Metodologija penetracijskog testiranja

Cetinić, Leon

Master's thesis / Diplomski rad

2021

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Transport and Traffic Sciences / Sveučilište u Zagrebu, Fakultet prometnih znanosti**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:119:515393

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

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

Repository / Repozitorij:

Faculty of Transport and Traffic Sciences - Institutional Repository

SVEUČILIŠTE U ZAGREBU
FAKULTET PROMETNIH ZNANOSTI

Leon Cetinić

METODOLOGIJA PENETRACIJSKOG TESTIRANJA

DIPLOMSKI RAD

Zagreb, rujan 2021.

SVEUČILIŠTE U ZAGREBU

FAKULTET PROMETNIH ZNANOSTI

DIPLOMSKI RAD

METODOLOGIJA PENETRACIJSKOG TESTIRANJA

PENETRATION TESTING METHODOLOGY

Mentor : prof. dr. sc. Dragan Peraković

Student : Leon Cetinić

JMBAG: 0135244366

Zagreb, rujan 2021.

SVEUČILIŠTE U ZAGREBU
FAKULTET PROMETNIH ZNANOSTI
POVJERENSTVO ZA DIPLOMSKI ISPIT

Zagreb, 9. lipnja 2021.

Zavod: **Zavod za informacijsko komunikacijski promet**
Predmet: **Sigurnost i zaštita informacijsko komunikacijskog sustava**

DIPLOMSKI ZADATAK br. 6504

Pristupnik: **Leon Cetinić (0135244366)**
Studij: Promet
Smjer: Informacijsko-komunikacijski promet

Zadatak: **Metodologija penetracijskog testiranja**

Opis zadatka:

U radu je potrebno dati općeniti prikaz problematike penetracijskog testiranja. Potrebno je utvrditi sigurnosne aspekata informacijskog sustava prema odabranom realnom primjeru iz prakse. Dati prikaz analize i funkcionalnosti dostupnih alata za tu namjenu. Kroz praktičan rad, prikazati provedbu penetracijskog testiranja i otkrivanje ranjivosti nad virtualnim okruženjem. Detaljno prikazati analizu prikupljenih podataka te prikazati završni zaključak rada.

Mentor:

prof. dr. sc. Dragan Peraković

Predsjednik povjerenstva za diplomski ispit:

METODOLOGIJA PENETRACIJSKOG TESTIRANJA

SAŽETAK

Svrha ovog rada je analizirati i prikazati saznanja o penetracijskom testiranju kao načinu zaštite informacijskih sustava te definirati metodologiju penetracijskog testiranja. Proces testiranja provodi se pomoću faze pripreme i prikupljanja informacija te izvršavanja napada nad sustavom i analiziranjem podataka dobivenih kroz cjelokupno istraživanje. Cilj rada je prikazati postupak penetracijskog testiranja primjenom praktičnog dijela diplomskog rada, gdje će se provesti testiranje nad virtualnim okruženjem koristeći metode i alate opisane u radu. Praktični dio rezultirat će određenim ranjivostima sustava za koja će se ponuditi sigurnosna rješenja, dok će se kroz teorijski dio opisati metodologija izvođenja i analiza ponuđenih rješenja.

KLJUČNE RIJEČI: penetracijsko testiranje; metodologija; prikupljanje podataka; vektori napada; sigurnost informacijskog sustava

PENETRATION TESTING METHODOLOGY

SUMMARY

The purpose of this paper is to analyze and present the findings about penetration testing as a way of protecting information systems and to define the methodology of penetration testing. The testing process is carried out through the stages of preparation and collection information, execution attacks on the system and analysis of data obtained throughout the research. The aim of the paper is to present the procedure of penetration testing by applying the practical part of the thesis, where testing in a virtual environment will be conducted using the methods and tools described in the paper. The practical part will result in certain vulnerabilities of the system for which security solutions will be offered, while the theoretical part will describe the methodology of implementation and analysis of the offered solutions.

KEY WORDS: penetration testing; methodology; collecting information; vector attacks; information system security

SADRŽAJ

1. Uvod ... 1

2. Općenito o penetracijskom testiranju ... 3

2.1. Potreba za provođenjem penetracijskog testiranja ... 5

2.2. Klasifikacija penetracijskog testiranja ... 8

2.2.1. Vrste penetracijskog testiranja s obzirom na pristup informacijama ... 9

2.2.2. Vrste penetracijskog testiranja s obzirom na agresivnost ... 11

2.2.3. Vrste penetracijskog testiranja s obzirom na perimetar izvođenja ... 11

2.2.4. Vrste penetracijskog testiranja s obzirom na tehniku izvođenja ... 11

2.2.4.1. Mrežni penetracijski testovi ... 12

2.2.4.2. Fizički penetracijski testovi ... 14

2.2.4.3. Penetracijsko testiranje aplikacija ... 14

2.2.4.4. Penetracijsko testiranje tehnikom socijalnog inženjeringa ... 15

2.2.5. Vrste penetracijskog testiranja s obzirom na početnu točku izvođenja ... 16

3. Utvrđivanje sigurnosnih aspekata informacijskog sustava ... 17

3.1. Klasifikacija prijetnji informacijskih sustava ... 17

3.2. Metode zaštite informacijskog sustava ... 19

3.3. Faze procesa penetracijskog testiranja ... 21

3.3.1. Prikupljanje podataka ... 22

3.3.2. Mapiranje mreže ... 23

3.3.3. Identificiranje ranjivosti ... 24

3.3.4. Iskorištavanje ranjivosti ... 24

3.3.5. Dokumentacija i izvještavanje ... 26

3.4. Standardi penetracijskog testiranja ... 27

3.4.1. Open Source Security Testing Methodology Manual ... 27

3.4.2. The Open Web Application Security Project ... 28

3.4.3. National Institute of Standards and Technology ... 29

4. Analiza i funkcionalnosti alata ... 30

4.1. Kali Linux ... 30

4.2. Nessus ... 32

4.3. Nmap ... 33

4.4. SQLmap ... 35

4.5. DirBuster ... 36

4.6. Netcat ... 37

4.7. JoomScan 37
4.8. John The Ripper 38
5. Provedba penetracijskog testiranja i otkrivanje ranjivosti nad virtualnim okruženjem 39
5.1. Faza prikupljanja podataka 39
5.2. Faza mapiranja i identificiranja ranjivosti 40
5.3. Faza iskorištavanja ranjivosti 42
6. Analiza prikupljenih podataka 47
6.1. Izvještaj o fazi prikupljanja podataka 47
6.2. Izvještaj o fazi mapiranja mreže 49
6.3. Izvještaj o fazi identificiranja ranjivosti 50
6.4. Izvještaj o fazi iskorištavanja ranjivosti 51
7. Zaključak 52
LITERATURA 53
POPIS KRATICA 56
POPIS SLIKA 58
POPIS TABLICA 58

1. Uvod

Informacijska sigurnost važan je faktor u poslovanju organizacija iz razloga što nudi povjerljivost, raspoloživost i cjelovitost podataka, ali i sustava koji su neophodni za siguran i održiv rad. Razvojem tehnologije povećava se i broj kibernetičkih napada na računalne sustave, a kako bi se izbjegli neželjeni događaji, potrebno je zaštiti sustav određenim sigurnosnim aspektima. Jedan od primjera održavanja sigurnosti je i penetracijsko testiranje u kojem se stručna osoba postavlja u ulogu napadača kako bi pronašao sigurnosne propuste unutar sustava. Ispitivač uz postojeće znanje, iskustvo, predviđene alate i metodologiju pokušava pronaći ranjivosti sustava poput neovlaštenog pristupa sustavu i podacima, onemogućavanje rada sustava ili dobivanje privilegija potrebnih za kontrolu sustava. Nakon testiranja, analiziraju se saznanja o pojedinim ranjivostima te o mogućim štetnostima po sustav. Prikupljena saznanja i informacije dokumentiraju se u izvješće kao pregled vezan uz penetracijsko testiranje. Izvješće opisuje faze izvođenja, efikasnost korištenih alata te prednosti i nedostatke kao rezultat ispitivanja. Na temelju zaključka donesenog u dokumentu određuje se potrebno djelovanje kako bi se dodatno osigurao sustav i kako bi se onemogućilo iskorištavanje utvrđenih ranjivosti.

Istraživanja su pokazala kako nedefinirani sigurnosni aspekti predstavljaju veliku problematiku prilikom razvoja organizacija i stvaranjem strateških planova te samim time postaju mete neovlaštenim napadačima. Zbog toga se smatra obavezno provođenje opsežnog penetracijskog testiranja najmanje jedan put godišnje s ciljem pružanja povećanja zaštite informacijskih sustava i procjene učinkovitosti sigurnosnih mjera. Uočeno je kako problematika kibernetičkih napada postaje sve veća, a razlog tome je svakodnevno otkrivanje novih prijetnji po sustav koje je teško kontrolirati i na adekvatan način se zaštiti od njih. Iz tog razloga penetracijski ispitivači moraju imati znanje za izvođenjem testiranja koje je potrebno kontinuirano nadograđivati u skladu s novim vrstama prijetnji. U ovom radu proveden je praktični dio penetracijskog testiranja korištenjem spoznaja i metodologije opisanih u radu, dok se kroz izvještaj testiranja nude saznanja kako ispravno zaštiti sustav.

Diplomski rad se sastoji od sedam funkcionalno povezanih cjelina:

1. Uvod
2. Općenito o penetracijskom testiranju
3. Utvrđivanje sigurnosnih aspekata informacijskog sustava
4. Analiza i funkcionalnosti alata
5. Provedba penetracijskog testiranja i otkrivanje ranjivosti nad virtualnim okruženjem
6. Analiza prikupljenih podataka
7. Zaključak

U drugom poglavlju definirane su glavne spoznaje penetracijskog testiranja kao i terminologija potrebna za razumijevanje rada. Objašnjene su potrebe i važnosti izvođenja penetracijskih testiranja koje su potkrijepljene grafičkim prikazima. U nastavku cjeline obrazložene su različite klasifikacije penetracijskog testiranja te koji pristup je potrebno primijeniti nad odgovarajućim organizacijama.

Treća cjelina prikazuje sigurnosne aspekte informacijskog sustava te važnost zaštite sustava i kritičnih podataka. Cjelina obuhvaća sigurnosne mehanizme koje se u praksi primjenjuju kako bi se smanjile mogućnosti prodora u sustave. Utvrđene su faze procesa penetracijskog testiranja koje se sastoje od prikupljanja, mapiranja, identificiranja, napada i dokumentacije kako bi testiranje bilo u skladu sa zakonima i propisanim pravilima.

Cjelina četiri pod nazivom analiza i funkcionalnosti alata opisuje metode i alate korištene prilikom praktičnog dijela ovog rada. Ovo poglavlje pobliže objašnjava rad alata te na koji način se njima može napadati sustav. Alati koji se opisuju u ovom dijelu rada su: Nessus, Nmap, DirBuster, SQLmap, Netcat, Joomscan i John The Ripper. Također, opisana je Kali Linux distribucija koja se koristi prilikom provođenje penetracijskog testiranja te proces postavljanja virtualnog okruženja.

Peto poglavlje odnosi se na provođenje penetracijskog testiranja i otkrivanje ranjivosti nad virtualnim okruženjem. U ovom poglavlju primjenjuje se metodologija i sva ostala saznanja prikupljena u radu s ciljem uspješne provedbe praktičnog dijela te napada nad virtualnim okruženjem u svrhu dobivanja udaljenog pristupa nad ranjivim sustavom.

Šesta cjelina je analiziranje prikupljenih podataka koja se prikazuje kao dokumentirani izvještaj koji se u praksi dostavlja organizacijama na kraju testiranja kako bi se uočili sigurnosni propusti s kojima se organizacija susreće te su im ponuđena rješenja zaštite sustava u svrhu povećanja sigurnosnih aspekata.

Posljednja je cjelina zaključak koji se referencira na krajnja razmišljanja i rezultate prikupljene u ovome radu. U zaključku je sažet cijeli diplomski rad i prikazuje glavne spoznaje vezane uz temu rada.

2. Općenito o penetracijskom testiranju

Učestalost korištenja i razvoj informacijsko komunikacijskog (IK) sustava u javnim i privatnim sektorima zahtijeva strateške pristupe prilikom zaštite privatnosti korisnika i organizacija. Posljednjih 40 godina konstantno se povećava potreba za komunikacijom na daljinu te prijenos i razmjenu podataka. Prednosti koje se postižu IK sustavima je razvoj novih usluga, tehnologija i različitih terminalnih uređaja poput IoT i Cloud Computing-a. Također, povećava se količina podataka koje generiraju korisnici i uređaji, a prometni sadržaj vrlo često sadrži povjerljive i osjetljive informacije privatnih i poslovnih korisnika. Nacionalnim programima definiraju se organizacijski i upravljački aspekti informacijske sigurnosti na nacionalnim razinama sa ciljem razvoja zakona, propisa, metoda i postupaka u okvirima informacijske sigurnosti. Republika Hrvatska 2015. godine kao članica Europske Unije izdala je nacionalnu strategiju vezanu uz informacijsku sigurnost kojom se nastoji pratiti dinamika razvoja IK okruženja i ranjivosti sa ciljem rješavanja složenih pitanja kibernetičke sigurnosti. Republika Hrvatska svoja strateška planiranja temelji na smjernicama izdanih od strane ENISA-e (engl. *European Network and Information Security Agency*), vodećeg tijela Europske Unije za pitanja o kibernetičkoj sigurnosti [1]. Zbog potrebe za zaštitom informacijskih sustava proizlaze određeni mehanizmi zaštite nad različitim sustavima i uređajima. Velik broj uređaja, ali i kompleksnost sustava zahtijeva kombiniranje različitih metoda kako bi se omogućilo sigurno djelovanje sustava. Cilj zaštite je osigurati sve aspekte sigurnima od nenamjernih prijetnji kako bi sustavi bili dostupni i ispravni. Jedan od primjera mehanizma zaštite je i penetracijsko testiranje koje se za razliku od ostalih načina zaštite sustava koristi sa svrhom pružanja preventivne zaštite.

Penetracijsko testiranje (*engl. Penetration testing*) definira se kao tehnika provođenja autoriziranog napada nad informacijskim sustavom gdje se uz odgovarajuće znanje, tehnike i alate oponaša proces napada stvarnog napadača. Proces testiranja sadrži ispitivanje ranjivosti sustava kao i pružanje dokaza o koncepciji i vrsti napada kako bi se dokazale ranjivosti i pružila ispravna rješenja za zaštitu informacijskog sustava. Proces uključuje aktivnu analizu sustava na potencijalne ranjivosti, uključujući lošu ili nepravilnu konfiguraciju, hardverske i softverske nedostatke te operativne i tehničke slabosti informacijskog sustava. Kvalitetnija izvedba testiranja kao rezultat daje cjelovitost procjene slabosti, odnosno višu razinu sigurnosti sustava. Prilikom izvođenja testiranja potrebno je odrediti pravila primjene, pravne implikacije i vrste informacija kojima se pristupa. Ispitivač penetracijskog testiranja (engl. *Pentester*) je stručna osoba za računalnu sigurnost specijalizirana za provođenje procesa penetracijskog testiranja nad sustavom koristeći različite vrste alata i metodologije za pronalaženje ranjivosti sustava. Osim pronalaženja ranjivosti, ima zadatak primijeniti napade u cilju dobivanja neovlaštenog pristupa na način koji bi potencijalni napadači to iskoristili u svrhu prikupljanja kritičnih podataka vezanih unutar informacijskog sustava.

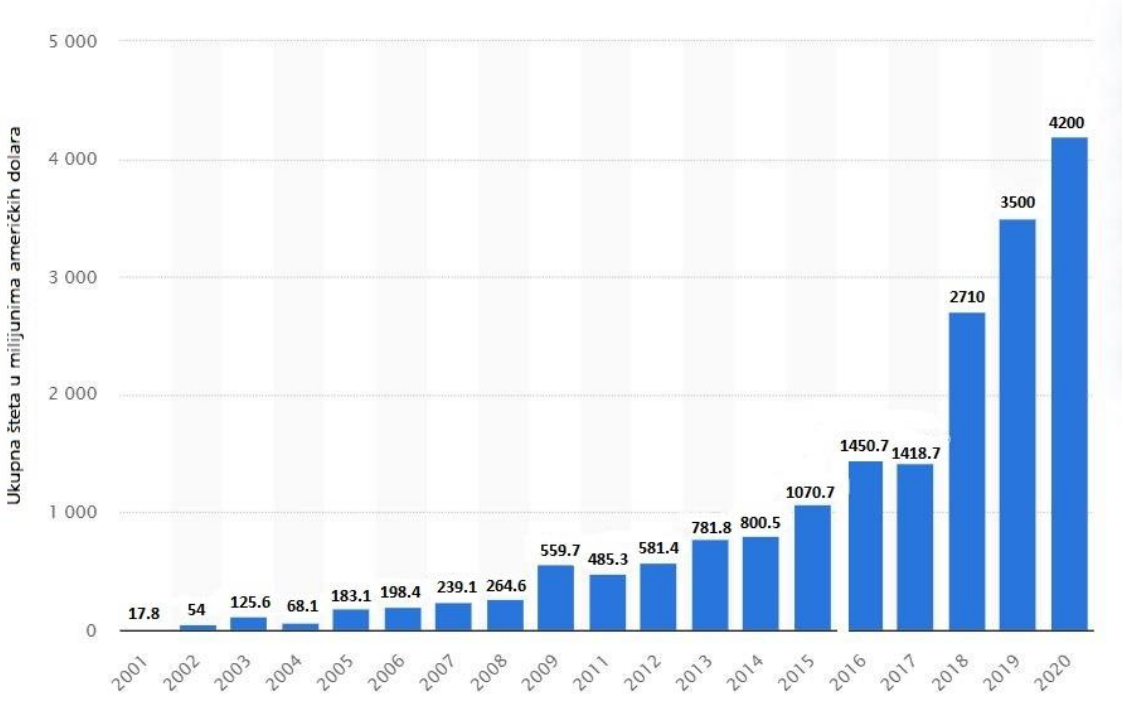
Perimetar izvođenja penetracijskog testiranja varira od klijenta do klijenta kao i postupci, metode i alati korišteni prilikom testiranja. Različiti su zahtjevi koje mogu zatražiti organizacije od pentestera ovisno o njihovim potrebama za zaštitom informacijskog sustava. Potreba organizacija za provedbom penetracijskog testiranja može imati za cilj zaštitu financijskog sektora, zaštitu osobnih i kritičnih podataka, preventivnu zaštitu sustava ili provođenje zaštite uslijed već počinjenog kibernetičkog napada. Različite tvrtke, organizacije ili entiteti imaju potrebu za zaštitom vlastitog sustava, među njima najveću potražnju za penetracijskim testiranjem imaju organizacije iz javnog i financijskog sektora čiji neovlašteni pristup može stvoriti veliki financijski deficit ili prijetnju državnoj sigurnosti. Većina organizacija nisu svjesna, dovoljno informirana ili jednostavno nemaju dovoljno znanja za kvalitetnom zaštitom stvarnih složenih komunikacijskih struktura i nad njima imaju malu ili gotovo nikakvu kontrolu [2]. Nadalje, njihovi rizici se povećavaju korištenjem aplikacija ili baza podataka u njihovoj infrastrukturi. Takvi dodatni rizici koji se ne kontroliraju u dovoljnoj mjeri mogu povećati broj sigurnosnih napada koji ostavljaju velike posljedice za tvrtke. Sigurnost kao jedan od glavnih pitanja informacijskog sustava potrebna je zbog rastuće povezanosti računala putem interneta, sve veće proširivosti, složenosti i različitosti sustava koje je gotovo nemoguće pratiti u korak s današnjim razvijanjem novih ili poboljšanja postojećih sustava. Svejedno, potrebno je poslovnim, ali i privatnim osobama omogućiti adekvatnu informacijsku zaštitu od potencijalnih rizika. Izvođenjem testova zaštite nije moguće dokazati apsolutnu sigurnost nekog sustava, jer niti jedan sustav nije u potpunosti zaštićen od svih oblika prijetnji, namjernih ili ne namjernih. Ono što svako testiranje mora omogućiti klijentu ili organizaciji nad kojom se vrši penetracijsko testiranje je određena razina sigurnosti, a nikako ne garantirati generalnom sigurnošću.

Sve veća pojava uređaja omogućila su stvaranje IoT (engl. *Internet of Things)* okruženja koja su ujedno pridonijela i povećanju sigurnosnih prijetnji u svijetu. Ukoliko se IoT okruženja sagledaju iz perspektive informacijsko komunikacijskog sustava, tada svaki pojedini element ovog sustava može biti opisan kroz ranjivosti kojima je podložan, što stvara iznimnu sigurnosnu prijetnju okruženju. U istraživanju kompanije US Business Insider, 39% ispitanika ukazalo je da su sigurnost i privatnost glavni problem IoT okruženja. Međutim, kako bi se odgovorilo na sigurnosne prijetnje osnovana je IoTSF (engl. *Internet of Things Security Foundation*) organizacija sa ciljem promicanja znanja i pružanja sigurnosnih metoda za zaštitu IoT okruženja [3].

IoT uređaji ranjivi su na velik broj kibernetičkih napada, a jedan od vektora napada je i phishing, pomoću kojeg se pokušava pristupiti informacijama unutar IoT okruženja. Jedan od načina zaštite uređaja omogućuje se korištenjem umjetne inteligencije, a cilj obrane je zaštiti IoT uređaje od neželjenih poruka korištenjem određenih funkcija. Ovaj mehanizam zaštite najprije prikuplja i analizira elektroničku poštu, a zatim filtrira neželjene poruke iz dolaznog prometa te ih zatim razvrstava u neželjenu poštu [4].

