem to be
injectable
[21:12:19] [CRITICAL] all tested parameters do not appear to be
injectable. Try to increase values for '--level'/'--risk' options
if you wish to perform more tests. If you suspect that there is
some kind of protection mechanism involved (e.g. WAF) maybe you
could try to use option '--tamper' (e.g. '--tamper=space2comment')
and/or switch '--random-agent'
```

Rezultat 24. Isječak SQLMap napada

4.3.4. XSSStrike

XSSStrike [26] je alat otvorenog koda dizajniran za otkrivanje i iskorištavanje XSS (Cross-Site Scripting) ranjivosti u web aplikacijama. XSSStrike koristi napredne tehnike kako bi automatizirao proces pronalaženja XSS ranjivosti, čineći ga izuzetno korisnim za penetracijske testere.

XSS ranjivosti omogućuju napadaču umetanje zlonamjernog koda u web stranicu, koji se potom izvršava u pregledniku drugih korisnika. Ovo može dovesti do krađe kolačića, otmice sesija, lažiranja sadržaja i drugih zlonamjernih aktivnosti. [4]

XSSStrike koristi kombinaciju heurističkih i automatiziranih metoda za identificiranje i iskorištavanje XSS ranjivosti. Alat analizira HTML i JavaScript kod na ciljanim web stranicama kako bi otkrio potencijalno ranjive točke. Koristeći različite *payload* unose i tehnike injekcije, XSSStrike testira ove točke kako bi provjerio jesu li ranjive na XSS napade.

Neke od ključnih značajki XSSStrike-a uključuju automatsko otkrivanje ranjivosti. XSSStrike može automatski pretraživati ciljani URL za XSS ranjivosti, čime se štede vrijeme i resursi sigurnosnih analitičara. Ova funkcionalnost omogućava brzu identifikaciju potencijalnih sigurnosnih prijetnji bez potrebe za ručnim pretraživanjem.

Generiranje *payload* unosa je još jedna značajna značajka XSSStrike-a. Alat koristi različite unose za testiranje ranjivosti, uključujući one koji ciljaju različite kontekste unutar HTML i JavaScript koda. Ova raznolikost u unosima osigurava da alat može identificirati širok spektar XSS ranjivosti.

Analiza koda je također ključna značajka XSSStrikea. Alat provodi dubinsku analizu koda web stranice kako bi identificirao i iskoristio XSS ranjivosti, uzimajući u obzir različite filtere i zaštite koje bi mogle biti implementirane. Ova sposobnost omogućava preciznije i učinkovitije prepoznavanje sigurnosnih propusta.

```
xssstrike –url http://target.com

http://192.168.100.56/sugarcrm/index.php?module=Accounts&action=Sh
owDuplicates

      XSStrike v3.1.5

[~] Checking for DOM vulnerabilities
[+] WAF Status: Offline
[!] Testing parameter: module
[-] No reflection found
[!] Testing parameter: action
[-] No reflection found

http://192.168.100.56/sugarcrm/index.php?module=Contacts&action=Sh
owDuplicates

      XSStrike v3.1.5

[~] Checking for DOM vulnerabilities
[+] WAF Status: Offline
[!] Testing parameter: module
[-] No reflection found
[!] Testing parameter: action
[-] No reflection found
```

Rezultat 25. Isječak XSSStrike napada

4.3.5. WAFNinja

WAFNinja [27] je alat dizajniran za testiranje i zaobilaženje Web Application Firewall (WAF) sustava. Iako su WAF-ovi učinkoviti u filtriranju većine zlonamjernih zahtjeva, postoji potreba za alatima kao što je WAFNinja koji pomažu sigurnosnim stručnjacima u evaluaciji sigurnosti WAF-a i otkrivanju potencijalnih slabosti.

Alat je sposoban simulirati različite napade na web aplikacije kako bi utvrdio koliko je WAF učinkovit u prepoznavanju i blokiranju tih napada. Razvijen je s ciljem pružanja sveobuhvatne platforme za testiranje raznih tehnika zaobilaženja WAF-ova i različite tehnike kamuflaže *payload* unosa.

Ovi unosi mogu uključivati SQL injekcije, XSS, i druge vrste napada. Ako WAF blokira standardne napadačke obrasce, WAFNinja koristi sofisticirane metode za zaobilaženje, uključujući modifikaciju i obfuskaciju napadačkih kodova kako bi izbjegao detekciju.