2.1. Potreba za provođenjem penetracijskog testiranja

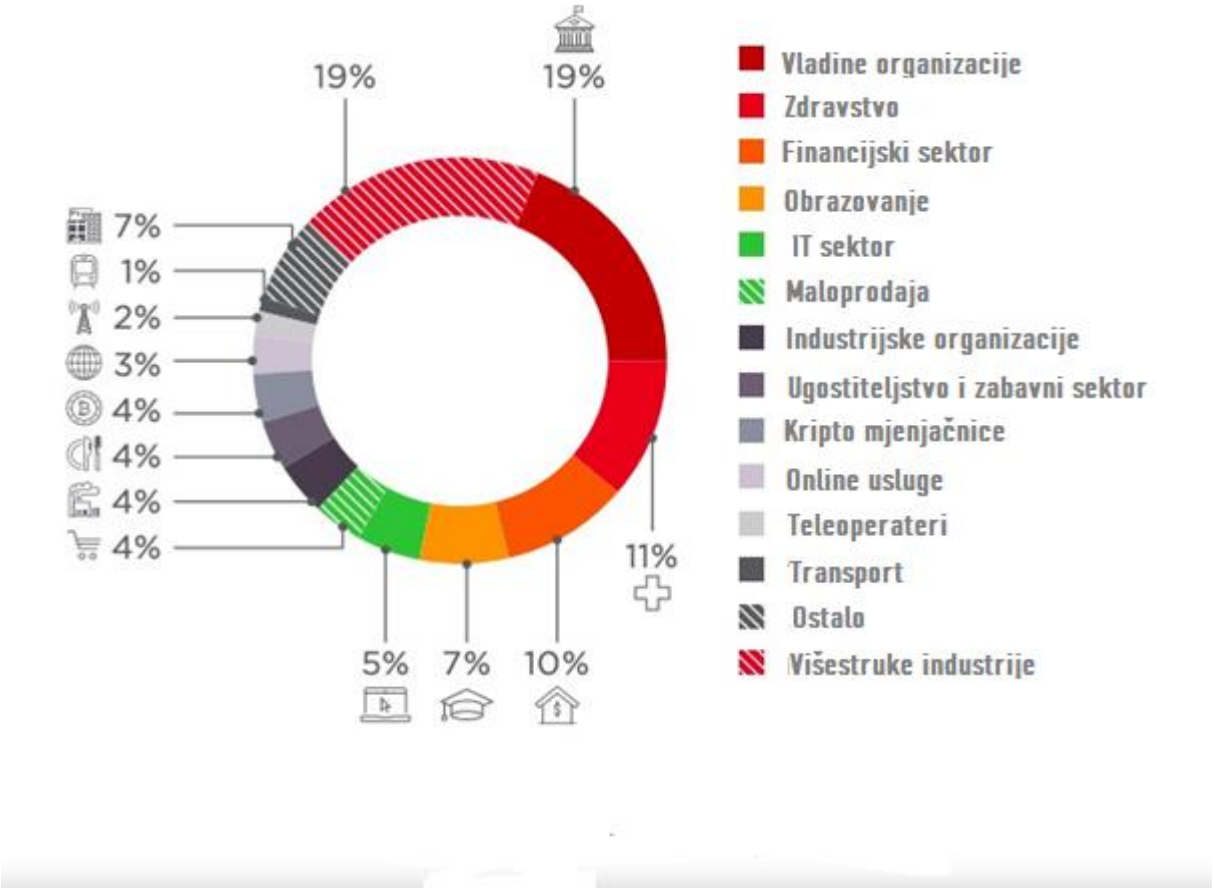
Sigurnost informacijskih sustava jedan je od glavnih značaja za organizacije, sukladno tome zahtjeva od njih da na adekvatan način zaštite svoju informacijsku imovinu slijedeći sveobuhvatan i strukturiran pristup koji pruža zaštitu od rizika s kojima se organizacija može suočiti. Sigurnost informacijskih sustava danas nailazi na veće prepreke i zahtjeve nego u prošlosti, razlog tome su rastuća povezanost računala putem interneta, globalna povezanost tržišta, manjak sigurnosnih pravila i protokola, razvoj novih tehnologija te sve veća proširivost sustava i nekontrolirani rast veličina i složenosti sustava. Različiti stručnjaci pokušavaju povećati sigurnost informacijskih sustava korištenjem raznih sigurnosnih mehanizama, među kojima je i postupak penetracijskog testiranja [5].



Grafikon 1. *Financijska šteta prouzročena kibernetičkim napadima (2001.-2020.)*
Izvor: [6]

Potreba za provođenjem penetracijskog testiranja proizlazi iz sve češćih napada na organizacije, najčešće s ciljem krađe financijskih sredstava tih organizacija. Prikaz toga može se uočiti na grafikonu 1 koji predstavlja vrijednost štete u milijunima američkih dolara prouzročenih kibernetičkim kriminalom u razdoblju od 2001. godine do 2020. godine. Graf se odnosi na područje cijelog svijeta te za one kibernetičke napade koji su prijavljeni. Podaci su prikupljeni od strane IC3 (engl. *Internet Crime Complaint Center*) organizacije koja je zadužena za prihvaćanje, razvoj i upućivanje kaznenih prijava vezanih uz pojavu internetskog kriminala. Graf prikazuje eksponencijalni rast financijskih gubitaka posljedicom kibernetičkog kriminala, a sličan eksponencijalni rast očekuje se i u narednim godinama. U odnosu na 2020. godinu u kojoj je ukupna šteta iznosila preko 4 milijarde američkih dolara, u 2010. godini ukupna šteta iznosila je gotovo devet puta manje. Najveću financijsku štetu imaju Sjedinjene Američke Države koje su 2014. godine činile 83,96% svih prijavljenih gubitaka uzrokovanih kibernetičkim kriminalom [6].

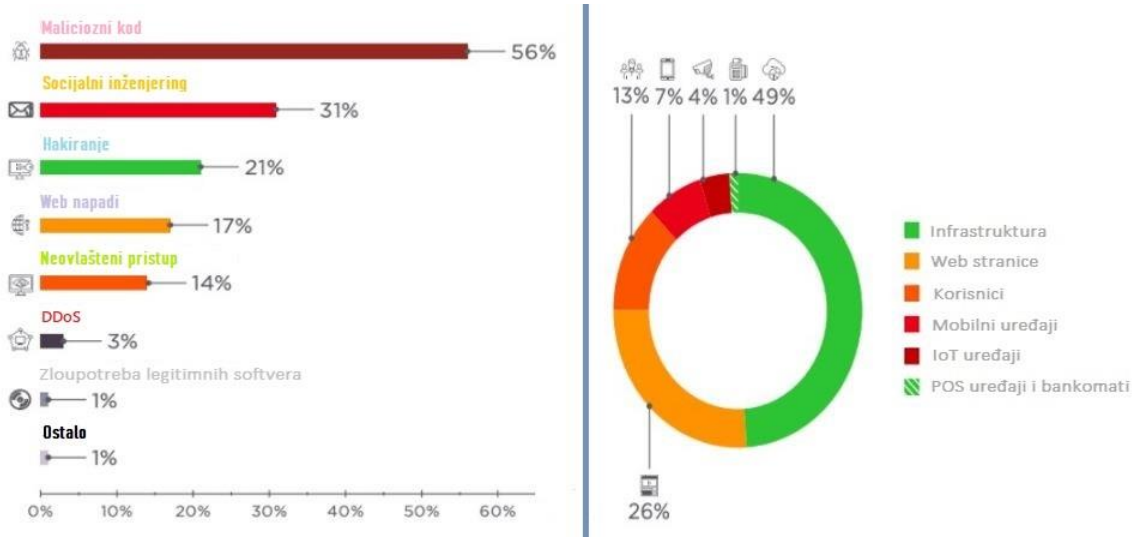
Testiranjem se omogućuje proaktivan pristup sigurnosti na način da se istražuju stvarni rizici i točno stanje infrastrukturne sigurnosti i prije nego se šteta dogodi. Ispitivači prilagođavaju svoje metodologije i alate vrsti testiranja, prilagođavaju se promjenama i određuju napad prilagođen stvarnim okruženjima. Provođenjem penetracijskog testa organizacijama se nude planovi za uklanjanje sigurnosnih ranjivosti te se ocjenjuje razina sigurnosne svijesti među zaposlenicima i učinkovitost sigurnosne politike. Na temelju rezultata testiranja, tvrtke mogu raditi planove i procjene za daljnja ulaganja i napredak organizacije. Ovisno o vrsti organizacije povećavaju se ili smanjuju mogućnosti za potencijalnim kibernetičkim napadima, zbog toga je potrebno procijeniti interes informacija ili financija organizacije te preventivno zaštititi one organizacije koji se smatraju najčešćim metama kibernetičkih napadača.



***Grafikon 2**. Učestalost kibernetičkih napada nad organizacijama*
Izvor: [7]

Grafikon 2 daje prikaz učestalosti kibernetičkih napada nad različitim organizacijama. Podacima iz 2018. godine zabilježeno je da su najrizičnije vladine organizacije koje su u 19% slučajeva mete napadača, uz vladine organizacije 19% napada pretrpe i razne višestruke industrije. Cilj napada nad vladinim organizacijama uglavnom je u svrhu neovlaštenog zadobivanja osjetljivih informacija, dok kod većine ostalih organizacija najčešći uzroci napada su financijski. Sve veći zamah imaju neovlašteni napadi nad kripto mjenjačnicama iz razloga što je lakše sakriti ukradene kripto valute od stvarnog novca [7]. U kolovozu 2021. godine nad kripto mjenjačnicom je ukradeno više od 600 milijuna dolara, što se smatra najvećim kibernetičkim napadom nad kripto mjenjačnicama.

Testiranja mogu prouzročiti negativne posljedice po sustav na način da zagušuju mrežu ili prouzroče pad sustava, stoga postoji mogućnost da se dogodi upravo ono što se testiranjem pokušava spriječiti. Kako bi se osiguralo od takvih scenarija potrebno je dobiti sva potrebna odobrenja te vršiti testiranja u vremenu koje neće napraviti štetu za samu organizaciju. Postoji mogućnost pogrešnih rezultata, na primjer ukoliko su zaposlenici bili pripremljeni za test nije stvoreno ispravno okruženje i nisu postignuti potpuno ispravni rezultati. Organizacije mogu steći lažnu sigurnost nakon provedbe penetracijskog testiranja misleći kako su u potpunosti zaštićeni. Organizacijama ključan faktor predstavljaju cijena i vrijeme potrebno za izvođenje testiranja i upravo zbog toga se određene tvrtke odlučuju za jeftine ili kratkotrajne penetracijske testove koji neće u potpunosti zahvatiti kritične dijelove sustava kojima je potrebna sigurnosna zaštita. Također, potrebno je obratiti pažnju i na pravne i regulatorne zakone koji nisu u svim aspektima u potpunosti definirani. U obzir se uzima da će ispitivači koji su vanjski suradnici imati za vrijeme, ali i nakon izvršenja testiranja saznanja o privatnim i kritičnim informacijama vezanih uz tu organizaciju te postoji mogućnost da iz određenih razloga kompromitiraju sustav ostavljajući otkrivene ranjivosti ili zloupotrijebe dobivene informacije.



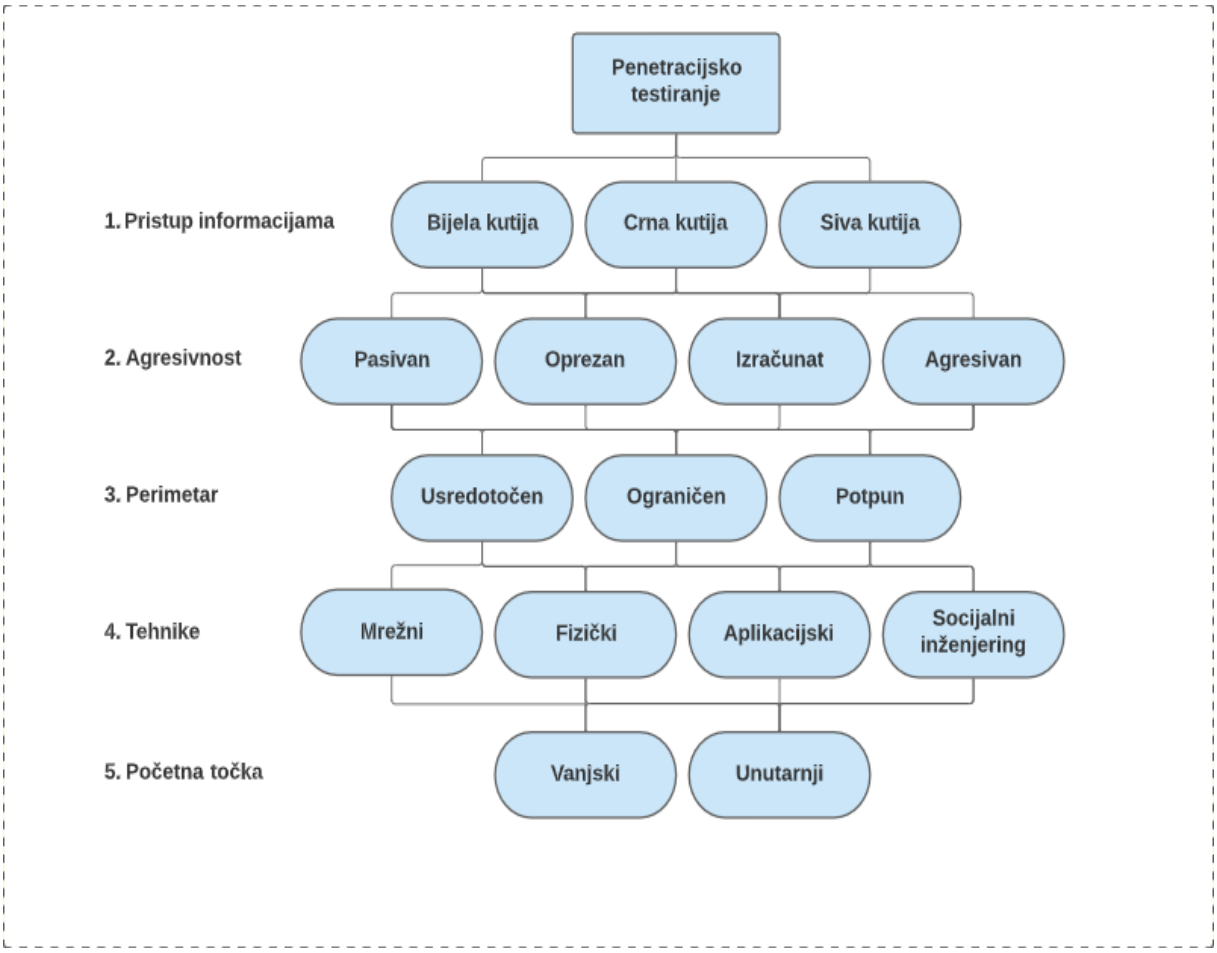
Grafikon 3. *Prikaz najčešćih vrsta napada i ciljnih sustava*
Izvor: [7]

Rezultati testiranja iz 2018. godine vidljivi su na grafikonu 3, gdje je s lijeve strane statistički prikaz najučestalijih vrsta napada, dok su s desne strane prikazani sustavi koji su najčešće mete kibernetičkih napada. Određeni kibernetički napadi mogu sadržavati više tipova napada nad jednim sustavom, a u grafikonu vidimo kako je više od pola napada uzrokovano korištenjem malicioznih kodova. Socijalni inženjering također ima veliki postotak korištenja, točnije 31%, a razlog tome je što se socijalni inženjering često koristi kao popratni napad koji dovodi do proboja i omogućuje eksploataciju drugih vrsta napada. Najučestaliji napadi vrše se nad infrastrukturama sustava, gotovo 50%, a slijede ih web stranice. U budućnosti se očekuje povećanje napada nad IoT uređajima čija primjena je još u razvoju. Postotak izvođenja napada nad POS uređajima i bankomatima smanjio se od 2017. godine s 3% na 1%, razlog tome je kvalitetnija zaštita tih sustava.

2.2. Klasifikacija penetracijskog testiranja

Klasifikacija penetracijskih testova moguća je na više načina, razlog tome je što različiti testovi zahtijevaju različite pristupe, metode, alate i potrebe. Klasifikacija testiranja ovisit će o veličini organizacije, sigurnosnim aspektima, vremenu i opsežnosti testiranja. S obzirom na izvođenje, penetracijsko testiranje možemo podijeliti na temelju pet kategorija:

- Pristup informacijama
- Agresivnost
- Perimetar
- Tehnike
- Početna točka



Slika 1. *Klasifikacija penetracijskog testiranja*
Izvor: [8]

Slika 1 prikazuje kategorizaciju prilikom izvođenja testiranja koje ovisno o razini sigurnosti pojedine tvrtke daje određene učinkovitosti uz minimalne rizike. Također, moguća je kombinacija različitih kriterija s ciljem ostvarivanja kvalitetnijih rezultata penetracijskog testiranja, pa je tako moguće odraditi testiranje svih tehnika koje bi činilo ispitivanje mrežnog, fizičkog i aplikacijskog dijela sustava uz korištenje tehnike socijalnog inženjeringa.

2.2.1. Vrste penetracijskog testiranja s obzirom na pristup informacijama

Penetracijske testove s obzirom na pristup (engl. *Approach*) možemo podijeliti u tri kategorije. Pristup temeljen na crnoj kutiji (engl. *Black box*) je zasnivan na potpunom nepoznavanju informacija o sustavu, pristup sive kutije (engl. *Grey box*) je pristup djelomično poznatom sustavu s minimalnim početnim informacijama, dok je pristup bijele kutije (engl. *White box)* potpuno poznat sustav ispitivaču i prije početka testiranja. Pristup testiranju razlikuje se ovisno o informacijama i tehnologijama koje poznajemo i koje su nam ponuđene prije samog izvođenja penetracijskog testiranja.

Black box je pristup temeljen na minimalnim ili gotovo nikakvim saznanjima ispitivača nad sustavom koji djeluje, odnosno ispitivač je postavljen u ulogu prosječnog napadača bez dodatnih internih saznanja i poznavanja ciljanog sustava. Pristupom temeljenim na crnoj kutiji pokušava se utvrditi ranjivosti koje je moguće iskoristiti izvan mreže, to znači da se zasniva na dinamičkoj analizi trenutno pokrenutih programa i konfiguracije u sustavu [9]. U nastavku rada bit će prikazan napad na virtualno okruženje koje će biti konfigurirano kao simulacija napada sustava black box tipa. Kao i u drugim ispitivanjima temeljenim na pristupu crne kutije zadatak prikazan u nastavku rada dati će minimalne informacije o sustavu i upravo je to razlog provedbe kvalitetne i opširne obrade faze izviđanja kako bi ispitivač mogao prodrijeti što dublje u mrežu ciljanog sustava te otkriti što više ranjivosti. Testiranjem black box pristupom izbjegavaju se testovi samo na onim dijelovima sustava koji se smatraju važnim, međutim isto tako se zbog toga mogu ostali elementi rizika podcijeniti. U konačnici, kao i nakon svakog testiranja ispitivač je dužan predati izvješće koje sadrži sve potrebne informacije vezane o realnom stanju ispitivanog sustava [10].

Neke od prednosti su:

- Black box pristup najbliži je stvarnim napadima koje napadači vrše nad sustavom te se često preporučuje u okvirima penetracijskog testiranja.
- Koristi se niz alata otvorenog koda i različite vrste tehnika za probijanje sustava.
- Black box tipu pristupaju pouzdani i visoko kvalificirani ispitivači koji koriste široki raspon napada poput XSS (engl. *Cross Site Scripting*), SQLi (engl. *SQL injection*), spoofing, DoS (engl. *Denial of Service*), zadobivanje privilegija pristupa i sličnih.

Nedostaci korištenja black box pristupa su sljedeći:

- Veća učinkovitost testiranja može klijentima dati lažan osjećaj sigurnosti iz razloga što ne postoji idealno provođenje penetracijskog testiranja i nije moguće identificirati sve vrste napada na koje je taj sustav ranjiv
- Pružanje black box pristupa je skuplji za organizacije u odnosu na ostala dva tipa pristupa.
- Vrijeme testiranja je ograničeno dok u stvarnim primjerima neovlaštenih napada to nije slučaj.

White box način ispitivanja provode ispitivači nad unaprijed poznatoj strukturi mreže s ciljem pronalaska ranjivosti internog sustava. Ono što white box pristup razlikuje od prethodnog je spoznaja ispitivača o strukturi softvera i arhitekturi sustava te postupak izvođenja koji se najčešće odnosi na kritične i jezgrene dijelove sustava. U penetracijskom testiranju temeljenom na white box pristupu ispitivači preskaču dijelove prikupljanja osnovnih informacija i izviđanja iz razloga što su im određeni podaci unaprijed poznati, poput informacija o mreži, operativnom sustavu, aplikacijama, pristupu, podacima izvornog koda, IP adresama, mrežnim karticama, konfiguraciji i slično. Ispitivači koriste različite vrste statičkih i dinamičkih analiza te razne alate s ciljem što veće zaštite internog sustava, a rezultat toga je izvještaj koji zahtijeva jasne i potpune informacije dobivene ovim načinom testiranja [11]. Moguća su testiranja sigurnosnih mjera viših razina poput zaštite pristupa podataka korištenjem različitih korisničkih računa s različitim razinama ovlasti.

Nekoliko je prednosti koje se propisuju ovom vrstom testiranja, a neke od njih su:

- Vrijeme vršenja je kraće zbog velikog broja informacija o sustavu kojima raspolažu ispitivači na samom početku testiranja.
- Veća količina informacija omogućuje temeljitije ispitivanje.
- Veća je šansa za otkrivanjem ranjivosti zbog toga što se ispitivanjem daje veći značaj u određene kritične dijelove sustava.
- Troškovi korištenja ovog tipa testiranja su manji nego u slučaju black box testiranja.

Međutim, osim prednosti white box pristup ima i svoje nedostatke, a neki od njih su:

- S obzirom na velik broj informacija koji su dostupni ispitivaču postoji šansa da će ispitivač zaobići neke od bitnih sigurnosnih prijetnji koje bi black box pristup detektirao.
- Može stvoriti lažnu pouzdanost organizacijama vezanim uz njihovu informacijsku sigurnost sustava iz razloga što white box pristupi nisu simulirani kao stvarni napadi.
- Zahtijevaju se sofisticiraniji alati i metode potrebne za dobivanje poboljšane učinkovitosti.

Grey box testiranje ispitivaču omogućuje određenu razinu pristupa i nekolicinu znanja o internom sustavu određene mete, odnosno ovo je tip pristupa koji se opisuje kao spoj prethodna dva. Provođenje napada putem pristupa sive kutije može se opisati kroz dva scenarija. Prvi scenariji temelji se na prijetnji iznutra, odnosno ispitivaču se daju podaci potrebni za dobivanje prava korisnika nižih razina koje zatim ispitivač iskorištava za provođenje daljnjih napada, dok drugi scenariji omogućuje ispitivaču da kao običan korisnik ili zaposlenik zadobije privilegije viših razina ili mogućnost prikupljanja podataka drugih korisnika. Prednost ovog tipa testiranja je preciznost opsega ispitivanja koja je u skladu s prioritetima. Koristi se prilikom testiranja novih sustava ili sustava s posebno osjetljivim informacijama. Iako postupci testiranja sive kutije uvelike smanjuju potencijalne rizike, kao i kod prethodna dva tipa pristupa ispitivači ne mogu jamčiti apsolutnu sigurnost sustava zbog brzog razvoja znanja i tehnika na globalnoj razini.