WAFNinja također može provoditi *fuzzing* napade, gdje sustavno šalje razne modificirane zahtjeve na ciljni server kako bi otkrio ranjive točke koje bi WAF mogao propustiti.

Fuzzing je tehnika koja pomaže u pronalaženju nepoznatih ranjivosti generiranjem neočekivanih ili slučajnih podataka i unošenjem istih u aplikaciju. Ako WAFNinja otkrije da određeni modificirani unos prolazi kroz WAF, to može ukazivati na potencijalnu ranjivost ili slabost u WAF-ovoj zaštiti.

WAFNinja - Penetration testers favorite for WAF Bypassing

URL: http://192.168.0.113/wp-admin/theme-editor.php?file=FUZZ&theme=FUZZ
TYPE: sql
DELAY: 0
PROXY:
PREFIX:
POSTFIX:

Fuzz	HTTP Status	Content-Length	Expected	Output	Working
123<234	200	2810	123<234	123%3C2	Probably
9928!=1239	200	2820	9928!=1239	9928%21%3D	Probably
abc'	200	2804	abc'	abc%	Probably
abc"	200	2804	abc"	abc%	Probably
or	200	2796	or	or	Yes
and	200	2798	and	and	Yes

Slika 3. Rezultat WAFNinja napada

Faza iskorištavanja predstavlja ključnu komponentu penetracijskog testiranja, omogućujući sigurnosnim stručnjacima da demonstriraju stvarne mogućnosti napada na identificirane ranjivosti. Korištenje naprednih alata poput Metasploit Frameworka, Hydre, SQLMapa, XSSStrikea i WAFNinje omogućava temeljito testiranje sigurnosti različitih sustava i aplikacija. Svaki od ovih alata nudi specifične funkcionalnosti koje su ključne za otkrivanje i iskorištavanje ranjivosti, pružajući dubinsku analizu i mogućnost zaobilaženja sofisticiranih sigurnosnih mehanizama.

Metasploit, sa svojom širokom bazom modula za iskorištavanje ranjivosti, omogućava precizno ciljanje i iskorištavanje ranjivosti u mrežnim i web okruženjima. Hydra nudi efikasnu metodu za *brute-force* napade na različite mrežne protokole, dok SQLMap automatizira proces otkrivanja i iskorištavanja SQL injekcija, čime značajno povećava efikasnost sigurnosnih procjena.

XSSStrike omogućava brzo i učinkovito identificiranje XSS ranjivosti, a WAFNinja pruža napredne tehnike za testiranje i zaobilaženje Web Application Firewall sustava.

Kombiniranjem rezultata dobivenih iz ovih alata, penetracijski testeri mogu stvoriti cjelovitu sliku o sigurnosnom stanju ciljanih sustava i aplikacija. Ovo im omogućava ne samo da identificiraju potencijalne prijetnje, već i da preporuče odgovarajuće mjere za poboljšanje sigurnosti. U konačnici, cilj iskorištavanja u penetracijskom testiranju je osigurati da su sustavi otporni na napade, štiteći kritične podatke i osiguravajući integritet i pouzdanost mrežnih i aplikacijskih resursa.

4.4. Izvještavanje

Kako se radi o automatiziranom penetracijskom testu, ključno je razumjeti da program nema uvid u kontekst specifičnih aspekata testiranja. Stoga, dobiveni izvještaj služi kao vodič penetracijskom testeru, pomažući mu da shvati što se dogodilo u svakom modulu testiranja. Ovaj izvještaj omogućava testeru da pravilno i precizno opiše sve aspekte testiranja u svom konačnom izvještaju.

Svi relevantni izvještaji povezani s programom i različitim modulima pohranjeni su u direktorijima *scan_reports* i *attack_reports*. Ovi direktoriji sadrže detaljne informacije koje pružaju dublji uvid u rezultate skeniranja i napada. Ako općenite informacije unutar

datoteke izvještaja *report.pdf* nisu dovoljne za potpuno razumijevanje rezultata, tester može pristupiti ovim direktorijima i pronaći vrlo detaljne opise svakog skena i napada.