2.2.2. Vrste penetracijskog testiranja s obzirom na agresivnost

Ispitivanju se može pristupiti raznim intenzitetom i stupnjevima agresivnosti (engl. *Aggressiveness*) koji određuju dubinu testa. Testiranje se s obzirom na agresivnost može podijeliti u četiri metrike koje čine: pasivan, oprezan, izračunat i agresivan penetracijski test. Najniži stupanj agresivnosti je pasivan (engl. *Passive*) način koji se odnosi na otkrivanje ranjivosti tijekom testiranja, ali se te ranjivosti ne iskorištavaju. Oprezan (engl. *Cautious)* način ima svrhu izvođenja testiranja s ciljem korištenja onih ranjivosti koje neće izazvati koliziju nad ciljanim sustavom, primjer takvog načina je korištenje poznatih zadanih lozinki ili pristup direktorijima na web poslužiteljima. Izračunat (engl. *Calculated*) način omogućuje korištenje sigurnosnih ranjivosti uključujući i one s mogućim negativnim posljedicama po sustav. Zahtjev koji se stavlja pred ispitivačima je potreba i pouzdanost korištenja ovog tipa agresivnosti. Primjer izračunatog načina uključuje brute force napade koji omogućuju automatsko otkrivanje lozinki te iskorištavanje ranjivosti buffer overflow metoda koji služi za iščitavanje podataka prelijevanjem međuspremnika u susjednu memoriju. Agresivan (engl. *Aggressive*) stupanj koristi se prilikom generiranja golemih količina mrežnog prometa te iskorištavanja svih vrsta potencijalnih ranjivosti. Primjer agresivnog pristupa su DoS napadi koji omogućuju uskraćivanje usluga korištenjem velikih količina prometa kako bi se zasitili poslužitelji te kao i u izračunatom stupnju agresivnosti putem buffer overflow napada [12].

2.2.3. Vrste penetracijskog testiranja s obzirom na perimetar izvođenja

Perimetrom se određuje opseg testiranja, odnosno nad kojim dijelovima sustava će se vršiti penetracijsko testiranje. Odabir perimetra ovisit će o vremenu te o broju izvođenja testiranja nad određenim sustavom jer ako je vrijeme ograničeno ili je više puta u kratkom periodu obavljeno potpuno testiranje, nema potrebe za testiranjem potpunog sustava. S obzirom na perimetar izvođenja, testiranja mogu biti usredotočena, ograničena ili potpuna. Usredotočen (engl. *Focused*) penetracijski test zahtijeva točno određeni dio sustava koji se najčešće odnosi na novo implementiran ili proširen podsustav, tada se podsustav odvaja od potpunog sustava i testira kao zasebna cjelina. Ograničen (engl. *Limited*) test odnosi se na eksterni ili interni dio sustav koji je neovisan o drugom, dok potpuni (engl. *Full*) zahtijeva testiranje kompletnog sustava koji uključuje baze podataka, aplikacije, podsustave za obradu i prijenos podataka, kritične veze, mrežnu infrastrukturu, pristupnu točku i druge zaštitne ciljeve organizacija. Perimetar se određuje pristankom svih stranaka uz dogovoreni način koji je definiran prije provođenja penetracijskog testiranja [13].

2.2.4. Vrste penetracijskog testiranja s obzirom na tehniku izvođenja

S obzirom na tehniku izvođenja, penetracijske testove možemo podijeliti u mrežne, fizičke i aplikacijske testove te socijalni inženjering. Svaka od ovih tehnika zahtjeva posebne metode i vrste napada kako bi se pronašle određene ranjivosti i zaštitio sustav. Prije izvođenja potrebno je odrediti ostale klasifikacije testiranja poput pristupa, agresivnosti i perimetra kako bi se odredile smjernice kojima će se ispitivač voditi prilikom testiranja odabranom tehnikom.

2.2.4.1. Mrežni penetracijski testovi

Mrežni penetracijski testovi konvencionalni su načini napada sustava koji simuliraju stvarne postupke neovlaštenih napadača, a koriste se za identificiranje ranjivosti nad mrežom, operativnim sustavom, mrežnim protokolima i uređajima. Većina današnjih mreža koristi TCP/IP protokol, zbog čega se ovaj tip testiranja još naziva i penetracijskim testom baziranim na IP-u (engl. *Internet Protocol*). Za izvršavanje uspješnog testa potrebna su četiri koraka. Prvi korak sastoji se od prikupljanja informacija koji ovisi o pristupu. Drugi korak sastoji se od izviđanja i mapiranja kojima se pronalaze ranjivosti pomoću skeniranja mreže na način da se identificira put za narušavanje mreže. Treći korak je prodiranje u mrežu i sastoji se od provedbi napada nad ranjivostima otvorenih portova koristeći tehnički pristup u kojima se koriste napadi poput: DoS, *Brute force*, *buffer overflow* ili *SQL injection* napada, dok se ljudski pristup dobiva putem socijalnog inženjeringa. Posljednji korak sastoji se od izvještavanja i određivanja preporuka za sanaciju pronađenih prijetnji.

Mrežni testovi najčešće uključuju:

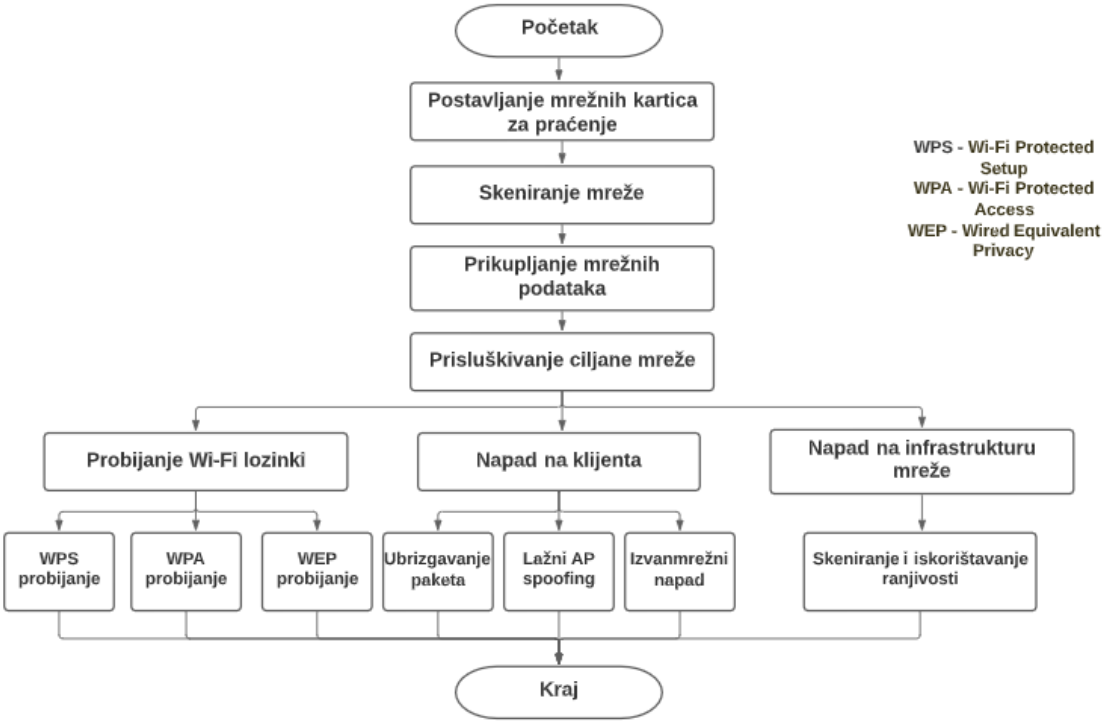
- Zaobilaženje vatrozida (engl. *Firewall*)
- Testiranje i pronalaženje ranjivosti u konfiguraciji mrežnih uređaja
- IPS/IDS izbjegavanje
- Skeniranje i testiranje otvorenih portova
- SSH (engl. *Secure Shell*) napadi
- Analizu proxy servera
- Ranjivosti mreže i protokola

Vatrozid je softverski ili hardverski uređaj koji na temelju određenog skupa pravila za kontrolu pristupa (engl. *Access Control List* - ACL) pregledava i filtrira dolazni i odlazni promet. Vatrozidi se obično postavljaju u demilitariziranoj zoni (engl*. Demilitarized zone* - DMZ) te služe kao jedan od prvih linija za zaštitu od kibernetičkih napada i zbog toga otežava napade ispitivačima penetracijskog testiranja i zlonamjernim napadačima. Testiranje prodora vatrozida je proces kojim se obavlja lociranje, istraživanje i prodiranje u određeni vatrozid kako bi se dosegao dio mreže nekog sustava. Prodiranje u vatrozid smatra se jednim od ključnih dijelova testiranja, jer predstavlja prvu liniju obrane od vanjskih napada [14].

Bežične LAN mreže se brzo razvijaju unutar različitih sredina, a razlog tome je jednostavnost upotrebe bez uspostavljanja žičanih veza. Međutim, način na koji bežične tehnologije funkcioniraju i raširenost korištenja dovelo je do stvaranje velikog broja sigurnosnih ranjivosti. Shodno tome, većina korisnika bežičnih mreža ne razumiju u potpunosti način rada što otvara mogućnosti za nepravilno korištenje koji potencijalni napadači mogu iskoristiti. Nekoliko je metoda zaštite bežičnih mreža, ali niti korištenje svih pristupa sigurnosti ne znači da je postignuta potrebna razina zaštite. Sigurnosni rizici koji nastaju u slučaju neadekvatno zaštićenih bežičnih mreža su sljedeće: neovlašteni pristup internoj mreži putem bežične mreže, presretanje osjetljivih (i nešifriranih) informacija prenesenih putem bežične

mreže, krađa identiteta i namjerna zloupotreba identiteta legitimnog vlasnika, napadi na bežičnu mrežu ili uređaj koji koristi bežičnu mrežu za pokretanje anonimnih napada na druge mreže, stvaranje lažnih pristupnih točaka i slično. Postoji nekoliko metoda zaštite bežičnih mreža među kojima su zaštita od neovlaštenog pristupa korištenjem filtriranja statičke IP adrese, MAC filtriranja adrese i skrivanja SSID-a, kao i ograničavanje raspona signala ili usmjeravanja pristupne točke signala. Također, jedan od najvažnijih metoda zaštite predstavljaju enkripcijske metode zaštite bežičnih mreža. Razvijanjem WEP (engl. *Wired Equivalent Privacy*) algoritma kao metode zaštite bežičnih mreža temeljenih na standardu 802.11 pružila se određena sigurnost, međutim kasnijim istraživanjima dokazano je da je WEP standard ranjiv, te ne predstavlja adekvatno rješenje za budućnost. Zatim su razvijeni standardi WPA (engl. *Wi-Fi Protected Access*) i WPA2 standardi koji uz adekvatnu konfiguraciju predstavljaju učinkovitu, gotovo nezaobilaznu sigurnosnu zaštitu bežičnog prometa [15].

Ispitivanje bežičnih mreža bitno je zbog velike raširenosti tehnologije koju koristi većina tvrtki, a u posljednje vrijeme porastom IoT uređaja pametne kuće također mogu biti mete napada neovlaštenih napadača. Slika 2 prikazuje glavne tehničke metode prilikom testiranja prodora bežičnih mreža koje uključuju postavljanje mrežnih kartica za praćenje, skeniranje mreža, prikupljanje podataka i praćenje ciljane mreže. Zatim se koriste neki od mrežnih napada poput razbijanje Wi-Fi lozinki, ubrizgavanje paketa, izvanmrežni napadi ili stvaranjem lažne pristupne točke (engl. *Fake Access Point spoofing*) kako bi se pristupilo sustavu i kritičnim podacima [16].



Slika 2. *Proces izvođenja mrežnih penetracijskih testova*
Izvor: [16]

Prilikom mrežnog penetracijskog testiranja koriste se alati poput: Aircrack (koristi se za razbijanje WEP i WPA lozinki), Reaver (koristi se prilikom bruteforce napada za dobivanje WPA/WPA2 lozinki), Infernal Twin (koristi se za stvaranje lažne bežične pristupne točke za dohvaćanje mrežnih podataka), Wireshark (koristi se za analiziranje mrežnih paketa) i drugi. Osim Wi-Fi mreža, penetracijska testiranja uključuju i testiranja telefonske mreže, mobilnih komunikacijskih mreža, bluetooth tehnologija i drugih.

Anomalija u mrežnom prometu predstavlja podatke koji odstupaju od prethodno definiranog normalnog ponašanja određene promatrane pojave. Promatrano sa aspekta informacijsko komunikacijskog sustava, anomalije u komunikaciji, odnosno mrežnom prometu često su posljedica nezakonitih mrežnih aktivnosti u sustavu, a anomalije mrežnog prometa imaju potencijal negativnog utjecaja na rad informacijsko komunikacijskog sustava ili usluga. Jedan od čestih uzroka anomalija u mrežnom prometu je DDoS napad. Tijekom posljednja dva desetljeća brojne su studije usmjerene na razvoj metoda, modela i sustava koji mogu detektirati DDoS promet u stvarnom vremenu. Ipak, broj napada i količina DDoS napada konstantno se povećava, zbog čega su potrebna daljnja istraživanja u području otkrivanja sigurnosnih prijetnji ove vrste. Kibernetički napadi poput DDoS napada i dalje su česti te mogu imati brojne negativne učinke na predviđene performanse informacijsko komunikacijskih sustava i dostupnost njihovih usluga [17].

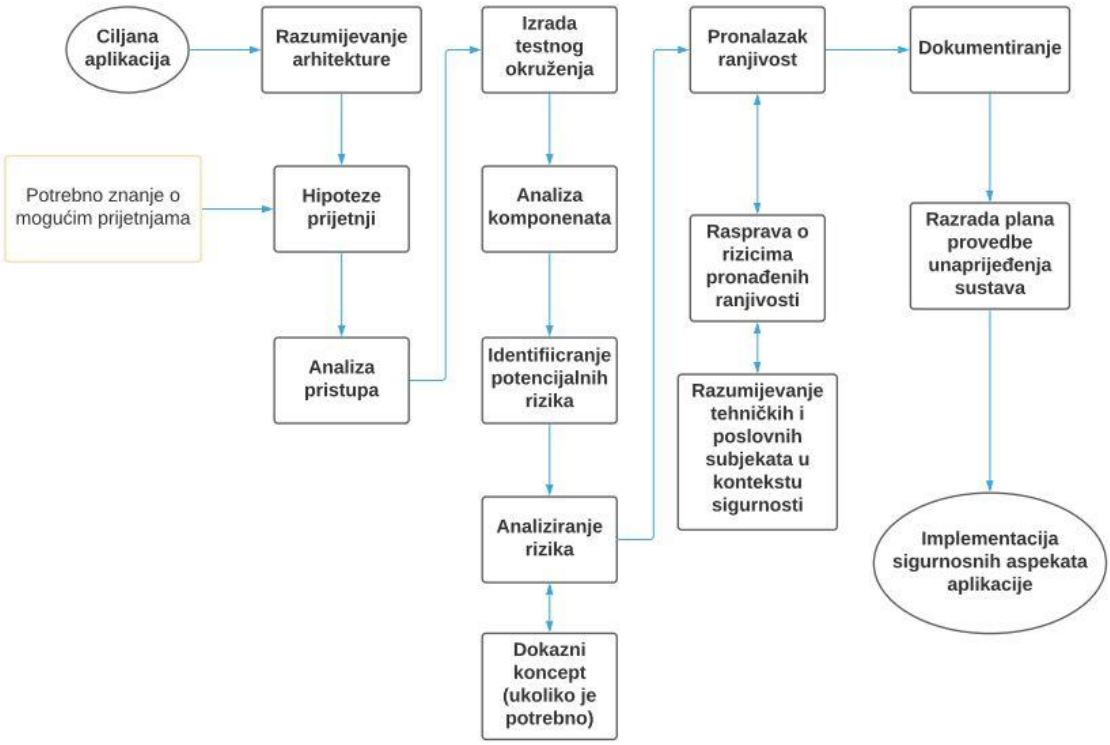
2.2.4.2. Fizički penetracijski testovi

Fizički penetracijski testovi imaju važnu ulogu u zaštiti sigurnosnih aspekata prilikom provođenja sigurnosne politike i povećanja svijesti zaposlenika na odgovarajuću razinu. Za razliku od izvođenja penetracijskih testova digitalnim putem, fizički testovi zahtijevaju fizički ulazak na ograničena mjesta i izravnu komunikaciju sa zaposlenicima unutar tvrtke. Ono što definira ovaj tip penetracijskog testiranja je izravna interakcija sa zaposlenicima koja sadržava etičke, pravne i sigurnosne implikacije. Faza testiranja započinje prikupljanjem informacija o ciljanoj imovini, njenom vlasniku i određenim zaposlenicima te predlaganjem scenarija napada. Scenariji napada prije izvršavanja mora biti dostavljen ovlaštenoj osobi koja mora taj scenariji potvrditi. Nakon pripreme, započinje faza izvršavanja koja može sadržavati neke od sljedećih zadataka: postavljanje predmeta u sigurnosno ograničenim prostorima, snimanje, prisluškivanje dijelova sustava, određivanje lokacija i slično. Tijekom napada ispitivači se mogu suočiti s nepredvidivim situacijama zbog uključenog ljudskog elementa te se od njih može zahtijevati improvizacija, što u digitalnim dijelovima testiranja nije slučaj [18].

2.2.4.3. Penetracijsko testiranje aplikacija

Penetracijsko testiranje aplikacija može se opisati kroz model prikazan na slici 3. Potrebno je razumijevanje arhitekture kako bi se mogle postaviti hipoteze vezane uz prijetnje, zatim je potrebna detaljna analiza komponenata i mogućih rizika koje prijete sustavu. Nakon otkrivanja ranjivosti, započinje rasprava te se obrađuju mogući štetni učinci pojedinih ranjivosti nad tehničkim i poslovnim subjektima, potrebno je napomenuti kako su faze pronalaženja ranjivosti i rasprava u iteraciji, sve dok se ne pronađu sve ranjivosti sustava. Dokumentiranje i

razrada plana ovisit će o mogućnostima implementacije sigurnosnih aspekata. Složenost i vrijeme trajanja testiranja ovisi o veličini i vrsti aplikacije.



Slika 3. *Provođenje penetracijskog testiranja aplikacija*
Izvor: [19]

Prilikom testiranja ispitivači pokušavaju steći neovlaštenu kontrolu nad sustavom ili dijelom sustava na način da koriste različite metode i alate. Ispitivači bi trebali osigurati izvođenje testova s korisnicima različitih uloga jer se sustav može različito ponašati u odnosu na korisnike drugačijih ovlasti. Bitno je slijediti kriterije i definirani postupak izvođenja napada i otkrivanja ranjivosti. Neki od alata za korištenje testiranja aplikacija su: Vega, Burp Suite, NetSparker, ZAP, Acunetix i slični. Dokumentiranje je faza koja nudi određene sanacije pronađenih ranjivosti, a ukoliko je potrebno može doći do ponovnog testiranja ranjivosti kao potvrdu točnosti rezultata. Čišćenje je postupak kojim se vraćaju sve promjene u nazad kako bi sustav bio u istom stanju kao i prije početka testiranja. U dokumentaciji osim ponuđenih rješenja nalaze se i podaci poput troškova, razine ranjivosti, vjerojatnosti za određenim događajima te koji su utjecaji pojedinih ranjivosti na poslovanje [19].

2.2.4.4. Penetracijsko testiranje tehnikom socijalnog inženjeringa

Penetracijsko testiranje socijalnog inženjeringa je skup napada kojim se pokušava prevariti zaposlenike tvrtke kako bi se utvrdila sigurnosna razina organizacije na tu vrstu prijetnje. Napadi korištenjem socijalnog inženjeringa mogući su u raznim oblicima, a najučestaliji su pecanje (engl. *Phishing)*, pecanje podataka putem telefonskog poziva (engl. *Vishing*), pecanje podataka putem SMS poruka (engl. *Smishing*), lažno predstavljanje (engl. *Impersonation*), USB podmetanje (engl. *USB Drops*) i zaobilaženje (engl. *Tailgating*). Dezinformiranjem zaposlenika ili poslovnih partnera može doći do prikupljanja osjetljivih podataka koji omogućuju pristup

inače zaštićenim resursima. Kako bi se sustavi osigurali od napada potrebno je podići svijest zaposlenika vezanu uz informacijsku sigurnost [20].

- Phishing je metoda koja pokušava navesti korisnika na otvaranje zlonamjernog koda kako bi zarazio njegovo računalo.
- Vishing je sličan phising-u, ali se odvija putem telefonskog poziva i nastoji prevariti korisnike za dobivanje osjetljivih podataka.
- Smishing je također sličan phishing-u, ali se odvija putem SMS poruke.
- Impersonation je metoda lažnog predstavljanja kojom se korisniku predstavlja pod lažnim identitetom u svrhu dobivanja pristupa osjetljivim podacima.
- USB drops je metoda koja koristi USB sa zlonamjernim kodovima koji omogućuje stražnji ulaz u sustav prilikom njegovog priključenja u računalo.
- Tailgating je metoda zaobilaženja mjera fizičke sigurnosti. Primjer korištenja je na mjestima gdje tvrtke zahtijevaju skeniranje kartica za pristup sustavu.

Postoje četiri glavna koraka prilikom provođenja penetracijskog testa socijalnim inženjeringom koja uključuju planiranje i opseg, identifikaciju vektora napada, pokušaj penetracije sustava i izvještavanje. Planiranje i opseg je korak kojim će se obuhvatiti način izvođenja testa, a određuje osobe nad kojima će se provesti napadi. Vektor napada identificira vrstu napada socijalnog inženjeringa, a najčešće su to phishing napadi. Treći korak je pokušaj provođenja vektora napada s ciljem dobivanja željenih informacija. Dokumentacija daje osvrt na izvođenje i analizu napada te prijedloge podizanja svijesti osoba kako bi se izbjegli potencijalni napadi socijalnog inženjeringa.

2.2.5. Vrste penetracijskog testiranja s obzirom na početnu točku izvođenja

Penetracijsko testiranje s obzirom na početnu točku izvođenja razlikujemo ovisno s kojeg mjesta se odvija test. Napadi započinju s mjesta početne točke (engl. *Starting point*) prilikom koje ispitivač spaja svoje računalo na mrežu. Početne točke uglavnom su napadi nad vatrozidima, web poslužiteljima, pristupnim točkama i bežičnim mrežama. Web poslužitelji česti su razlog napada zbog različitih funkcija koje nudi, a ujedno time rezultiraju i velikim opsegom ranjivosti. Ovisno o početnoj točki razlikujemo unutarnji penetracijski test (engl. *Inside penetration test*) koji započinje testiranjem u internoj mreži bez da zaobilazi sigurnosne mehanizme te zbog toga može jasnije procijeniti pogrešku prilikom konfiguracije mreže te definirati razloge ranjivosti nad kritičnim podacima ili nad osobom koja ima pristup internoj mreži. Drugi tip početne točke je vanjski penetracijski test (engl. *Outside penetration test*) koji omogućuje otkrivanje i evaluaciju potencijalnih rizika pristupa eksternom i internom dijelu mreže.

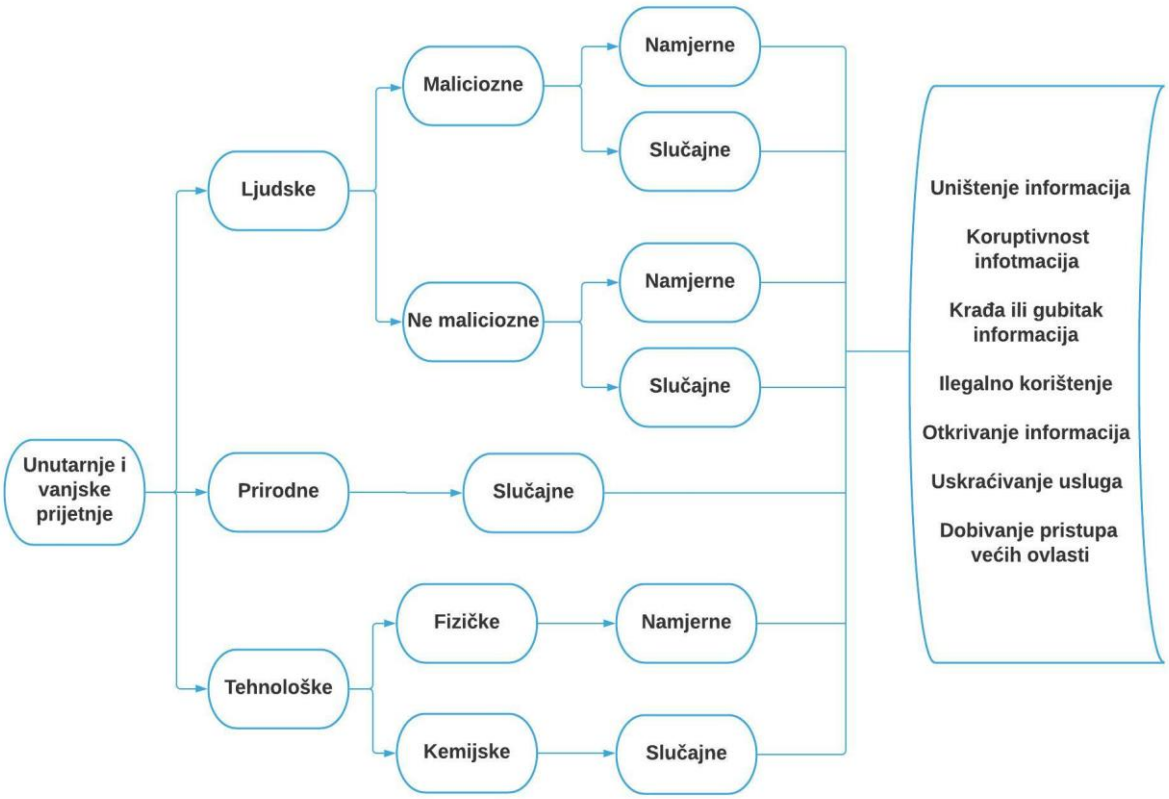
3. Utvrđivanje sigurnosnih aspekata informacijskog sustava

Sustavi su opširan pojam koji se opisuju kao skup elemenata povezanih u svrsishodnu cjelinu, a cilj sustava je određene ulazne forme pretvoriti u željeni izlazni rezultat djelovanjem različitih procesa. Informacijski sustav je sastavni dio svakog ciljno orijentiranog sustava. Osnovna svrha informacijskog sustava je permanentna opskrba potrebnim informacijama svih razina upravljanja i odlučivanja u danom tehnološkom, odnosno organizacijskom obliku. Sastoji se od elemenata prikupljanja (ulaz), obrade, pohrane (izlaz) i povratne veze. Sigurnost informacijskog sustava opisuje se kao stanje povjerljivosti, cjelovitosti i raspoloživosti podataka, koje se postižu primjenom propisanih mjera i standarda informacijske sigurnosti te organizacijskom podrškom za poslove planiranja, provedbe, provjere i dorade mjera [21]. Informacijska sigurnost postiže se raznim mehanizmima s ciljem zaštite informacija od raznih prijetnji u svrhu stjecanja kvalitetnijeg poslovnog napretka i stvaranja sigurnijih poslovnih prilika i planova. Takva vrsta zaštite postiže se primjenom CIA trokuta, koji opisuje zaštitu sustava kroz faze povjerljivosti, integriteta i dostupnosti.

Povjerljivost (engl. *Confidentiality*) se predstavlja kao prevencija prilikom sprječavanja neovlaštenih otkrivanja informacija. Postoje dvije metode zaštite povjerljivosti podataka, fizička koja dozvoljava samo ovlaštenim osobama pristup podacima dok je pristup ostalima onemogućen te logička metoda zaštite koja koristi mrežne sigurnosne protokole, različite tipove autentifikacije i kriptiranje podataka. Integritet (engl. *Integrity*) podataka jest jamstvo da se skup informacija prenosi s početne točke na krajnju bez da se izmjenjuje sadržaj poruke. Integritet se osigurava uz pomoć usluga vatrozida, komunikacijskih sigurnosti te korištenjem metoda hashiranja. Dostupnost (engl. *Availability*) je jamstvo da će informacija biti pravovremeno, bez prekida i na pravo mjesto poslana potrošaču bez obzira na mjesto slanja. Za pružanje dostupnosti potrebno je da infrastruktura, sigurnosne kontrole, cloud usluge i komunikacijske usluge ispravno funkcioniraju u svakom trenutku rada sustava. Dostupnost se osigurava tolerancijom na greške, autentifikacijom i mrežnom sigurnošću [22].

3.1. Klasifikacija prijetnji informacijskih sustava

Do ranjivosti sustava dolazi zbog slabosti pojedinih podsustava koje potencijalni napadači koriste. Manje organizacije zaštitu sustava mogu planirati na jednom ili dva kriterija klasifikacija prijetnji. Na primjer, manje organizacije ne moraju raditi potpunu zaštitu internog sustava zbog manjeg broja zaposlenika, dok veće organizacije moraju odrediti sve izvore prijetnji kako bi u najvećoj mjeri zaštitili svoj sustav. Klasifikacija omogućuje organizaciji da prepozna prijetnje koje utječu na njihovu imovinu kako bi ju mogli unaprijed zaštititi. Takav pristup može omogućiti izgradnju sigurnosnih sustava ili pomoći pri stvaranju hibridnih tehnika zaštita analizirajući kombinaciju različitih klasifikacija prijetnji [23]. Slika 4 prikazuje klasifikaciju prijetnji za organizacije te raščlanjuje prijetnje na namjerne i slučajne, maliciozne i ne maliciozne te ljudske, prirodne i tehnološke.



***Slika 4**. Klasifikacija prijetnji*
Izvor: [24]

Sigurnosne prijetnje mogu biti prouzročene iz vanjskog ili unutarnjeg dijela sustava, a bitan je segment jer određuje izvor nastanka prijetnje. Interne prijetnje javljaju se kada napadači ili prijetnja ima odobren pristup mreži ili fizičkom dijelu organizacije. Segment interne prijetnje često je rezultat zaposlenika unutar organizacija. Vanjske prijetnje mogu nastati od pojedinca ili organizacija koje nemaju ovlašteni pristup mreži, a napade izvršavaju fizičkim upadima, kibernetičkim napadima kroz žične i bežične mreže te upadom u sustav putem napada na vanjske suradnike ili treće strane [24].

Aktere provođenja prijetnji čine ljudi, prirodne katastrofe i tehnološki utjecaji. Prema istraživanjima objavljenim u knjizi *Computer Crime: A Crimefighter's Handbook* najveći postotak rizika za organizacije uzrokovan je ljudskim faktorom u internom dijelu mreže, koji se najčešće događaju zbog nedovoljne pažnje i edukacije zaposlenika. Kako bi se spriječila ova prijetnja potrebno je uvesti mjere educiranja kojima se smanjuje vjerojatnost njihovih pogrešaka [25]. Prirodne prijetnje uzrokovane su prirodnim katastrofama poput potresa, poplava, požara, vjetra i oluje te prijetnje poput ratova i terorističkih napada. Tehnološke prijetnje uzrokovane su fizikalnim ili kemijskim procesom nad opremom i ostalom imovinom sustava. Fizikalni procesi uključuju korištenje fizičkih sredstava za ulazak u ograničena područja te krađu ili oštećenje hardvera ili softvera. Kemijskim prijetnjama smatraju se procesi nad hardverskim i softverskim elementima koji rezultiraju neispravnošću ili kvarom, takav tip tehnološke prijetnje nastaje najčešće slučajnim, ali i neispravnim rukovođenjem zaposlenika organizacija.

Neovlašteni napadači prilikom napada na sustav imaju cilj i motiv, a s obzirom na motiv napadi mogu biti maliciozni ili ne maliciozni. Maliciozni napadi imaju za cilj napasti sustav uz pomoć malicioznih kodova dok ne maliciozni napadi nastaju uz prisustvo slabe zaštite i kontrole koji omogućuju pogreške i iskorištavanje ranjivosti, najčešće od strane zaposlenika. Svaka od klasificiranih prijetnji može se kategorizirati u namjerne ili slučajne događaje. Rizici koji mogu nastati djelovanjem jedne ili više klasificiranih prijetnji su:

- Uništavanje informacija (engl. *Destruction of information*) ili određenih komponenata koje rezultiraju prekidu rada sustava.
- Koruptivnost informacija (engl. *Corruption of* information) - smatra se neovlaštenom izmjenom datoteka ili informacija na glavnom računalu ili serveru, a najčešće je uzrokovano malicioznim kodom poput Trojanskog konja.
- Otkrivanje informacija (engl. *Disclosure of information*) - definira širenje informacija neovlaštenim osobama.
- Krađa usluge (engl. *Theft of service*) - neovlašteno korištenje računalnih ili mrežnih usluga.
- Uskraćivanje usluge (engl. *Denial of Service*) - namjerno blokiranje računalnih ili mrežnih sustava slanjem velikih količina zahtjeva sve dok se sustav ne preoptereti.
- Povećanje privilegija (engl. *Privilege escalation*) - način zadobivanja većih ovlasti s potencijalnim širenjem na cijeli sustav.
- Ilegalno korištenje (engl. *Illegal* usage) - korištenje standardnih funkcija, ali na nedozvoljen način koji se kasnije definiraju kao zlonamjerni napadi.

3.2. Metode zaštite informacijskog sustava

Zaštita informacijskog sustava započela je korištenjem računala u organizacijama koje su koristile komunikaciju putem interneta, pod pojmom zaštita podrazumijeva se provođenje mjera u svrhu osiguranja rada informacijskog sustava. Veliki broj organizacija svoju efikasnost i funkcioniranje temelje na informacijsko komunikacijskom sustavu i stoga veliku važnost daju njenoj zaštiti. Poslovni informacijski sustav čini infrastruktura, svi fizički uređaji i oprema kojom upravlja čovjek s ciljem postizanja što kvalitetnijih ciljeva tvrtke. Također, prilikom odabira metoda zaštita sustava potrebno je odrediti kompleksnost tog sustava. Stoga, informacijske sustave možemo podijeliti u jednostavne, složene i inteligentne. Sigurnosna politika određuje se dokumentom koji sadrži izjavu ili očitovanje odgovornih osoba čime daju uvjerenje, ciljeve i razloge te načine dostizanja željenih postignuća u području informacijske sigurnosti.

Dokument sigurnosne politike predstavlja hijerarhijski strukturirani skup propisa koji sadrži smjernice, naputke, različite razine standarda i procedure. Sukladno Zakonu o informacijskoj sigurnosti, središnje tijelo nadležno za pitanja informacijske sigurnosti je Ured vijeća za nacionalnu sigurnost dok je za tehnička pitanja informacijske sigurnosti nadležan Zavod za sigurnost informacijskih sustava. CERT (engl. *Computer Emergency Response Team*) i Zavod za sigurnost informacijskih sustava surađuju na prevenciji i zaštiti od računalnih ugroza sigurnosti informacijskih sustava te sudjeluju u izradi preporuka i normi u Republici Hrvatskoj iz područja sigurnosti informacijskih sustava [26].

Metode zaštite informacijskog sustava mogu se podijeliti u tri kategorije:

- Metoda fizičke zaštite
- Metoda programske zaštite
- Metoda organizacijske zaštite

Hrvatski zakon o informacijskoj sigurnosti fizičku zaštitu definira kao područje informacijske sigurnosti prilikom kojeg se utvrđuju mjere i standardi informacijske sigurnosti za zaštitu objekta, prostora i uređaja u kojem se nalaze klasificirani podaci. Fizička zaštita informacijskih sustava kontrolira se brojnim fizičkim sigurnosnim sustavima kao što su: sustavi za nadzor, alarmi, sustavi za detekciju požara i suzbijanja plinova, sustavi za kontrolu pristupa, sustavi za detekciju upada, adekvatna oprema zaštitara, sustavi za zaštitu okoline i slično. Kontrola pristupa provodi se sigurnosnom provjerom koja se zakonom definira kao područje informacijske sigurnosti u okviru kojeg se utvrđuju mjere i standardi sigurnosti, a primjenjuju se na osobe koje imaju pristup klasificiranim podacima [27]. Osim kontrole pristupa bitnu stavku u fizičkoj zaštiti ima i zaštita opreme koja je ključan segment u radu informacijskih sustava. Pojedini dio opreme razmatra se zasebno te mu se ovisno o namjeni, karakteristikama i vrijednosti određuje posebna razina fizičke zaštite. Sve vrste fizičkih sigurnosnih sustava su važne za održavanje sigurnosti, međutim potrebna je kombinacija više vrsta zaštita kako bi upad u sustav bio što teži [28].

Metoda programske zaštite sastoji se od pružanja zaštite softverskom dijelu sustava. U većini organizacija najvažnija je zaštita podataka koja u slučaju neadekvatne zaštite može stvoriti nepovratnu financijsku i neodrživu štetu tvrtkama. Zakon [27] opisuje sigurnost podataka kao područje informacijske sigurnosti za koje se utvrđuju mjere i standardi informacijske sigurnosti koje se primjenjuju kao opće zaštitne mjere za prevenciju, otkrivanje i otklanjanje štete od gubitka ili neovlaštenog otkrivanja klasificiranih i neklasificiranih podataka. Zaštita podataka u informacijskim sustavima obavlja se putem kriptiranja, postavljanja autentifikacije prilikom pristupa kritičnim podacima ili zaštitom podataka korištenjem sigurnosne kopije (engl. *Backup*). Mrežni dio sustava zaštićuje se konfiguracijom vatrozida, mrežnih uređaja te korištenjem VPN (engl. *Virtual Private Network*) usluga. Ostale metode programske zaštite su korištenje antivirusnih sustava za zaštitu od malicioznih programa, redovito ažuriranje softvera, određivanje razine pristupa i ovlasti, dvokoračna autentifikacija za pristup sustavu i slično.

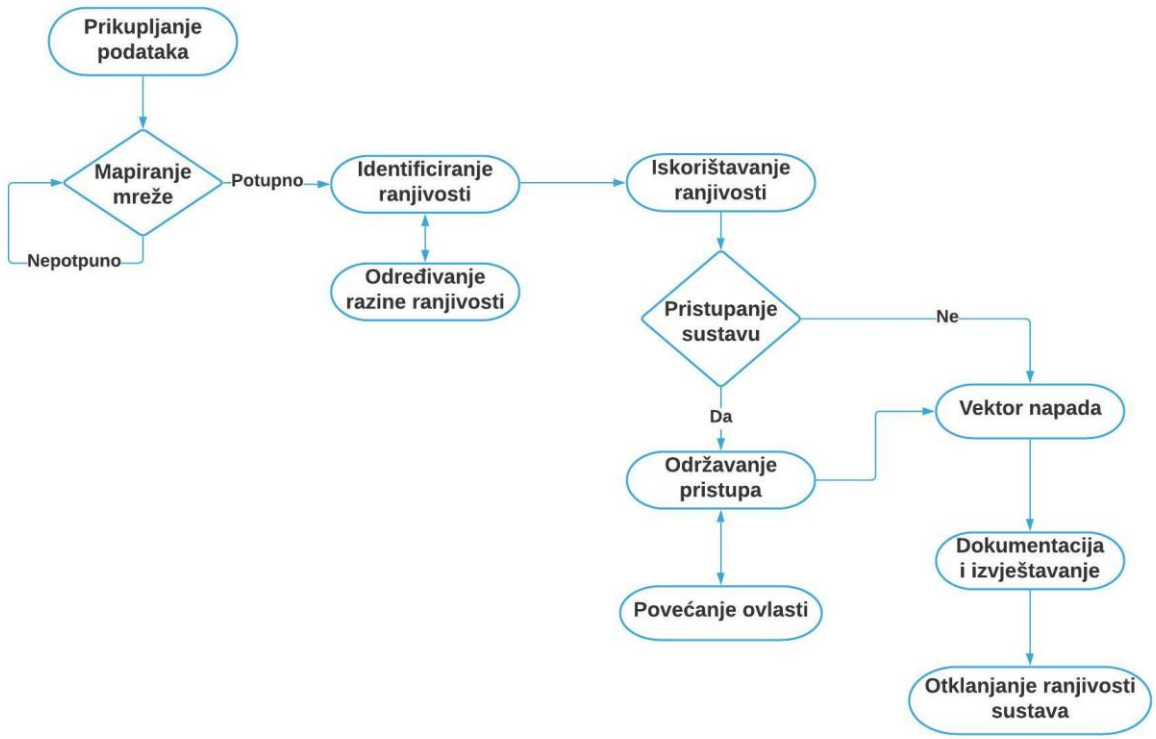
Metode zaštite organizacija svode se na infrastrukturnu zaštitu. Kako bi se adekvatno zaštitila infrastruktura informacijskog sustava potrebno je educirati zaposlenike, preventivno održavati sigurnost sustava te provoditi kontrolu zaštite trećih strana, *outsourcing*, sigurnosnu poslovnu suradnju i ostale sigurnosne mehanizme. Sigurnosna poslovna suradnja definira se kroz zakon [27] kao područje informacijske sigurnosti u kojem se primjenjuju propisane mjere i standardi informacijske sigurnosti za provedbu natječaja ili ugovora s klasificiranom dokumentacijom koji obvezuju pravne i fizičke osobe. Outsourcing se koristi za održavanje sigurnosti sustava uz pomoć vanjskih tvrtki ili pojedinca, primjer toga je i penetracijsko testiranje. Potrebno je ugovorno odrediti ovlasti sudionika trećih strana kako bi se pravno osigurali od nastanka šteta, također potrebno je trećim stranama odrediti ispravnu kontrolu pristupa.

3.3. Faze procesa penetracijskog testiranja

Faze procesa određuju tijek izvođenja penetracijskog testiranja kako bi ono bilo funkcionalno i kvalitetnije izvršeno. Metodologija penetracijskog testiranja određuje fazu procesa, a njena primjena koristi se kao smjernice prilikom testiranja. Prema dogovoru s organizacijom određuju se tipovi testiranja kao i pristupi koji utječu na faze koje su definirane za potrebe testiranje. Proces penetracijskog testiranja podijeljen je u 5 faza:

- Prikupljanje informacija
- Mapiranje mreže
- Identificiranje ranjivosti
- Iskorištavanje ranjivosti/penetracija sustava
- Dokumentacija i izvještavanje

Prije izvođenje testiranja po fazama potrebno je upoznati i steći dovoljno saznanja o okruženju koje se napada. Kako bi mogao pristupiti testiranju ispitivač mora razmotriti pitanja kao što su: koji pristup testiranju primijeniti, na koji način će se izvoditi, koji uvjeti su potrebni za uspješno izvršavanje, vremenski rok u kojem je potrebno uspješno odraditi testiranje, koje će poslovne ciljeve postići organizacija nakon krajnjeg rezultata, ima li ispitivač dovoljno znanja i informacija potrebnih za uspješno ispitivanje te ima li dovoljno znanja o tehnologiji i njenim osnovnim funkcionalnostima. Slika 5 prikazuje dijagram aktivnosti kojim se opisuju faze procesa penetracijskog testiranja.



Slika 5. *Proces izvođenja penetracijskog testiranja*

Nakon što se postigao dogovor oko ispitivanja sustava neke organizacije i područje koje će se tim testiranjem zahvatiti potrebno je prikupiti podatke koji će biti ključan proces pri daljnjem testiranju, a nakon toga se odrađuje mapiranje mreže. Kada je obavljeno potpuno mapiranje mreže i prikupljeni su svi podaci za nastavak testiranja potrebno je identificirati ranjivosti i odrediti im razine prijetnji. Nadalje, potrebno je iskoristiti ranjivosti koje mogu dati pristup sustavu nakon čega će biti potrebno zadržati pristup i ukoliko je moguće povećati ovlasti te potencijalno proširiti pristup na cijeli sustav. Ako faza iskorištavanja ranjivosti nije imala cilj pristupanja sustavu (primjer DoS napada) tada se automatski provode vektori napada nad sustavom. Cijeli proces potrebno je dokumentirati te zatim otkloniti identificirane ranjivosti.

3.3.1. Prikupljanje podataka

Prikupljanje informacija je prva faza procesa koja uključuje prikupljanje što većeg opsega informacija s ciljem ostvarivanja kvalitetnijeg testiranja. Za uspješnu eksploataciju sustava potrebno je prikupiti iz raznih izvora informacije vezane uz okruženje sustava koji se testira. Količina prikupljenih podataka ovisit će o vremenu i kreativnosti ispitivača, a većina informacija bit će prikupljena od javno dostupnih informacija korištenjem interneta. Ova faza može se podijeliti na dvije glavne kategorije:

- Aktivno prikupljanje informacija
- Pasivno prikupljanje informacija

Aktivno prikupljanje informacija podrazumijeva korištenje alata i automatiziranih metoda za pronalaženje korisnih informacija koji će proširiti opseg pronalaska ranjivosti. Također, potrebno je napomenuti kako metode aktivnog prikupljanja nisu uvijek preporučene metode iz razloga što mogu biti lako zapažene od strane sustava za detekciju upada (*Intrusion Detection System* – IDS), sustava za sprječavanje upada (*Intrusion Preventions System* – IPS) ili vatrozida koji će zatim generirati log zapise njihove prisutnosti. Aktivno prikupljanje informacija koristi alate i metode poput *nslookup* alata koji služi za dobivanje naziva domene, pojedinostima IP adrese i traženjem DNS zapisa. Nadalje, koristi se *whois* metoda koja sadrži bazu podataka gotovo svih web stranica na internetu. Najčešće informacije koje se mogu dohvatiti korištenjem ove metode su podaci o vlasniku pojedinih web stranica, a najčešće su to adrese pošte.