Direktorij *scan_reports* sadrži sveobuhvatne rezultate mrežnog ili web skeniranja. Ovaj direktorij omogućava testeru detaljnu analizu svih aspekata mrežnog okruženja i pronalazak mogućih sigurnosnih prijetnji.

Direktorij *attack_reports* pruža detaljne informacije o različitim napadima provedenim tijekom testiranja. Ovdje se nalaze opisi uspješnih napada, korištenih metoda iskorištavanja, kao i procjena utjecaja tih napada na sigurnost mreže. Ove informacije su ključne za razumijevanje stvarnog utjecaja otkrivenih ranjivosti i pomažu testeru da precizno ocijeni razinu rizika.

Završni izvještaj *report.pdf* ukratko objašnjava nalaze i preporuke te pruža opći pregled rezultata testiranja.

Table of Contents

Slika 4. Sadržaj Izvještaja

Scan Report: nessus.json

Vuln Index	Plugin ID	Plugin Name	Severity	Plugin Family	Count
1	10107	HTTP Server Type and Version	0	Web Servers	2
2	22964	Service Detection	0	Service detection	2
3	24260	HyperText Transfer Protocol (HTTP) Information	0	Web Servers	2
4	48204	Apache HTTP Server Version	0	Web Servers	2
5	10287	Traceroute Information	0	General	1
6	10863	SSL Certificate Information	0	General	1
7	11219	Nessus SYN scanner	0	Port scanners	1
8	11936	OS Identification	0	General	1
9	19506	Nessus Scan Information	0	Settings	1
10	21643	SSL Cipher Suites Supported	0	General	1
11	25220	TCP/IP Timestamps Supported	0	General	1
12	35291	SSL Certificate Signed Using Weak Hashing Algorithm	2	General	1
13	35716	Ethernet Card Manufacturer Detection	0	Misc.	1
14	45590	Common Platform Enumeration (CPE)	0	General	1
15	50845	OpenSSL Detection	0	Service detection	1
16	51192	SSL Certificate Cannot Be Trusted	2	General	1
17	54615	Device Type	0	General	1
18	56984	SSL / TLS Versions Supported	0	General	1
19	57041	SSL Perfect Forward Secrecy Cipher Suites Supported	0	General	1
20	57582	SSL Self-Signed Certificate	2	General	1
21	66334	Patch Report	0	General	1
22	69551	SSL Certificate Chain Contains RSA Keys Less Than 2048 bits	1	General	1

Slika 5. Rezultat Nessus skeniranja u izvještaju, lakše za razumjeti nego originalni JSON dokument

Scan Report: nmap.txt

Nmap scan report for 192.168.100.56

Host is up (0.00017s latency).

MAC Address: 08:00:27:F7:09:7B (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 0.13 seconds

Running Quick Scan...

Starting Nmap 7.92 (https://nmap.org) at 2024-06-06 11:34 EDT

Nmap scan report for 192.168.100.56

Host is up (0.00032s latency).

Not shown: 97 filtered tcp ports (no-response)

PORT STATE SERVICE

22/tcp closed ssh

80/tcp open http

443/tcp open https

MAC Address: 08:00:27:F7:09:7B (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 2.04 seconds

Running Service Version Detection...

Starting Nmap 7.92 (https://nmap.org) at 2024-06-06 11:34 EDT

Nmap scan report for 192.168.100.56

Host is up (0.00035s latency).

Not shown: 997 filtered tcp ports (no-response)

PORT STATE SERVICE VERSION

22/tcp closed ssh

80/tcp open http Apache httpd

Slika 6. Rezultat Nmap Skeniranja u Izvještaju

Automatizirani izvještajni sustav omogućava penetracijskom testeru da efikasno prati i dokumentira sve faze testiranja, osiguravajući da nijedan ključni detalj ne bude zanemaren. Kroz korištenje ovih izvještaja, tester može osigurati visoku razinu preciznosti i potpunosti u svom radu, što je ključno za efektivnu procjenu i poboljšanje sigurnosti mrežnog okruženja.

Zaključak

U ovom radu istraženo je i implementirano automatizirano penetracijsko testiranje u složenim mrežnim okruženjima, koristeći skripte razvijene u programskom jeziku Python, te razne alate iz Kali Linux distribucije. Ova metoda omogućuje bržu i učinkovitiju procjenu sigurnosti mreža, što je posebno važno u današnjim dinamičnim i složenim mrežnim okruženjima.

Automatizirano penetracijsko testiranje predstavlja značajan napredak u sigurnosnoj industriji, omogućujući penetracijskim testerima da efikasno identificiraju ranjivosti bez potrebe za ručnim radom koji može biti dugotrajan i sklon greškama. Korištenjem niza alata poput Nmapa, Nessusa, Burp Suitea, Metasploita i drugih, omogućena je sveobuhvatna analiza sigurnosnog stanja mreže, od početnog izviđanja i skeniranja, preko iskorištavanja ranjivosti, pa sve do završnog izvještavanja.

Posebna pažnja posvećena je organiziranju izvještaja i izvještavanju općenito, koje je ključna komponenta cijelog procesa. Dobiveni izvještaji služe kao vodič penetracijskim testerima, pružajući im detaljne informacije o svim fazama testiranja. Izvještaji su pohranjeni u direktorijima *scan_reports* i *attack_reports*, koji sadrže sveobuhvatne podatke o rezultatima skeniranja i napada. U slučaju potrebe za detaljnijim uvidom, ovi direktoriji pružaju dodatne informacije koje nadopunjuju skraćene izvještaje u datoteci *report.pdf*.

Automatiziranje penetracijskog testiranja nije samo poboljšalo učinkovitost procesa, već je i omogućilo konzistentnost rezultata, smanjilo ljudske pogreške i osiguralo bržu identifikaciju sigurnosnih prijetnji. Uvođenjem ovakvih sustava, organizacije mogu značajno unaprijediti svoju sigurnost i smanjiti rizik od potencijalnih kibernetičkih napada.

Međutim, unatoč svim prednostima, automatizirani alati nisu zamjena za iskusne sigurnosne stručnjake. Ljudski faktor i dalje ostaje ključan za interpretaciju rezultata, donošenje odluka o prioritetima u otklanjanju ranjivosti i prilagodbu metoda testiranja specifičnim potrebama organizacije. Stoga je idealna praksa kombinirati automatizirane alate s ručnim pregledom i analizom, kako bi se osigurala maksimalna pokrivenost i sigurnost mrežnog okruženja.

Automatizirano penetracijsko testiranje predstavlja snažan alat u kibernetičkoj sigurnosti, pružajući brzu i učinkovitu metodu za identifikaciju i uklanjanje sigurnosnih prijetnji. Ovaj diplomski rad doprinosi razumijevanju i implementaciji ovih sustava, ističući važnost kontinuiranog razvoja i prilagodbe sigurnosnih praksi u skladu s evolucijom prijetnji u digitalnom svijetu.

Nakon završetka ovog diplomskog rada, planiram nastaviti raditi na programu kako bih ga dodatno unaprijedio, proširio njegove funkcionalnosti i držao ga u koraku s novim alatima i tehnologijama.

Na ovaj način, nadam se da će program postati još korisniji alat za mene i ostale sigurnosne stručnjake, doprinositi poboljšanju sigurnosti sustava i pružiti značajnu podršku u borbi protiv kibernetičkih prijetnji.

Literatura

[1] F. Abu-Dabaseh and E. Alshammari, "Automated Penetration Testing: An Overview," 2018, pp. 121-129. doi: 10.5121/csit.2018.80610.

[2] V. Saber, D. ElSayad, A. M. Bahaa-Eldin and Z. Fayed, "Automated Penetration Testing, A Systematic Review," 2023 International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC), Cairo, Egypt, 2023, pp. 373-380. doi: 10.1109/MIUCC58832.2023.10278377.

[3] M. P. Shah, "Comparative Analysis of the Automated Pen Test Tools," MSc Internship, National College of Ireland, 2020.

[4] G. Jayasuryapal, P. M. Pranay, H. Kaur and Swati, "A Survey on Network Penetration Testing," 2021 2nd International Conference on Intelligent Engineering and Management (ICIEM), London, United Kingdom, 2021, pp. 373-378. doi: 10.1109/ICIEM51511.2021.9445321.