Pasivno prikupljanje informacija temelji se na pretraživanju interneta s ciljem pronalaska podataka iskoristivih za pronalaženje potencijalne ranjivosti sustava. Sustav se ne napada direktno kao što je to u aktivnom prikupljanju što može biti prednost iz razloga što ova metoda ne zahtijeva interakciju s ciljnim sustavom i neće generirati log zapise. Tragove i potrebne informacije pronaći će se pretraživanjem izvora usko vezanih uz organizaciju koje se mogu naći na forumima, reklamama, portalima, oglasnim pločama, interesnim grupama, blogovima, člancima, društvenim mrežama, organizacijama koje sadrže javne informacije i slično. Moguće je pretraživati zaposlenike organizacija, a najčešće je to putem raznih društvenih mreža koja korištenjem phishing ili neke druge metode socijalnog inženjeringa može pridobiti relevantne podatke potrebne za nastavak testiranja [29].

3.3.2. Mapiranje mreže

U odnosu na prethodnu fazu, mapiranje mreže prikuplja opširnije podatke o mreži korištenjem alata za skeniranje. Neki od alata koji se koriste prilikom skeniranja mreže su Nmap, Zenmap, p0f i Masscan, a u nastavku rada detaljno će se analizirati Nmap alat te prikazati njegova primjena prilikom penetracijskog testiranja virtualnog okruženja. Cilj ove faze je prikupiti informacije o vrsti i verziji operativnog sustava, portovima, servisima, usmjerivačima i vatrozidima mrežnog sustava. Ova faza temelji se na većem broju aktivnih testova nad ciljanim sustav i stoga je potrebno oprezno koristiti alate kako se sustav ne bi preopteretio. Otkrivanje operativnog sustava korištenjem Nmap alata odrađuje se na način da šalje veliki broj TCP (engl. *Transmission Control Protocol*) i UDP (engl. *User Datagram Protocol*) paketa na udaljeno računalo i istražuje svaki bit vraćenih paketa, zatim Nmap uspoređuje rezultate s bazom podataka u kojoj se nalazi preko 1500 operativnih sustava te ispisuje njegove detalje ukoliko ga je uspješno pronašao. Skeniranje portova je proces provjere priključaka na način da šalje zahtjeve na svaki port i traži odgovor na poslani zahtjev te je jedan je od najučestalijih procesa koji se obavljaju tijekom penetracijskog testiranja. Portovi se mogu opisati kroz 3 glavna stanja: otvoren, zatvoren i filtriran. Otvoreni portovi prihvaćaju TCP i UDP pakete i oni su primaran cilj tijekom skeniranja jer ostavljaju mogućnost za napad. Zatvoreni portovi primaju zahtjev, ali nema usluge "slušanja", ta vrsta porta je i dalje dostupna te može biti korisna jer prikazuje da je host na mreži i koristi IP adresu. Kod filtriranih portova poslan je paket, ali se ne može utvrditi je li port otvoren ili zatvoren te postoji mogućnost da je paket blokiran od strane vatrozida. Način na koji će se izvršiti napad nad određenim otvorenim portom ovisit će o servisu porta. Tablica 1 prikazuje najčešće otvorene TCP i UDP portove te njihove najučestalije servise [30].

Tablica 1. *Najučestaliji TCP i UDP portovi*

TCP
PORT 80 - HTTP (Hypertext Transfer Protocol)
Port 23 - Telnet
Port 443 - HTTPS (Hypertext Transfer Protocol Secure)
Port 21 - FTP (File Transfer Protocol)
Port 22 - SSH (Secure Shell)
Port 25 - SMTP (Simple Mail Transfer Protocol)
Port 110 - POP3 (Post Office Protocol)
Port 445 - Microsoft-DS
Port 139 - Netbios-ssn (NETBIOS Session Service)
Port 143 - IMAP (Internet Message Access Protocol)
UDP
Port 631 - IPP (Internet Printing Protocol)
Port 161 - SNMP (Simple Network Management Protocol)
Port 137 - NETBIOS-NS (NETBIOS Name Service)
Port 123 - NTP(Network Time Protocol)
Port 138 - NETBIOS-DGM (NETBIOS Datagram Service)
Port 1434 - MS-SQL-DS (Microsoft SQL server)

Izvor: [30]

3.3.3. Identificiranje ranjivosti

Ova faza je temeljni proces koji predstavlja ranjivosti sustava te je potrebno da se svaki pristup, funkcija i proces odradi na pravilan način korak po korak jednak predloženom modelu metodologije penetracijskog testiranja određenog prije samog testiranja. Proces koristi prikupljene podatke iz prethodnih faza, određuje njihovu vrijednost i značaj za nastavak testiranja te na temelju razina ranjivosti određuje scenariji izvršenja napada. Identificiranje ranjivosti obavlja se korištenjem automatiziranih alata ili ručnom provedbom ili kombinacijom tih dviju metoda. Nedostatak ručnog analiziranja je duže vrijeme potrebno za obavljanje identifikacijske faze. Uspješnost odrađivanja faze ovisi o vrsti alata koji će se koristiti za analiziranje kao i količina znanja ispitivača, koji osim znanja mora imati i najnovije informacije o ranjivostima i njenim specifikacijama. Proces identifikacije sadrži dva glavna postupka, analiziranje koda (engl. *Code Analysis* – CA) i analiziranje ranjivosti (engl. *Vulnerability Analysis* – VA). Analiza koda koristi se za analizu izvornog koda kako bi se pronašli sigurnosni nedostaci te kao pomoć ispitivaču da koristi sigurnosne relevantne dijelove koda kako bi učinkovitije pronašao nedostatke. Analiza ranjivosti teži sigurnosnim rizicima koji predstavljaju softverske ranjivosti kako bi se odredio broj ranjivosti u softveru koji je u fazi razvoja ili je već implementiran. Uspješnost pronalaženja ukupne ranjivosti (engl. *Total Analysis* – TA) određuje se sljedećom jednadžbom: TA = CA + VA [31].

Korištenjem automatiziranih alata omogućuje se brzo i kvalitetno skeniranje ciljnih sustava za pronalazak ranjivosti. Neki od alata za skeniranje ranjivosti su: Nessus, Acunetix, BurpSuite, Nikto i drugi. Nessus je jedan od najpoznatijih alata za skeniranje ranjivosti s više od dva milijuna preuzimanja diljem svijeta te pruža opsežnu pokrivenost obuhvaćajući više od 59 000 CVE-ova (engl. *Common Vulnerabilities and Exposures*). Detaljan opis i analiza Nessus alata objašnjena je u sljedećem poglavlju, dok je njegova primjena prikazana u poglavlju praktičnog dijela izvođenja penetracijskog testiranja nad virtualnim okruženjem. Neke od najpoznatijih mrežnih stranica s velikom bazom podataka ranjivosti su Exploit database, WhiteSource i VulDB, a nude detalje vezane o pojedinim ranjivostima kao što su CVE, vrstu, način iskorištavanja, proces zaštite, datum otkrivanja ranjivosti i slično.

3.3.4. Iskorištavanje ranjivosti

Tijekom faze iskorištavanja ranjivosti pokušava se omogućiti proces pristupa sustavu korištenjem ranjivosti koje su prikupljene u prethodnoj fazi. Faza pristupanja smatra se najzahtjevnijom fazom, međutim potrebno je prilikom izvođenja faze procijeniti utjecaj rizika zbog mogućnosti stvaranja oštećenja nad ciljanim sustavom. Faza pristupanja zahtjeva veliki broj korištenja alata, skripti, enkodiranja podataka (engl. *Payload*) i vještina programiranja ispitivača u raznim programskim jezicima sa svrhom dobivanja pristupa, eskalacijom privilegija, DoS napada ili stjecanjem kontrole nad ograničenim dijelovima sustava. Ovisno o vrsti sustava postoje alati koji omogućuju eksploataciju sustava, a neki od njih su: Metasploit, BurpSuite, OWASP Zap, SQL Map, John The Ripper, Frida i slično. Nakon uspješnog prodora u sustav ispitivači ne smiju zanemariti mogućnost pristupa sustavu na više načina. Cilj ispitivača je pronaći i iskoristiti sve moguće načine kojima će zadobiti kontrolu nad sustavom, stoga je potrebno proći kroz sve ranjivosti prikupljenih u prethodnoj fazi.

Nakon dobivanja pristupa potrebno je povećati ovlasti koje ispitivač ima nad sustavom poput prava za korištenje baza podataka, pristup datotekama od značaja, kontrole sustava, napad na lokalne mreže sustava i slično. Povećanje privilegija (engl. *Privilege Escalation*) definira se kao napad koji omogućuje dobivanje neautoriziranog pristupa s većim pravima nastao iskorištavanjem programske pogreške, pogrešne konfiguracije, socijalnim inženjeringom ili neodgovarajuće kontrole pristupa. Vektori napada koji omogućuju izvođenje zlonamjernih aktivnosti sa ciljem povećanja ovlasti mogu biti: napada na lozinke, nagađanja lozinke, metodom surfanja preko ramena (engl. *Shoulder surfing*), napadom pomoću rječnika (engl. *Dictionary attack*), raspršivanje lozinkom, brute force napadima, tehnikom UAC (engl. *User Account Control*) zaobilaženja, enumeracijom identiteta ili zlonamjernim programom. U tablici 2 prikazano je kako su Windows operativni sustavi češće izloženi vektoru socijalnog inženjeringa, a razlog tome je veća prisutnost Windows sustava kod krajnjih korisnika, dok kod Unix i Linux operativnih sustava do povećanja privilegija najčešće dolazi zbog pogrešne konfiguracije. Niti jedan sustav nije imun na povećanje ovlasti, međutim moguće je dodatno zaštiti sustav kako bi se ta metoda napada otežala napadačima [32].

Tablica 2. *Vjerojatnost vektora napada nad operativnim sustavima*

Operativni sutav	Eksplotacija kredencija	Ranjivosti	Pogrešna konfiguracija	Maliciozni kod	Socijalni inženjering
Windows	Visoka	Visoka	Srednja	Visoka	Visoka
macOS	Visoka	Srednja	Niska	Srednja	Visoka
Unix	Visoka	Srednja	Srednja	Niska	Niska
Linux	Visoka	Visoka	Visoka	Srednja	Visoka
Android	Visoka	Visoka	Srednja	Visoka	Visoka
IoT	Visoka	Srednja	Visoka	Niska	Niska

Izvor: [32]

Daljnje popisivanje objekata (engl. *Enumerating further*) je faza koja pokušava zadobiti što veći broj korisnih informacija nakon eskaliranja privilegija. Djelovanje ove faze nastoji prodrijeti što dublje u mrežu, a sastoji se od sljedećih koraka:

- Pokušava otkriti kriptirane zaporke sustava koje nisu povezane na mrežu na način da se kopiraju datoteke /etc/passwd i /etc/shadow (u slučaju Linux sustava).
- Otkrivanje zaporki korištenjem sniffer alata te analiziranje prometa koji se najčešće obavlja alatima kao što je WireShark.
- Prikupljanje kolačića (engl. *cookies*) te iskorištavanje njihovih ranjivosti.
- Prikupljanje adresa elektroničke pošte.
- Identifikacije ruta i mreža.
- Mapiranjem.

Nadalje, prilikom penetracije sustava potrebno je uzeti u obzir i tehniku zakretanja (engl. *Pivoting*) koja koristi pristup kojeg ima napadač za pokušaj prodiranja kroz čitavu mrežu, neovisno o sigurnosnim aspektima tih podsustava. Vrlo često su komunikacije udaljenih organizacija ili njenih zaposlenika zaštićeni autentifikacijom ili enkripcijom korištenjem VPN sustava, međutim to ne jamči pouzdanost krajnjih točaka u komunikaciji što ostavlja prostor napadačima da kompromitiraju udaljene sustave. Prilikom napada sustava i ostvarivanja udaljenog pristupa potrebno je održavanje stalne i trajne prisutnosti nad kompromitiranim sustavom s ciljem održavanja anonimnosti. Održavanje pristupa i skrivanje tragova omogućuje se korištenjem metoda i alata kao što su skrivanje kanala, korištenje stražnjih vrata (engl. *Backdoor*) za pristup računalnim sustavima, korištenje rootkit alata, prikrivanjem tragova i datoteka te čišćenje log zapisnika i zaobilaženje antivirusnih alata. Potrebno je naglasiti kako korištenje backdoor i rootkit alata se rijetko koristi prilikom izvođenja penetracijskog testiranja iz razloga što može stvoriti rizike i omogućiti stvarnim napadačima da iskoriste prisutnost backdoor ili rootkit alata za izvedbu stvarnih napada. Nakon korištenja svih koraka testiranja i prodora do krajnje mogućnosti sustava potrebno je ciljani sustav vratiti u početno stanje koje je bilo prije penetracijskog testiranja [33].

3.3.5. Dokumentacija i izvještavanje

Nakon što je penetracijsko testiranje završeno potrebno je klijentu predati izvješće sa krajnjim rezultatima i saznanjima vezanim uz testiranje. Rezultati moraju sadržavati opis i objašnjenje svih usluga korištenih u radu, provedenu metodologiju, prikupljena saznanja o sustavu te rješenja i preporuke s ciljem daljnje zaštite sustava. Faza dokumentacije je proces koji se odvija tijekom cijelog tijeka izvođenja testiranja iako se predstavlja kao krajnji proces testiranja. Potrebno je predati precizno izvješće koje ne ostavlja otvorena pitanja ili određene nejasnoće. Izvještaj sadrži osjetljive informacije vezane o sustavu organizacije, stoga je potrebno pružiti visoku povjerljivost u skladu s politikom organizacije. Nakon što je završena faza dokumentacije potrebno je na siguran način dostaviti izvještaj klijentu korištenjem enkripcije gdje je ključ poznat samo klijentu. Ispitivač ima etičku obvezu zadržati saznanja o organizaciji za sebe, dostaviti ih u pravo vrijeme i na pravo mjesto te stvoriti broj kopija izvješća jednak broju klijenata kojima se ti isti dokumenti trebaju dostaviti. Izvještaj se sastoji od izvršnog sažetka (engl. *Executive Summary*), detaljnog izvješća (engl. *Detailed Report*) i sirovih rezultata (engl. *Raw Output*) koji su hijerarhijski podijeljeni s obzirom na količinu informacija koju sadrži. Svaki dio izvješća gleda se kao zasebna cjelina, a skup svih cjelina daje potpuni izvještaj. Izvršni sažetak daje kratki pregled najbitnijih stavki testiranja čija je forma strukturirana da bude razumljiva osobama koja nemaju previše znanja o kibernetičkoj sigurnosti. Ako su prilikom testiranja pronađene ranjivosti potrebno ih je detaljno objasniti te obrazložiti kakve bi negativne utjecaje ranjivost mogla stvoriti organizaciji. Detaljan izvještaj sadrži detaljnu analizu rezultata i tehničke pojedinosti. Ovaj tip izvještaja nudi određena rješenja pojedinih ranjivosti, a predviđen je za razumijevanje kod stručnih osoba kibernetičke sigurnosti. Sastoji se od tehničkih detalja i sirovih rezultata te analize izlaznih vrijednosti dobivenih korištenjem pojedinih alata unutar penetracijskog testiranja. Ukoliko se određene informacije ne nalaze u izvješću, potrebno je napraviti odvojeni dokument koji će biti posebno dostavljen uz jamstvo o povjerljivosti tog dokumenta [34].

3.4. Standardi penetracijskog testiranja

Različite metodologije penetracijskog testiranja donose različite rezultate, ovisno o alatima, metodama i standardima. Ažurirani standardi i metodologije penetracijskog testiranja pružaju organizacijama mogućnosti zaštite sustava i ispravljanje ranjivosti u skladu s kibernetičkom sigurnošću. Određeni standardizirani pristupi odredili su metodologiju penetracijskog testiranja vlastitim viđenjem najoptimalnijeg pristupa. Postoji nekoliko standarda penetracijskog testiranja, a u ovom radu obradit će se OSSTMM (engl. *Open Source Security Testing Methodology Manual*), OWASP (engl. *Open Web Application Security Project*) i NIST (engl. *National Institute of Standards and Technology*).

3.4.1. Open Source Security Testing Methodology Manual

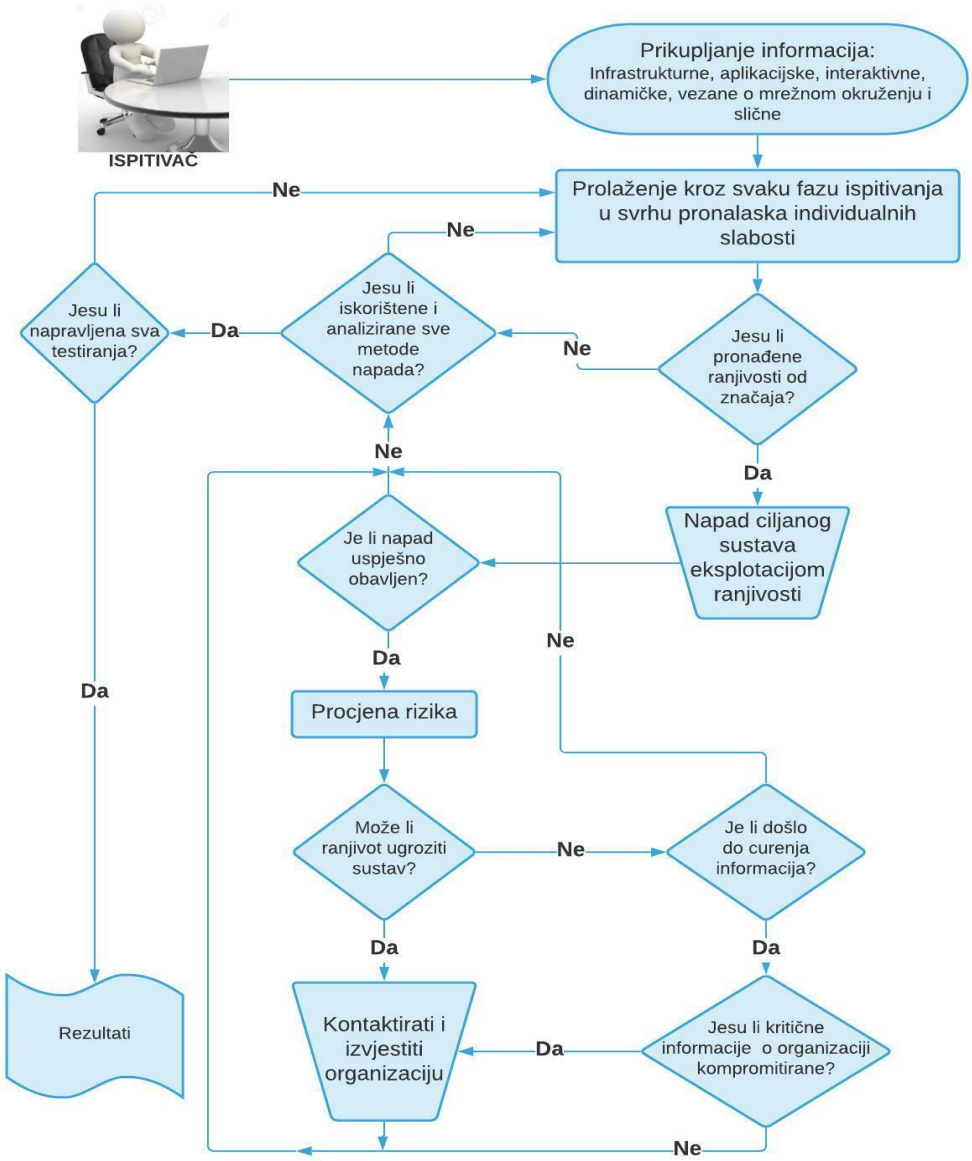
OSSTMM je globalno dostupan priručnik za provođenje penetracijskog testiranja u svrhu postizanja temeljitog sigurnosnog testa te pružanja znanstvene metodologije za točnu karakterizaciju operativne sigurnosti (engl. *Operation Security* - OpSec) dobivanjem kvantitativnih rezultata na dosljedan i pouzdan način. Rezultat testiranja vođen OSSTMM metodologijom daje sigurnost na operativnoj razini bez pretpostavki već s jasnim dokazima. Ova metodologija sastavljena je 2000. godine, a do danas je nekoliko puta unaprijeđena s ciljem postizanja što točnijih testiranja. Razni ispitivači doprinose idejama s ciljem obavljanja preciznijih radnji i učinkovitijeg sigurnosnog ispitivanja. OSSTMM je napisan kao dokument o sigurnosnom istraživanju i dizajniran je za provjeru činjenične sigurnosti i prezentaciju metrike na profesionalnoj razini. Dokument nudi smjernice koje će, ako se pravilno slijede, omogućiti ispitivaču da izvrši certificiranu reviziju priručnika. Smjernice koje osiguravaju ispravno provođenje zahtijevaju temeljitu provedbu testa te obuhvaćanje svih potrebnih aspekata, zatim provođenje mora biti u skladu sa zakonom, a rezultati moraju biti mjerljivi, dosljedni i ponovljivi te moraju sadržavati činjenice izvedene iz samih penetracijskih testova. OSSTMM metodologija se može primijeniti na svim organizacijama neovisno o njenoj veličini, tehnologiji ili zaštiti [35].

OSSTMM priručnik temelji se na sljedećim poglavljima:

- Metrika operativne sigurnosti
- Analiza povjerenja
- Tijek rada
- Testiranje ljudske sigurnosti
- Testiranje fizičke sigurnosti
- Testiranje sigurnosti bežičnih mreža
- Testiranje telekomunikacijske sigurnosti
- Testiranje sigurnosti mrežnog prometa
- Propisi usklađenosti
- Izvještavanje pomoću STAR-a (engl. *Security Test Audit Report*)

3.4.2. The Open Web Application Security Project

OWASP je projekt osmišljen za razmjenu znanja i razvoja softvera otvorenog koda sa svrhom razumijevanja sigurnosti web aplikacija. OWASP *Foundation* je neprofitna organizacija koja upravlja projektom OWASP, osnovanog 2000. godine izdavanjem priručnika za detaljnu analizu i provedbu zaštite web aplikacija. Osim što nudi metodologiju izvođenja penetracijskog testiranja objašnjava i način na koji se trebaju zaštiti okruženja kako bi većim dijelom web aplikacije bile sigurne od potencijalnih napada. OWASP metodologija pruža identificiranje ranjivosti i logičkih nedostataka koji proizlaze iz nesigurnih razvojnih praksi. Priručnik pruža opsežne smjernice za svaku od 66 metoda testiranja penetracije, omogućavajući ispitivačima da identificiraju ranjivosti unutar širokog spektra funkcionalnosti koje se nalaze u današnjim modernim aplikacijama [36].