[5] S. Shah and B. M. Mehtre, "An automated approach to Vulnerability Assessment and Penetration Testing using Net-Nirikshak 1.0," 2014 IEEE International Conference on Advanced Communications, Control and Computing Technologies, Ramanathapuram, India, 2014, pp. 707-712. doi: 10.1109/ICACCCT.2014.7019182.

[6] O. Valea and C. Oprişa, "Towards Pentesting Automation Using the Metasploit Framework," 2020 IEEE 16th International Conference on Intelligent Computer Communication and Processing (ICCP), Cluj-Napoca, Romania, 2020, pp. 171-178. doi: 10.1109/ICCP51029.2020.9266234.

[7] H. -h. E, M. -n. Song, J. -d. Song, Y. Li and Z. -j. Ren, "The Research of Service Network Based on Complex Network," 2010 International Conference on Service Sciences, Hangzhou, China, 2010, pp. 203-207. doi: 10.1109/ICSS.2010.41.

[8] S. Kadam, B. Mahajan, M. Patanwala, P. Sanas, and S. Vidyarthi, "Automated Wi-Fi penetration testing," 2016. doi: 10.1109/iceeot.2016.7754855.

[9] Nmap. Dostupno na poveznici: https://nmap.org.

[10] TCPTraceroute. Dostupno na poveznici: https://linux.die.net/man/1/tcptraceroute.

[11] WhatWeb. Dostupno na poveznici: https://github.com/urbanadventurer/WhatWeb.

[12] dirsearch. Dostupno na poveznici: https://github.com/maurosoria/dirsearch.

[13] nikto. Dostupno na poveznici: https://github.com/sullo/nikto.

[14] Nessus. Dostupno na poveznici: https://www.tenable.com/products/nessus.

[15] Wireshark. Dostupno na poveznici: https://www.wireshark.org/.

[16] Who.is. Dostupno na poveznici: https://who.is/.

[17] Shodan. Dostupno na poveznici: https://www.shodan.io/.

[18] theHarvester. Dostupno na poveznici: https://github.com/laramies/theHarvester.

[19] Sublist3r. Dostupno na poveznici: https://github.com/aboul3la/Sublist3r.

[20] wafw00f. Dostupno na poveznici: https://github.com/EnableSecurity/wafw00f.

[21] WPScan. Dostupno na poveznici: https://wpscan.com/.

[22] Burp Suite. Dostupno na poveznici: https://portswigger.net/burp.

[23] Metasploit. Dostupno na poveznici: https://www.metasploit.com/.

[24] THC-Hydra. Dostupno na poveznici: https://github.com/vanhauser-thc/thc-hydra.

[25] sqlmap. Dostupno na poveznici: https://github.com/sqlmapproject/sqlmap.

[26] XSStrike. Dostupno na poveznici: https://github.com/s0md3v/XSStrike.

[27] WAFNinja. Dostupno na poveznici: https://github.com/khalilbijjou/WAFNinja.

Sažetak

Automatiziranje Penetracijskog Testiranja

u Složenim Mrežnim Okruženjima

Ovaj diplomski rad bavi se automatiziranim penetracijskim testiranjem u složenim mrežnim okruženjima, koristeći skripte razvijene u Pythonu i alate iz Kali Linuxa. Korišteni alati uključuju Nmap, Nessus, Burp Suite, Metasploit i mnoge druge, čime se omogućava efikasna identifikacija sigurnosnih ranjivosti. Automatiziranje procesa penetracijskog testiranja omogućava bržu i precizniju procjenu sigurnosti, smanjujući potrebu za ručnim testiranjem. Rezultati svih testiranja pohranjeni su u odgovarajućim direktorijima za slučaj potrebe detaljnije analize.

Ključne riječi:

- Sigurnost
- Python
- Kali Linux
- Automatiziranje
- Penetracijsko testiranje

Automating Penetration Testing

in Complex Network Environments

This Master Thesis deals with automated penetration testing in complex network environments using Python scripts and tools from Kali Linux. The tools used include Nmap, Nessus, Burp Suite, Metasploit, and many others, enabling efficient identification of security vulnerabilities. Automating the penetration testing process allows for faster and more accurate security assessments, reducing the need for manual testing. The results of all tests are stored in appropriate directories in case of the need for a more detailed analysis.

Keywords:

- Security
- Python
- Kali Linux
- Automating
- Penetration testing