Slika 6. *OWASP metodologija penetracijskog testiranja*
Izvor: [36]

Slika 6 prikazuje dijagram aktivnosti OWASP metodologije penetracijskog testiranja koja započinje prikupljanjem statičkih, dinamičkih, infrastrukturnih, interaktivnih informacija te informacija o mreži sustava. Sljedeći korak metodologije je pronaći sve tipove ranjivosti kojima je izložen sustav, ovaj proces se smatra završenim tek onda kada su detektirane sve potencijalne prijetnje. Proces ispitivanja ranjivosti sastoji se od određivanja opsega, načela, tehnika i okvira testiranja te od životnog ciklusa razvoja softvera. Fazu ispitivanja također čini opis ranjivosti poput: XSS, pogrešne konfiguracije, korištenja zastarjelih komponenata, monitoring metoda, nepotpune kontrole pristupa, otkrivanja osjetljivih podataka i drugih. Nadalje slijedi faza napada koja u slučaju uspješnog provođenja prelazi na fazu procjene rizika, a u protivnom proces se ponovno vraća na fazu pronalazaka ranjivosti. Organizaciju je potrebno kontaktirati u situacijama kada identificirana ranjivost može ugroziti sustav ili omogućiti curenje kritičnih informacija, dok se u ostalim slučajevima informacije daju nakon testiranja kroz izvještaj. Nakon što su napravljena sva testiranja potrebno je dokumentirati rezultate i proslijediti ih organizaciji [37].

3.4.3. National Institute of Standards and Technology

Nacionalni institut znanosti i tehnologije (NIST) je američka agencija za razvoj standarda i pravila za informacijsku sigurnost, gdje kroz skup smjernica pomaže organizacijama da izgrade i poboljšaju vlastite sustave u okviru kibernetičke sigurnosti. NIST dokument prikazuje se kao vodič kroz osnovne tehničke aspekte provođenja procjene informacijske sigurnosti. Koristi razne tipove ispitivanja i metoda koje bi organizacije mogle koristiti kao dio procjene potencijalnih utjecaja na sigurnosni sustav i elemente izvan njih. Kroz priručnik je definirana NIST-ova metodologija penetracijskog testiranja koja se sastoji od planiranja, otkrivanja, napada i izvještavanja. U fazi planiranja ne dolazi do stvarnog testiranja, već se određuju pravila dokumentiranja postupka, kao i faze izvođenja koja je temelj za uspješan penetracijski test. Faza otkrivanja sastoji se od dva dijela, gdje se kroz prvi dio obuhvaća postupak prikupljanja informacija (analiza IP adrese, podaci o zaposlenicima, sistemske informacije te informacije vezane o aplikaciji i uslugama) i skeniranje mrežnih portova u svrhu određivanja potencijalnih ciljeva. Drugi dio faze otkrivanja je analiza ranjivosti koja uključuje usporedbu usluga uz pomoć baza podataka te ručno identificiranje ranjivosti. Faza napada provodi se dobivanjem pristupa sustavu, zatim eskalacije privilegija koja omogućuje pretraživanje sustava uz pomoć prikupljenih podataka iz prethodnih faza. Posljednji proces u fazi napada je korištenje alata u svrhu dobivanja dodatnih informacija i zadržavanje pristupa. Faza izvještavanja odvija se sukladno s ostale tri faze penetracije, obično se vode pisani dnevnici i periodično se izvještavaju administratori sustava. Na kraju ispitivanja izrađuje se izvješće koje opisuje identificirane ranjivosti, ocjenu rizika i nudi smjernice kako bi se ublažile otkrivene slabosti [38].

4. Analiza i funkcionalnosti alata

U nastavku rada provedeno je testiranje nad virtualnim okruženjem korištenjem metodologije penetracijskog testiranja. Ispitivanje prati faze izvođenja testiranja uz korištenje automatiziranih alata i metoda napadom nad ranjivim sustavom. Za potrebe izvedbe testiranja potrebno je postaviti virtualnu mašinu unutar virtualnog okruženja, a za potrebe ispitivanja u ovom radu korišten je Oracle VM box konfiguriran sa svrhom postavljanja Kali Linux platforme predviđene za provođenje penetracijskog testiranja. Analiza nekih od konvencionalnih alata potrebnih za testiranje virtualnog okruženja su:

- Nessus
- Nmap
- SQLmap
- DirbBuster
- Netcat
- JoomScan
- John The Ripper

4.1. Kali Linux

Kali Linux je jedan od svjetski najsnažnijih i najpopularnijih platformi za korištenje penetracijskog testiranja, temelji se na Debian Linux distribuciji te sadrži više od 300 aplikacija i alata predviđenim za provođenje penetracijskog testiranja. Kali je poznat kao *BackTrack* koji je nastao spajanjem IWHAX, WHOPPIX i Auditor distribucija, a usmjeren je na sigurnosne stručnjake i IT administratore, ali studente i obične korisnike sa svrhom postizanja znanja o kibernetičkoj sigurnosti. Projekt Kali Linux započeo je 2012. godine, nakon što je *Offensive Security* odlučio kako je potreban sustav za informacijsku sigurnost s kvalitetnom, stabilnom i širokom dostupnošću. Kali se može koristiti na raznim uređajima kao što su osobna računala, poslužiteljskim sustavima, radnim stanicama, ARM procesorima te mobilnim uređajima (NetHunter). Kali Linux sadrži veliki broj alata predviđenih za različitu upotrebu i aktivnosti, a s obzirom na svrhu, alati se mogu podijeliti u sljedeće kategorije:

• Alati za prikupljanje informacija služe za dohvaćanje podataka ciljane mreže i njene infrastrukture, identificiranje računala, njihovih operativnih sustava i usluga koje pokreću te identificiranje potencijalno osjetljivih dijelova informacijskog sustava.

• Alati za analizu ranjivosti koriste se kod lokalnih ili udaljenih sustav s ciljem pronalazaka poznatih ranjivosti ili neispravnih konfiguracija. Koriste se skeneri za pronalaženje ranjivosti koji sadrže velike baze podataka s potencijalnim ranjivostima.

• Alati za analizu web aplikacija koriste se prilikom prepoznavanja pogrešnih konfiguracija i sigurnosnih slabosti aplikacija. Velika je potreba za ovom kategorijom alata iz razloga što su web aplikacije javno dostupne i zbog toga su idealna meta za neovlaštene napadače.

• Alati za napade nad bazama podataka najčešće se koriste u svrhu SQL injekcija i pristupanju podacima. Zbog sve većeg broja incidenata nad bazama potrebno je omogućiti kontrolirani i zaštićeni pristup uz pružanje cjelovitosti, povjerljivosti i integriteta podataka.

• Alati za napade nad lozinkama koriste se kao vektor napada nad sustavima autentifikacije s ciljem pronalazaka lozinki. Postoji velik broj alata za ove svrhe s različitim mogućnostima probijanja lozinki poput brute force napada, napadom pomoću rječnika, raspršivanje lozinkom i slično.

• Alati za napade bežičnih mreža koriste se u pristupu samoj mreži te pružanju informacija o strukturi i prometu koji prolazi kroz mrežu, izvještavajući kasnije o napadima. Ovu kategoriju čine i alati predviđeni za napade nad usmjerivačima i drugim mrežnim komponentama.

• Alati za provođenje obrnutog inženjeringa koriste se u ofenzivnim aktivnostima, gdje se primarnom metodom smatra identifikacija ranjivosti i iskorištavanja sustava, dok se kod obrambenih metoda koriste za analizu zlonamjernog softvera koji se koristi u ciljanim napadima.

• Alati za eksploataciju ranjivosti omogućuju preuzimanje kontrole nad udaljenim računalom ili sustavom. Također, alati se mogu koristiti za daljnje napade povećanja privilegija, bilo lokalno na kompromitiranom uređaju, ili na drugim uređajima dostupnim u lokalnoj mreži. Ova kategorija sadrži niz alata i uslužnih programa koji pojednostavljuju proces pisanja vlastitih skripti. Osim alata za preuzimanje kontrole ovu kategorizaciju čine i alati potrebni za održavanje razine pristupa.

• Sniffing i spoofing alati koriste se za dobivanje pristupa podacima u mreži. Kali Linux nudi alate koji omogućuju lažno predstavljanje legitimnog korisnika, kao i alate za presretanje komunikacije koji omogućuju analiziranje podataka koji prolaze mrežom.

• Alati za provođenje digitalne forenzike koriste se u svrhu utvrđivanja izvedbe napada i kako odgovoriti na njih. Forenzički alati na Kali platformi omogućuju rješavanje osnovnih zadataka poput prikupljanja mrežnog prometa, ali i onih kompleksnih kao što su rekonstrukcija i analiza napada.

• Alati za izvješćivanje pomažu prilikom procesa dokumentacije zaključnih rezultata penetracijskog testiranja. Svrha ovih alata je prikupiti sve informacije, odrediti odnose među njima i sastaviti ih u razne izvještaje.

• Alati socijalnog inženjeringa koriste se kao vektori napada s ciljem dobivanja informacija od strane drugih osoba. Alati služe za generiranje napada socijalnog inženjeringa poput phishing napada. Veliki broj kibernetičkih napada odrađuje se putem socijalnog inženjeringa, stoga je potrebno analizirati i koristiti ove alate u svrhu povećanja sigurnosti informacijskih sustava [39].

4.2. Nessus

Nessus je skener mrežne sigurnosti koji koristi zasebne dodatke (engl. *Plug-in*) koje služe za rukovanje provjerama ranjivosti. Glavne značajke ovog programskog alata su identifikacija slabosti mreža koje omogućuju napadaču pristup informacijama o sustavu računala u mreži i provjera dostupnih ažuriranja koja uklanjaju određene slabosti mreže. Nessus također testira sigurnost računala u mreži koristeći zadane i često korištene lozinke, provodi analizu slabosti mreže te provjeru mobilnih uređaja u mreži. Skeniranje je moguće izvršiti na jednom računalu, na spektru IP adresa ili nad cijelom podmrežom. Nessus alat dostupan je za Unix, Linux i Windows operativne sustave, što pomaže da postane sveobuhvatni alat te tako omogući skeniranje mješovitog okruženje jednim pristupom. Jedna od najatraktivnijih značajki Nessus alata je sustav otvorenog koda (engl. *Open source*) koji omogućuje korisnicima da svakodnevno doprinose razvoju ovog alata, što mu pomaže u ažuriranju. U roku od nekoliko dana od objavljivanja ranjivosti bit će dostavljeno nekoliko dodataka za saniranje ranjivosti. Nadalje, Nessus koristi Nmap za skeniranje portova koji je postao standard u sigurnosnoj industriji iz razloga što izuzetno brzo skenira portove. Nessus programski alat omogućuje korisniku izvođenje probabilističke analize s analitičkim modelima, vanjskim računalnim programima poput komercijalnih kodova. Nessus grafičko korisničko sučelje (engl. *Graphical User Interface* - GUI) visoko je konfigurirano i omogućuje prilagođavanje specifičnim aplikacijama te mogućnost komercijalnih ili interno razvijenih kodova koji se zatim lako integriraju u okvirima Nessus okruženja [40].

Nessus alat dizajniran je za automatizirana testiranja i otkrivanja poznatih sigurnosnih ranjivosti, a svoju veliku raširenost u primjeni zahvaljuje konstantnim nadograđivanjem sustava. Nadogradnja sustava ili rješenja novih ranjivosti primjenjuje se nakon što neka interesna skupina ili pojedinac otkriju slučajnim ili namjernim istraživanjima nove načine kršenja sigurnosti softverskog proizvoda koje se zatim objavljuju sigurnosnoj zajednici koja identificira i rješava problem te ga implementira unutar Nessus alata. Programeri ga zatim pregledavaju i dodaju na popis odobrenih dodataka. Za ranjivosti visokog rizika i visokog profila dodatak se često objavljuje istog dana kada su informacije o ranjivosti javno dostupne. Svaki dodatak je napisan za testiranje određene ranjivosti, a zapravo se koriste kako bi se iskoristile ranjivosti ili testirale poznate ranjive verzije softvera, također potrebna je česta nadogradnja dodataka kako bi alat ostao ažuran. Dodaci se mogu pisati na gotovo svakom jeziku, ali obično su napisani u NASL jeziku (engl. *Nessus Attack Scripting Language* - NASL). NASL je vlastiti jezik tvrtke Nessus, posebno dizajniran za pisanje testova ranjivosti. Nakon dovršetka skeniranja generira se izvješće u kojem je navedena svaka pronađena ranjivost. Ako je skeniranje bilo ispravno izvedeno izvješće će biti cjelovito i točno. Međutim, potrebno je od velike količine informacija interpretirati ranjivosti od značaja, koje se određuju stupnjem rizika, a lažne pozitivne podatke potrebno je ukloniti iz izvješća [41].

4.3. Nmap

Nmap (*Network Mapper*) je besplatan alat otvorenog koda za skeniranje ranjivosti i otkrivanje svojstva mreže. Mrežni administratori, IT stručnjaci i penetracijski ispitivači koriste Nmap za identifikaciju uređaja koji rade na određenim sustavima, otkrivanje dostupnih hostova i usluga koje nude, pronalaženje otvorenih portova i otkrivanje sigurnosnih rizika. Nmap se može koristiti za praćenje pojedinačnih hostova, ali i velikih mreža koje sadrže veliki broj uređaja i podmreža. Nmap je u konstantnom razvoju i izuzetno je fleksibilan, u osnovi je to alat za skeniranje portova koji prikuplja informacije slanjem sirovih paketa na sistemske portove. Sluša odgovore i utvrđuje jesu li portovi otvoreni, zatvoreni ili filtrirani. Ostali izrazi koji se koriste za skeniranje portova uključuju otkrivanje portova ili nabrajanje. Pakete koje šalje Nmap vraćaju se s raznim vrstama podataka, koji omogućuju identificiranje različitih mrežnih atributa dajući mrežni prikaz sustava. Različiti protokoli koriste različite vrste struktura paketa. Nmap koristi protokole transportnog sloja uključujući TCP, UDP i SCTP (engl. *Stream Control Transmission Protocol*), kao i protokole poput ICMP-a (engl. *Internet Control Message Protocol*) koji se koriste za slanje kontrolnih poruka. Zenmap je grafičko korisničko sučelje sigurnosnog alata Nmap koji osim skeniranja pruža veliki spektar funkcionalnosti [42].

Iako je osnova Nmap-ove funkcionalnosti skeniranje portova, ona omogućuje različite srodne mogućnosti, poput identificiranje uređaja na mreži uključujući poslužitelje, usmjerivače te njihovu fizičku povezanost. Nadalje omogućuje reviziju sigurnosti utvrđivanjem ranjivosti pojedinih sustava, otkrivanje operativnih sustava na uređajima u mreži te njihovu verziju softvera kao i razne vrste usluga. Nmap također koristi sustav skripti NSE (engl. *Nmap Scripting Engine*) koji omogućuje korisnicima da pišu i dijele jednostavne skripte za automatizaciju širokog spektra mrežnih zadataka koristeći programski jezik *Lua*. Korisnici se mogu osloniti na sve veći skup skripti distribuiranih s Nmap alatom ili napisati vlastite kako bi zadovoljili prilagođene potrebe. Kako bi se pojednostavio izbor skripti za izvođenje, svaka skripta postavljena je u jednu ili više kategorija. Trenutno definirane kategorije čine: *auth, broadcast, default.discovery, dos, exploit, external, fuzzer, intrusive, malware, safe, version* i *vuln.* Iako skeniranje portova nije protuzakonito potrebno je oprezno korištenje Nmap alata iz razloga što nudi snažne vektore napada koji se mogu vrlo lako predstaviti kao pokušaji napada nad sustavima [43]. Velika raširenost ovog alata koristi se kao podloga skeniranja i kod drugih alata te je jedan od najučestalijih alata prilikom penetracijskog testiranja legitimnih ispitivača, ali i neovlaštenih napadača prilikom prikupljanja podataka o ciljanoj mreži. Nmap se između ostalog koristi za provjeru ispravnosti rada vatrozida, ali i njegovo zaobilaženje te IDS sustava za detekciju neovlaštenog prometa koji prepoznaju skeniranje Nmap alatom. Uz pomoć Nmap alata ponekad je moguće izvesti skeniranje u potpunoj anonimnosti.

Tablica 3. *Prikaz najučestalijih sintaksi Nmap alata*

Vrsta skeniranja	Sintaksa	Primjer
TCP SYN	-sS	nmap -sS 127.0.01
TCP Connect	-sT	nmap -sT 127.0.01
Fin	-sF	nmap -sF 127.0.01
XMAS	-sX	nmap -sX 127.0.01
Null	-sN	nmap -sN 127.0.01
Detekcija verzije	-sV	nmap -sV 127.0.01
UDP	-sU	nmap -sU 127.0.01
IP protokol	-sO	nmap -sO 127.0.01
ACK	-sA	nmap -sA 127.0.01
Windows	-sW	nmap -sW 127.0.1
Lista	-sL	nmap -sL 127.0.1

-sS sintaksa jedna je od najčešćih prilikom skeniranja portova iz razloga što nudi brzo i nezamijećeno skeniranje jer ne uspostavlja TCP vezu. TCP SYN se predstavlja kao poluotvoreni pregled (engl. *Half open scanning*) jer se ne uspostavlja potpuna veza. Ukoliko je port otvoren računalo će odgovoriti sa SYN/ACK paketom, ako je zatvoren dati će odgovor RST (*reset*) paketom, a ukoliko je port filtriran odgovorit će se ICMP porukom.

-sT sintaksa koristi funkciju connect() umjesto SYN pregleda. To je ista funkcija koju pozivaju i druge aplikacije kada žele uspostaviti vezu. S obzirom na to da Nmap ima manji nadzor nad connect() funkcijom nego nad TCP paketima ovaj tip pregleda je nešto manje učinkovit nego SYN pregled. Ukoliko je otvoren port ova funkcija u potpunosti uspostavlja vezu, zbog toga proces skeniranja traje duže u odnosu na TCP SYN.

-sF, -sX, -sN pregledi koriste nedostatak u RFC standardu koji definira TCP protokol. Prilikom skeniranja sustava čija je implementacija TCP protokola u skladu s odgovarajućim RFC-om, potrebno je na svaki paket koji nema postavljenu SYN, RST ili ACK zastavicu u zaglavlju paketa odgovoriti RST paketom. Nmap iskorištava ovaj propust slanjem FIN, PSH i URG paketa.

-sU sintaksa predstavlja slanje UDP paketa koji sadrže samo zaglavlje. Ako Nmap zaprimi ICMP odgovor to označuje da je port zatvoren ili filtriran, ovisno o vrsti ICMP odgovora. Ukoliko port odgovori UDP paketom to znači da je otvoren. Ovaj način pregleda stanja portova je spor iz razloga što Nmap ponovno šalje pakete na iste portove kako bi se eliminirala mogućnost gubljenja paketa u mreži. Drugi problem stvaraju ICMP odgovori zatvorenih portova jer određeni sustavi ograničavaju količinu poslanih paketa.

-sV sintaksa može se koristiti za preciznije otkrivanje stanja UDP portova.

-sO sintaksa služi za pregled otvorenih protokola, a ne za pregled portova.

-sA je vrsta pregleda koja se razlikuje od ostalih u tome što ne otkriva otvorene portove, već služi isključivo za mapiranje vatrozida i za otkrivanje koji su portovi filtrirani. ACK pregled šalje samo ACK paket. Otvoreni i zatvoreni portovi kao odgovor šalju RST paket te ih Nmap označava kao nefiltriranim.

-sW pregled se razlikuje od -sA sintakse po tome što se mogu odrediti otvoreni i zatvoreni portovi u mreži. Window pregled koristi činjenicu da neki sustavi razlikuju otvorene i zatvorene portove prilikom slanja RST paketa. Na takvim sustavima TCP paket ima pozitivnu veličinu kad je port otvoren, dok je kod zatvorenih portova ta veličina jednaka nuli.

-sL otkriva svaku adresu na mreži bez potrebe slanja paketa ciljanoj mreži. Nmap također ovom sintaksom izvještava o ukupnom broju IP adresa na kraju skeniranja. Skeniranje liste koristi se prilikom detaljnih provjere IP adresa [44].

4.4. SQLmap

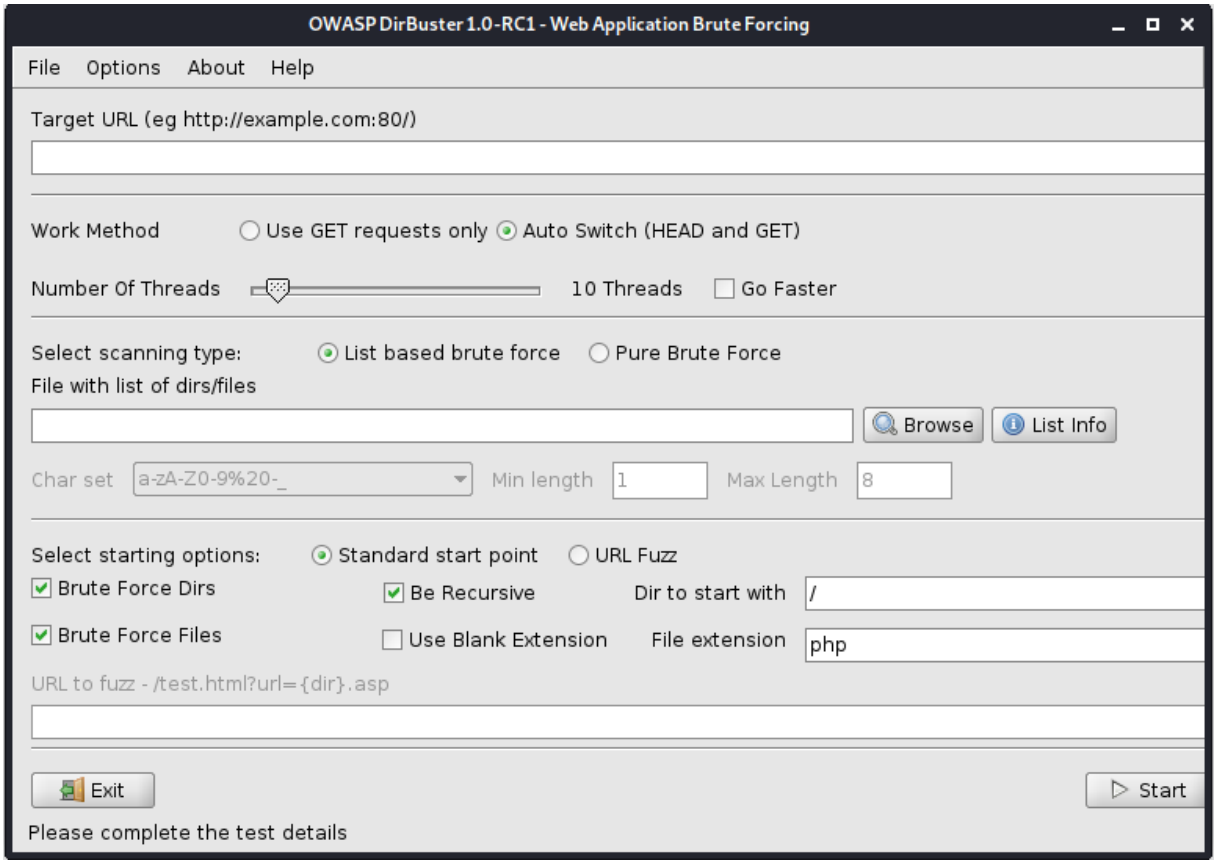
SQLmap je alat otvorenog koda s ciljem korištenja automatiziranih procesa otkrivanja, iskorištavanja i preuzimanja podataka s poslužitelja baza podataka. Sadrži veliki broj mehanizama za napade s ciljem otkrivanja ranjivosti baza podataka. SQLmap sadrži potpunu podršku za većinu sustava upravljanja bazama podataka (MySQL, PostgreSQL, Oracle, Microsoft SQL server, Microsoft Access, Sybase i drugi.). Alat sadrži sljedeće metode napada:

- Metode izvođenja svih tehnika SQL injekcija.
- Automatsko prepoznavanje formata raspršivanja lozinki i njihovo prodiranje uz pomoć napada temeljenih na rječniku.
- Nudi potpuno ispisivanje tablica baza podataka ili određenih unosa ovisno o potrebama ispitivanja. Također, nudi pretraživanje tablica kroz skup međusobno povezanih baza.
- Mogućnost preuzimanja i očitavanja datoteka s poslužitelja baze podataka koja se nalazi na datotečnom sustavu jednog od sustava za upravljanje bazama podataka koje su ranjive na ovaj alat.
- Uspostavljanje izvan pojasne TCP veze (engl. O*ut-of-band stateful* TCP) između napadača i poslužitelja baze podataka.
- Omogućuje dobivanje većih pristupa unutar sustava baza podataka.

SQLmap je dostupan svima te je omogućeno korisnicima da razvijaju alat sukladno novim prijetnjama i novim načinima napada. SQLi napadi jedni su od najvećih prijetnji sustava za održavanje povjerljivosti, integriteta i dostupnosti podataka. Na službenoj stranici alata postavljen je tekst kojim se ograđuju od potencijalnih napada izazvani SQLmap alatom: „Korištenje SQLmap alata za napad na ciljeve bez prethodnog pristanka je nezakonito. Krajnji korisnik dužan je pridržavati se svih važećih lokalnih i državnih zakona. Programeri ne preuzimaju nikakvu odgovornost i nisu odgovorni za bilo kakvu zlouporabu ili štetu uzrokovanu ovim programom.“ [45].

4.5. DirBuster

DirBuster je multi praktičan alat s grafičkim sučeljem (GUI) koji dolazi s instalacijom Kali Linux-a, svrha alata je pronalaženje datoteka i direktorija na udaljenim web serverima korištenjem brute force metode. Web serveri često imaju sakrivene direktorije, a DirBuster ih pokušava pronaći korištenjem različitih metoda pretraživanja. Napad zahtijeva korištenje rječnika putem kojeg će se izvesti napad, standardni rječnici dolaze zajedno s Kali sustavom te je u većini slučajeva dovoljno koristiti takve tipove rječnika. Kao i kod ostalih alata korištenje DirBuster alata nad web stranicama ili aplikacijama za koje korisnik nema dopuštenje nije legalno. Slika 7 prikazuje grafičko sučelje DirBuster alata.



Slika 7. *Prikaz korisničkog sučelja DirbBuster alata*

Potrebno je upisati URL mete napada te odrediti ostalu konfiguraciju poput brzine izvođenja, ekstenzija dokumenata koji se pretražuju, vrstu napada i slično. Nakon što alat završi s pretraživanjem sakrivenih datoteka i direktorija s web servera potrebno je otići u karticu rezultati i proučiti pronađene direktorije. DirBuster koristi kodove koje server povratno šalje kao odgovor na zahtjev. Neki od najčešćih odgovora su: (200) potvrda datoteke i omogućavanje čitanja; (400) datoteka ne postoji; (301) datoteka je premještena na određeni URL; (401) neovlašten je pristup datoteci; (403) zahtjev je valjan, ali poslužitelj odbija odgovoriti na zahtjev.

4.6. Netcat

Netcat je uslužni program naredbenog retka koji čita i piše podatke putem mrežnih veza, koristeći TCP ili UDP protokole. Ovaj alat je moćan u primjeni mrežnih analiza i smatra se nezaobilaznim programom za primjenu prilikom izvođenja penetracijskog testiranja. Netcat je dostupan na Linux, macOS, Windows i BSD operativnim sustavima, a koristi se prilikom uklanjanja pogrešaka, nadzora mrežnih veza, skeniranja otvorenih portova, prijenosa podataka, udaljene komunikacije i drugih raznih mogućnosti. Netcat alat je unaprijed instaliran na Kali Linux distribuciji te njegovo korištenje nad neovlaštenim sustavima smatra se ilegalnom provedbom. Dizajniran je kao pouzdan *back-end* alat koji se može koristiti izravno, ali i na način da upravlja drugim programima i skriptama. Istodobno, to je alat za ispravljanje pogrešaka i istraživanje mreže bogat značajkama jer može stvoriti gotovo bilo koju vrstu veze koja je potrebna. Osnovna sintaksa Netcat alata ima sljedeći oblik:

```
nc [host] [port] [opcije]
```

Skeniranje portova jedno je od najčešćih korištenja Netcat alata, skeniranje može sadržavati jedan port ili određeni raspon. Također, koristi se za pronalazak poslužiteljskog softvera i njegove verzije. Primjer pronalaska porta 22 i njegove verzije:

```
echo "EXIT" | nc 10.10.8.8 22
```

Netcat se koristi prilikom prijenosa datoteka s jednog računala na drugi te stvaranjem internetskog razgovora. Oba načina se provode stvaranjem osnovnog modela klijent-poslužitelj, na način da Netcat sluša na određenom portu (korištenjem sintakse -l) i uspostavi TCP vezu s poslužiteljem. Netcat alat koristi se i tijekom slanja različitih zahtjeva na udaljene poslužitelje [46].

4.7. JoomScan

JoomScan je alat otvorenog koda za skeniranje ranjivosti, razvijen s ciljem provođenja automatiziranih zadataka poput otkrivanja ranjivosti i osiguranja pouzdanosti unutar Joomla CMS (engl. *Content Management System* – CMS) sustava. Omogućen je unutar Kali Linux distribucije jednostavnom instalacijom, a izrađen je u programskom jeziku PERL. JoomScan koristi modularnu arhitekturu koja omogućuje prepoznavanje ranjivosti poput pogrešne konfiguracije i nedostataka na razini administratora koje mogu biti od velikog značaja za ispitivače penetracijskog testiranja. JoomScan jedan je od glavnih alata prilikom prikupljanja podataka i identificiranja ranjivosti sustava temeljenih na Joomla sustavu [47]. Glavne mogućnosti primjenom ovog alata su:

- Pronalaženje verzije Joomla sustava
- Skeniranje više od 1209 najznačajnijih ranjivosti
- Pronalaženje kritičnih komponenti za održavanje sigurnosti sustava
- Detekcija vatrozida
- Pronalaženje log i backup datoteka te ispisivanje izvještaja u tekstualnom i HTML obliku

4.8. John The Ripper

John The Ripper (JtR) je objavljen 1996. godine kao alat za probijanje lozinki predviđen za rad unutar UNIX sustava. Cilj alata je testirati snagu lozinke i probijanje kriptiranih lozinki korištenjem brute force metode ili napadom temeljenim na rječniku, također omogućuje brz proces izvedbe, automatsko detektiranje hash algoritma te jednostavnost prilikom korištenja. Osnovna verzija dostupna je na stranicama GitHub-a, dok je profesionalna verzija osmišljena za korištenje od strane ispitivača penetracijskog testiranja. Profesionalna verzija nudi dodatne mogućnosti poput optimizacije performansi, višejezičnih rječnika te 64-bitnu arhitekturu. U većini situacija John The Ripper alat će automatizirano pronaći odgovarajući hash zapis i pokušati ga razbiti jednom od metoda. Međutim, postoji mogućnost da JtR alat ne prepozna odgovarajući hash zapis te je u tom slučaju potrebno ručno definirati određeni hash format korištenjem sintakse: --format=[vrsta hash zapisa] u nastavku naredbe. JtR omogućuje tri različite metode korištenja koje korisnici mogu izabrati, a to su single crack mode, wordlist mode i incremental mode.

Jednostruki način probijanja (engl. *Single crack mode*) koristi informacije iz UNIX passwd direktorija te se smatra najbržim načinom i preporučuje se u jednostavnijim pristupima. Ovaj tip metode koristi se kada ispitivač pretpostavlja da je korisnik postavio uobičajenu ili jednostavnu lozinku poput (*admin;admin*), to jest temelji se na metodi pogađanja. Način rada temeljen na rječniku (engl. *Wordlist mode*) koristi napad rječnikom na način da isprobava svaku lozinku zapisanu unutar dokumenta predviđenog za izvršavanje napada. Inkrementalni način (engl. *Incremental mode*) sastoji se od brute force napada koji su najsnažniji način probijanja lozinki. Temelji se na načinu da probija sustav korištenjem svih kombinacija, što u većini slučajeva stvara preveliki vremenski period ispitivanja i zbog toga ova vrsta metode često nije učinkovita. JtR koristi sljedeće enkripcijske tehnologije [48]:

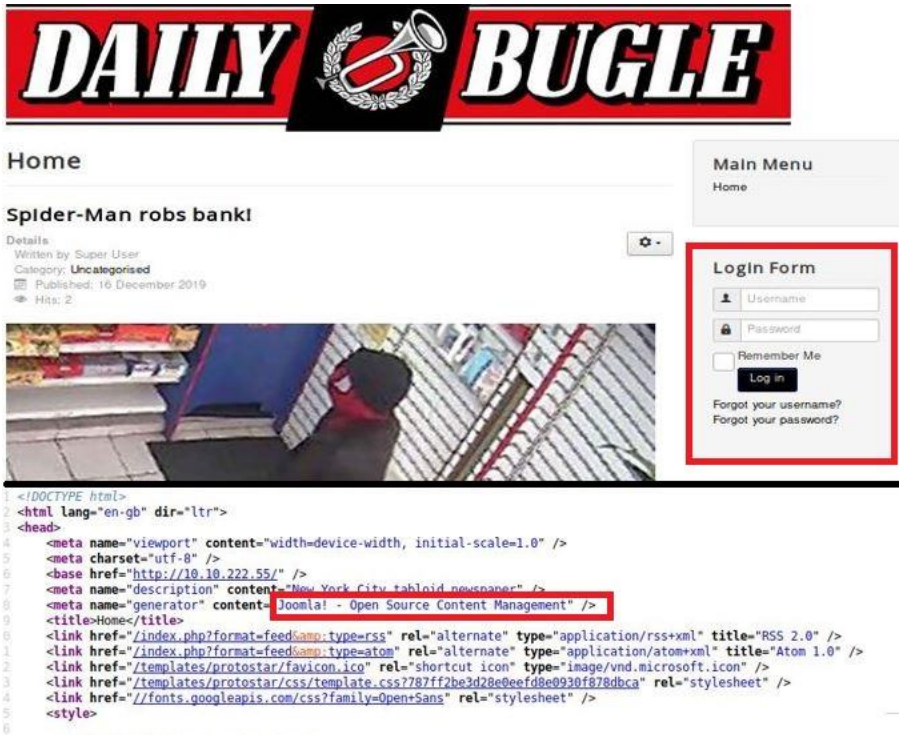
- UNIX crypt(3)
- DES
- bigcrypt
- BSDI
- FreeBSD MD5
- OpenBSD Blowfish
- Kerberos/AFS
- Windows LM
- DES-based tripcodes
- SHA-crypt hash
- SUNMD5 hash (Solaris)

5. Provedba penetracijskog testiranja i otkrivanje ranjivosti nad virtualnim okruženjem

U ovoj cjelini proveden je praktični dio diplomskog rada u kojem je postavljeno virtualno okruženje nad kojim je obavljeno penetracijsko testiranje, a za potrebe uspješnog provođenja testiranja korištena su saznanja prikupljena iz prethodnih poglavlja. Cilj praktičnog dijela je uspješno provođenje penetracijskog testiranja nad virtualnim okruženjem uz odgovarajući izvještaj s glavnim spoznajama testiranja i ponuđenim sigurnosnim koracima za uspostavljanje okruženja na višu sigurnosnu razinu. Kako bi testiranje bilo uspješno provedeno potrebno je uspostaviti *root* pristup nad ciljanim sustavom. Praktični dio prati metodologiju izvođenja opisanu kroz faze prikupljanja informacija, mapiranje mreže i identificiranja ranjivosti te iskorištavanja ranjivosti i dokumentacije. Dokumentacijom se smatra dostavljanje izvještaja koji je potrebno provoditi tijekom cijelog vremena ispitivanja. Izvještaj praktičnog dijela rada prikazan je kroz posljednje poglavlje i daje analizu i detaljno objašnjenje svih koraka provođenja napada nad ranjivim sustavom.

5.1. Faza prikupljanja podataka

Provođenje praktičnog dijela započinje fazom prikupljanja informacija, na način da se pristupi web stranici upisivanjem IP adrese u URL. Web stranica prikazuje dnevne novine s kraćim opisom članka te je potrebno pronaći što više informacija o web stranici kako bi faza identificiranja ranjivosti bila što opsežnija. Web stranica koristi HTTP (engl. *HyperText Transfer* Protocol) protokol što daje informacije kako stranica nije kriptirana i kako će se lakše moći provesti napadi penetracijskim testiranjem.



Slika 8. *Izgled web stranice ranjivog sustava*

5.2. Faza mapiranja i identificiranja ranjivosti

Na web stranici je vidljiva mogućnost prijave što se potencijalno u nastavku rada može iskoristiti za dobivanje pristupa. Nadalje, potrebno je pronaći tip i verziju web stranice s potrebom analiziranja i pronalazaka ranjivosti. Korištenjem opcije *View Page Source* moguće je pristupiti izvornom kodu stranice u kojem se vrlo često nalaze kritični podaci potrebni za prikupljanje informacija. Na slici 8 prikazan je kod u kojem je pronađena informacija da web stranica sadrži Joomla sustav za upravljanje web sadržajem otvorenog koda. Također, moguće je pretraživanje robots.txt direktorija, koji osim potvrde da se radi o Joomla sustavu prikazuje i određene direktorije koji će u nastavku biti analizirani.

Sljedeći korak penetracijskog testiranja zahtijeva korištenje automatiziranih alata za prikupljanje podataka i mapiranje mreže. Korištenjem jednog od najučestalijih alata Nmap, potrebno je skenirati mrežu virtualnog okruženja. Rezultatima skeniranja prikazana su tri otvorena porta: (Slika 9)

- 22/tcp - *SSH* servis verzije *OpenSSH 7.4*
- 80/tcp - *http* servisa verzije *Apache httpd 2.4.6*
- 3306/tcp - *mysql* servisa verzije *MariaDB*

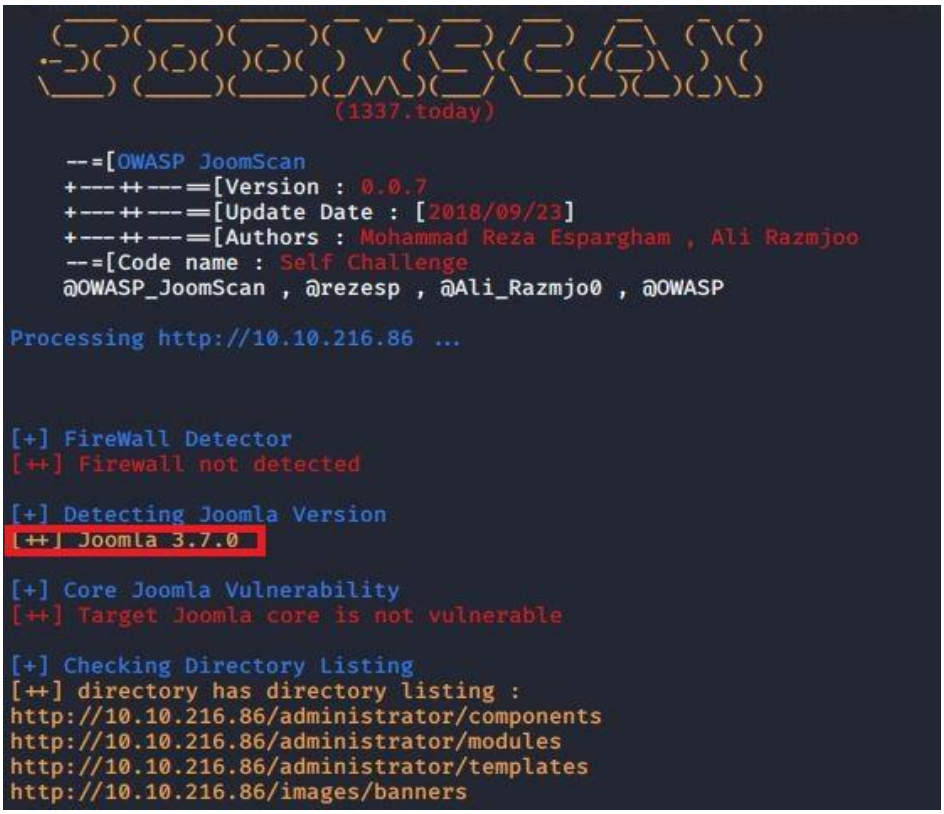
Korištenjem SSH otvorenog porta potencijalno će se ostvariti udaljeni pristup na ciljanom poslužitelju, dok će se kroz *mysql* port pokušati kroz SQL injection napade pristupiti informacijama iz baze podataka.

```
root@kali:~# nmap 10.10.65.247 -sV
Starting Nmap 7.80 ( https://nmap.org ) at 2021-08-27 10:28 CDT
Nmap scan report for 10.10.65.247
Host is up (0.053s latency).
Not shown: 997 closed ports
PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 7.4 (protocol 2.0)
80/tcp   open  http    Apache httpd 2.4.6 ((CentOS) PHP/5.6.40)
3306/tcp open  mysql   MariaDB (unauthorized)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 8.29 seconds
root@kali:~# _
```

Slika 9. *Rezultati skeniranja Nmap alatom*

Sljedeći alati kojima se prikupljaju nove informacije vezane o sustavu su DirBuster i JoomScan koji su ranije u radu opisani najbitnijim značajkama. DirBuster alat potrebno je postaviti na željenu IP adresu ranjivog sustava, odrediti *brute force* napad temeljen na *rockyou.txt* rječniku te započeti pretraživanje. Rezultati prikupljeni DirBuster alatom prikazali su nekoliko direktorija na poslužitelju koje je potrebno istražiti. Paralelno s pokretanjem DirBuster alata, postavljen je i JoomScan kako bi provjerili rezultate pronađene ovim alatom. Rezultati dobiveni ovim korakom (slika 10) su slični, uz napomenu da je JoomScan alat detektirao 3.7.0 verziju Joomla sustava. Pretraživanjem jednog od direktorija, točnije direktorija */administrator*, pristupili smo web stranici za prijavu administratorskim ovlastima.



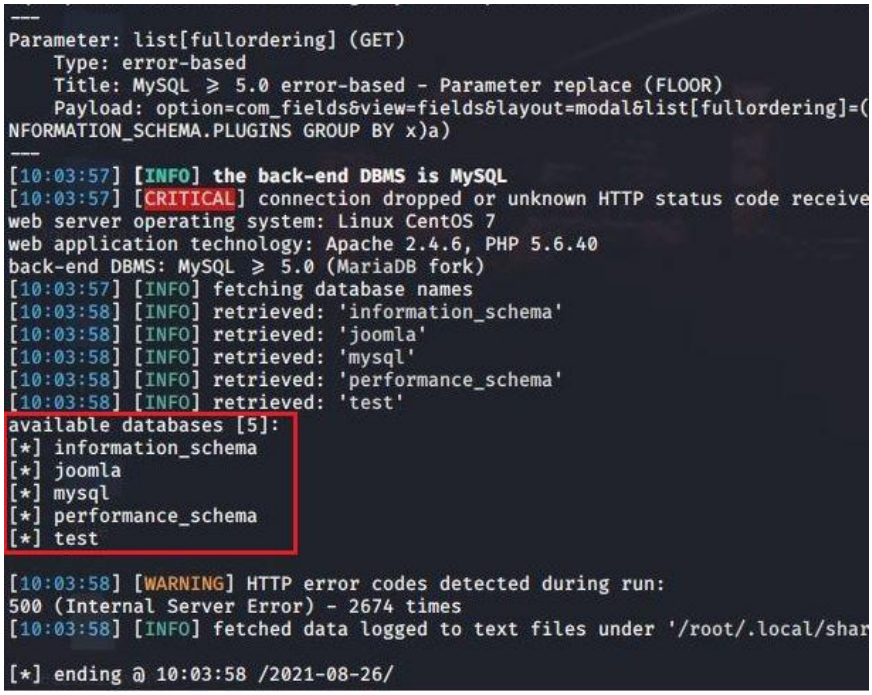
Slika 10. *Rezultati pretraživanja JoomScan alatom*

Sljedeći korak metodologije penetracijskog testiranja je identificiranje ranjivosti unutar kojeg je potrebno odrediti ranjivosti podataka koji su prikupljeni u prethodnoj fazi. Korištenjem naredbi: *searchsploit joomla 3.7.0 i searchsploit OpenSSH 7.4* pronađene su pojedine ranjivosti. Rezultat prikazuje kako je Joomla verzija 3.7.0 ranjiva na SQL Injection napad, kao i Cross-Site Scripting dok rezultati pretraživanja ranjivosti SSH-a verzije 7.4 daju nekoliko rezultata prikazanih na slici 11. Dohvaćanjem dokumenta (*php/webapps/42033.txt*) za dodatno objašnjenje SQL ranjivosti predložen je napad korištenjem sqlmap alata. Potrebno je napomenuti kako je moguć napad nad sustavom na više načina te kako nisu korištene sve ranjivosti i metode u praktičnom dijelu rada. Ostale ranjivosti kao i metode, među kojima je i korištenje Joomblah skripte detaljno su analizirani u posljednjoj cjelini, gdje su obuhvaćene sve pronađene ranjivosti i svi uspješno izvršeni napadi koji nisu prikazani u ovom poglavlju.

Slika 11. *Rezultati identificiranja ranjivosti*

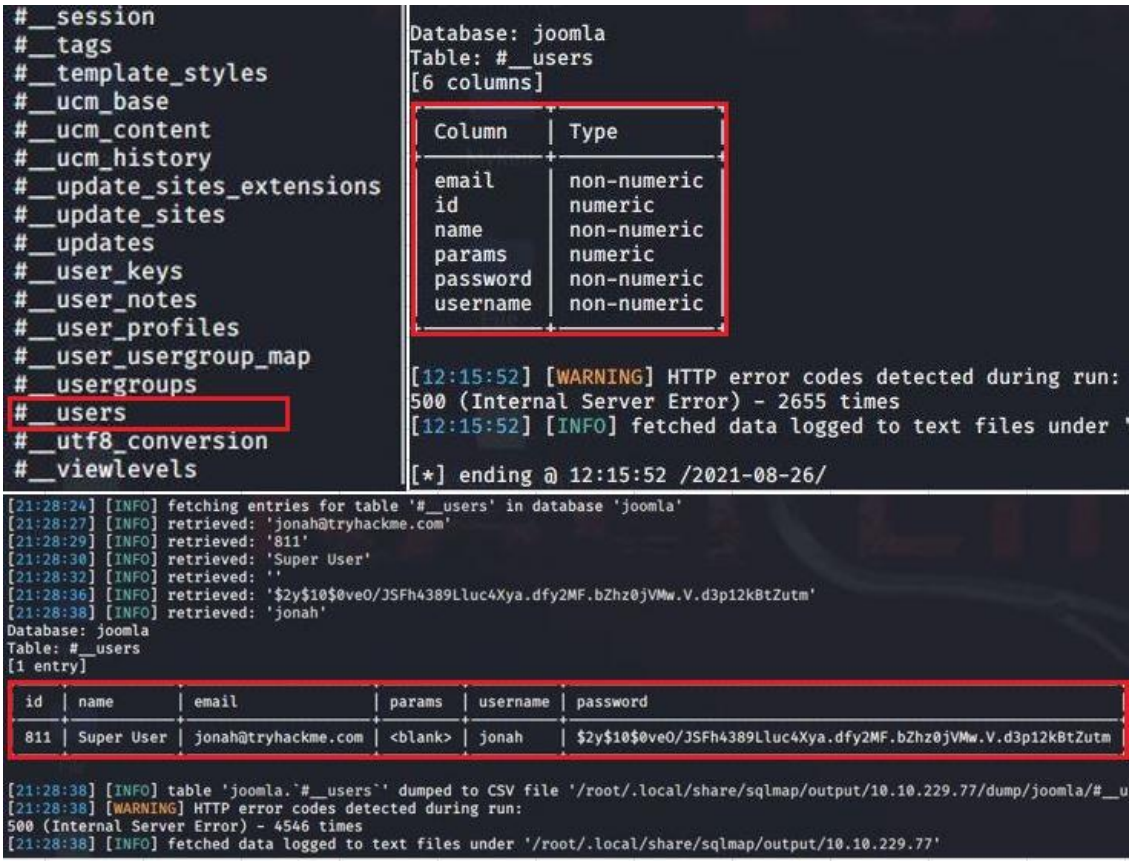
5.3. Faza iskorištavanja ranjivosti

Glavna i najzahtjevnija faza penetracijskog testiranja dolazi nakon identificiranja ranjivosti, a to je iskorištavanje ranjivosti, odnosno penetracija sustava. Dosad prikupljenim podacima, pronađene su pojedine ranjivosti sustava, a ključni cilj ispitivača je infiltrirati se unutar sustava udaljenim pristupom te ukoliko je ostvarivo povećati privilegije i prikupiti osjetljive podatke sustava. Prvi pokušaj iskorištavanja ranjivosti izvodi se korištenjem sqlmap alata za provođenje napada SQL injekcija nad Joomla 3.7.0 verzijom. Potrebna naredba korištena u sqlmap alatu za probijanje baze podataka prikupljena je s web stranice *Exploit Database* koja sadrži veliki broj ranjivosti i način njihovog iskorištavanja. Provedeni napad pronašao je pet baza podataka (Slika 12) te nakon analiziranja svake od njih utvrđeno je kako baza podataka joomla najviše odgovara potrebama ispitivača za nastavak penetracijskog testiranja.



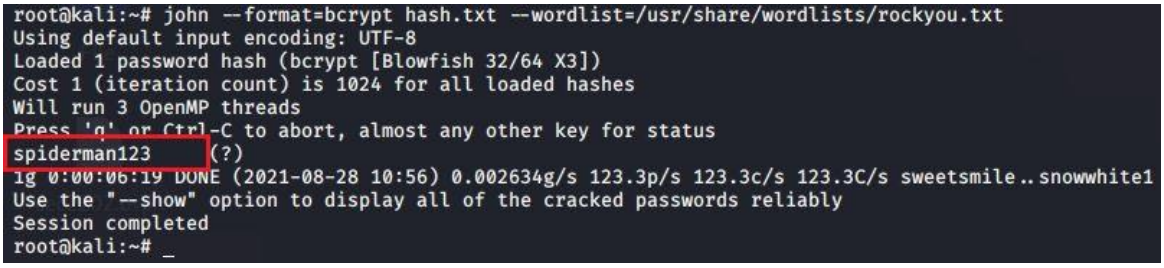
Slika 12. *Rezultat pronalaska baza podataka korištenjem sqlmap alata*

Daljnji napadi zahtijevaju pristup informacijama iz joomla baze podataka, gdje se sqlmap alatom dobivaju podaci o tablicama ispitivane baze. Zatim je potrebno analizirati svih 70 tablica dobivenih napadom te pronaći tablicu unutar koje su spremljeni podaci o korisnicima. Najveća vjerojatnost tablice s korisničkim podacima je *#__users* nad kojom je potrebno provesti ispitivanje. Napadom na tablicu pronađeni su stupci *email*, *id*, *name*, *paras*, *password* i *username*. Nakon potpune provedbe napada sqlmap alatom nad bazom dohvaćeni su podaci korisničkog imena i lozinke koji su potrebni za prijavu na web stranicu. Svi podaci prikupljeni u ovom napadu prikazani su na slici 13. Također, postupak prikupljanja informacija o administratoru moguće je izvesti korištenjem prethodno spomenute Python skripte Joomblah, a rezultat testiranja korištenjem tog alata jednak je sqlmap rezultatu.



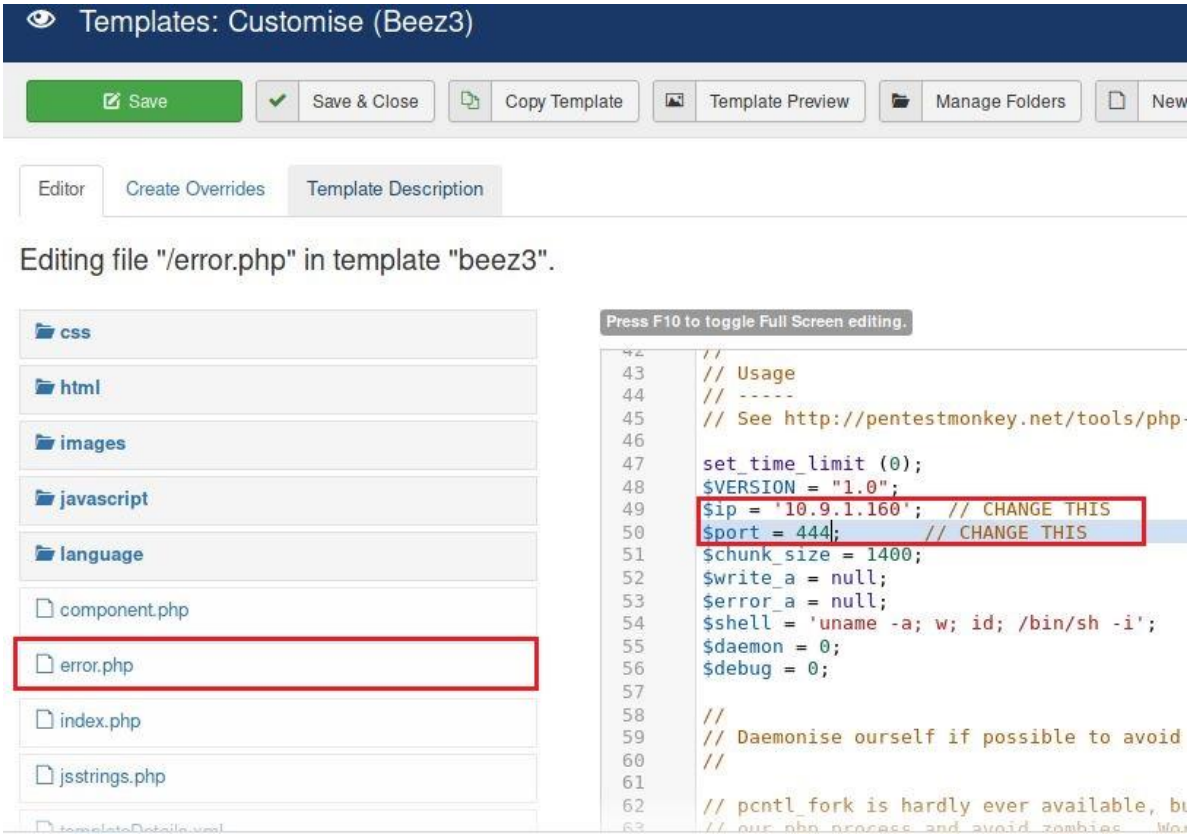
Slika 13. *Cjelokupni rezultat provedbe napada SQLmap alatom*

Informacije potrebne za pristupanje administratoru web stranice su korisničko ime: *jonah* i lozinka u hash obliku koju je potrebno probiti pronalaženjem odgovarajućeg format zapisa. Metoda pronalaska formata hash zapisa moguća je na više načina, među kojima su korištenjem *hashid* naredbe unutar terminala, uz pomoć *CyberChef* mrežne stranice koja sadrži veliki spektar mogućnosti među kojima je i analiziranje hash zapisa te korištenjem alata John The Ripper. U ovom radu koristila se metoda pronalaska formata kroz terminal, a korištenje naredbe *hashid* rezultiralo je pronalaskom *bcrypt* formata. Nakon pronalaska formata potrebno je razbiti hash zapis uz pomoć John The Ripper alata korištenjem naredbe prikazanoj na slici 14. Naredbom je potrebno definirati format zapisa (*bcrypt*), rječnik kojim će se provesti napad nad hash zapisom (*rockyou.txt*) te datoteku unutar koje je zapisan hash zapis, odnosno potpunu lokaciju do hash zapisa (*/root/hash.txt*). Rezultat korištenja alata prikazuje lozinku potrebnu za pristup web stranici s ovlastima administratora.



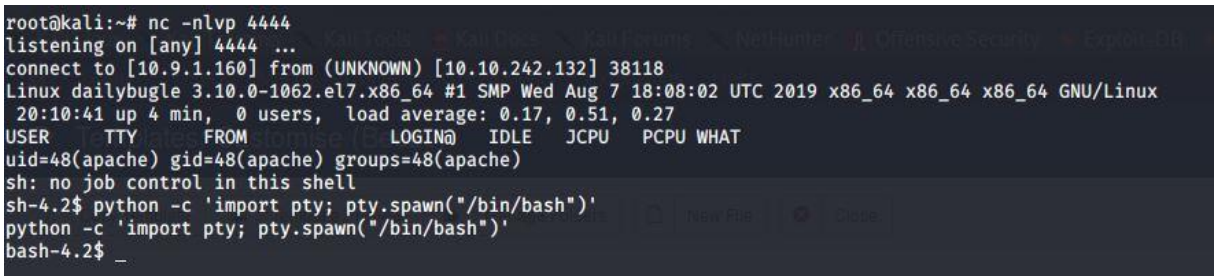
Slika 14. *Rezultat razbijanja hash zapisa i pronalazak lozinke*

Nakon pristupanja web stranici s privilegijama administratora omogućene su razne izmjene nad stranicom. Potrebno je detaljno analizirati svaku od mogućnosti te identificirati ranjivosti koje će ugroziti sigurnost sustava. Nakon detaljne analize pojedinih mogućnosti pronađena je mogućnost ubacivanja reverzibilnog shell-a unutar segmenta *template*. Potrebno je ubaciti shell umjesto koda unutar *error.php* datoteke koja će uspostaviti sesiju s poslužiteljem. Na stranici *github* kopiran je kod *php-reverse shell-a* koji je potrebno ubaciti te izmijeniti s podacima računala napadača (IP adresa i port slušanja) kako bi se uspostavila sesija između udaljenog poslužitelja i uređaja napadača (slika 15).



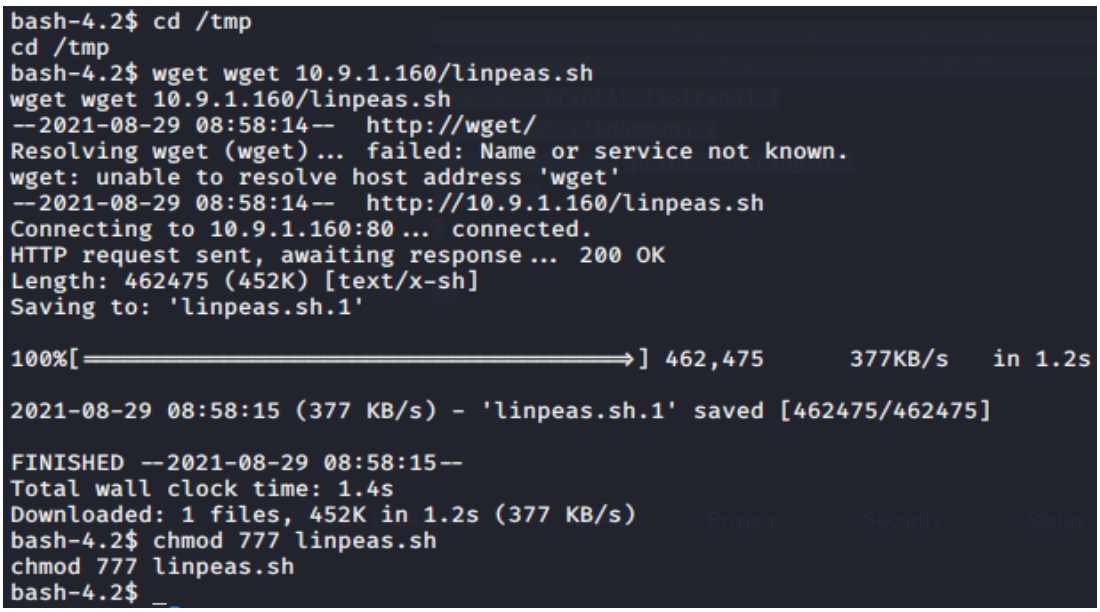
Slika 15. *Izmjena error.php koda web stranice s php reverse shell-om*

Nakon spremanja izmjena potrebno je uz pomoć Netcat alata postaviti slušanje istog porta koji je postavljen unutar *php reverse* shell-a. U ovome zadatku postavljeni port slušanja je 444. Zatim je potrebno ručno pokrenuti skriptu postavljenu na web stranicu na način da se pokrene *error* stranica čiji je izvorni kod zamijenjen reverzibilnim shell-om. URL put određen je kroz ime *tamplate-a* i *error.php-a*, odnosno potrebno je upisivanje sljedećeg URL puta: *http://10.10.65.247/templates/beez3/error.php*. Nakon što je uspostavljena sesija s ciljanim sustavom potrebno je unaprijediti shell. Jedna od mogućnosti je korištenje *pty* modula koji održava stabilnu konekciju te omogućuje stvaranje pseudo-terminala s ciljem zaobilaženja sudo naredbi (slika 16).



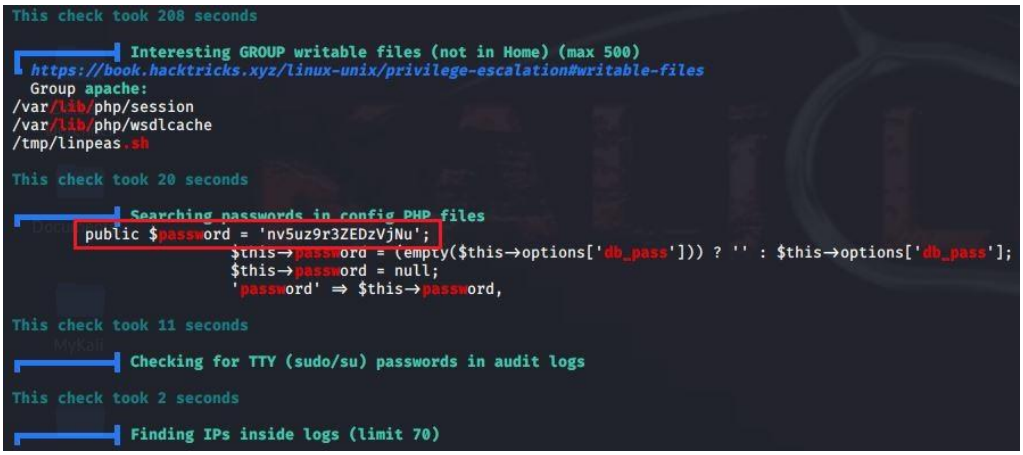
Slika 16. *Prikaz uspješno pokrenutog shell-a i stvaranje stabilne konekcije*

Daljnji postupak je prikupiti informacije potrebne za pristup sustavu putem SSH porta koji je ranije skeniran i definiran kao otvoren. Za početak potrebno je pretražiti sve korisne informacije korištenjem interaktivnog shell-a, mogući način je ručno pretraživanje pojedinih direktorija ili korištenje predviđenih skripti koji će obaviti ispitivanje umjesto ispitivača. Jedna od skripti koja to omogućuje je i Linpeas skripta predviđena za pretraživanje sustava s ciljem dobivanja veće razine privilegija. Kako bi se iskoristila Linpeas skripta potrebno ju je prebaciti na ciljano računalo. Proces slanja datoteka moguć je kroz više načina, a u ovom ispitivanju provedeno je korištenjem *SimpleHTTPServer* modula. Kako bi se uspješno preuzela skripta na udaljenom računalu potrebno je pretražiti koji direktoriji unutar interaktivnog shell-a daje privilegije za izvršavanje naredbi te pokretanje skripte. Korištenjem naredbe *ls -la* prikazuje se kako su najviše ovlasti omogućene kroz direktoriji /tmp, stoga je sljedeći korak ulazak u taj direktoriji putem terminala. Nakon toga, na ispitivačevom računalu je potrebno pokrenuti modul korištenjem naredbe: *python -m SimpleHTTPServer 80*, čime je otvoren HTTP server na portu 80 koji omogućuje drugim uređajima preuzimanje datoteka s tog računala. Ciljano računalo zatim koristi naredbu *wget http://IP_adresa/linpeas.sh* za preuzimanje skripte na svoju stranu računala. Nakon uspješnog preuzimanja potrebno je promijeniti ovlasti nad preuzetom skriptom korištenjem naredbe *chmod 777 linpeas.sh*, čime su omogućene privilegije svim korisnicima da čitaju, prepisuju i izvršavaju Linpeas skriptu (Slika 17).



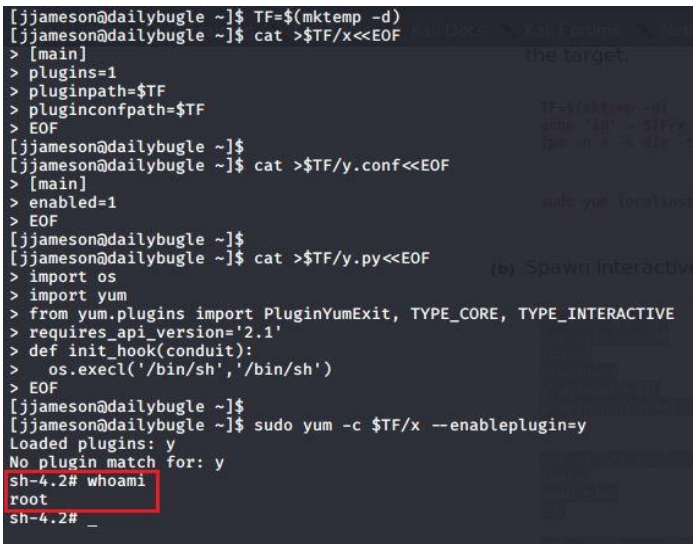
Slika 17. *Prikaz procesa preuzimanja i promjena ovlasti nad Linpeas skriptom*

Nakon pokretanja skripte na udaljenom računalu pronađene su potencijalne ranjivosti za povećanje privilegija, analizirani su razni direktoriji sa svrhom pronalaska korisničkom imena i lozinke korisnika većih ovlasti. Rezultati su prikazali korisničko ime *jjameson* koje bi moglo omogućiti pristup većih ovlasti, a kroz daljnje analiziranje rezultata Linpeas skripte pronađena je lozinka unutar *configuration.php* direktorija koja potencijalno može biti pristup jjameson korisniku.



Slika 18. *Rezultat korištenja Linpeas skripte i pronalazak lozinke*

Sljedeći zadatak testiranja je pristupiti sustavu putem SSH porta korištenjem informacija prikupljenih u prethodnom koraku. Za pristup je potrebno korištenje sljedeće naredbe: *ssh jjameson@10.10.65.247* te dodatnim upisivanjem lozinke prikazane na slici 18. Nakon uspješne prijave potrebno je dodatno istraživanje novih privilegija s ciljem postizanja root privilegija. Korištenjem naredbe *sudo -l* koja opisuje koje su dopuštene ili zabranjene naredbe za korisnika koji poziva naredbu, prikazane su mogućnosti korištenja *yum* metode. Na stranici *GTFOBins* [49] pronađena je naredba za postizanje root privilegija, vrsta koda i način korištenja prikazani su na slici 19. Virtualno okruženje uspješno je testirano te je omogućen root pristup na udaljenom računalu. Detaljna analiza i izvještaj provođenja penetracijskog testiranja ovog sustava opisan je kroz sljedeće poglavlje.



Slika 19. *Prikaz ostvarivanja root pristupa*

6. Analiza prikupljenih podataka

Nakon provođenja penetracijskog testiranja potrebno je sastaviti strukturiranu dokumentaciju s najvažnijim saznanjima stečenim prilikom izvođenja testiranja. Unutar dokumentiranog izvještaja potrebno je predočiti analizu prikupljenih podataka te da dati osvrt i prijedloge za rješavanje ranjivosti sustava. U ovom poglavlju analizirat će se informacije i ranjivosti prikupljeni u praktičnom dijelu rada te predviđene mjere zaštite sustava. Sustav provođenja metodologije razdvojen je u elemente izvještaja, gdje je svaki od elemenata opisan kao zasebna cjelina. Svrha izvještaja je dati jednostavan i lako razumljiv pogled na sigurnosne aspekte sustava te opisati ranjivosti kroz njihove razine i dodatno ukazati na one ranjivosti koje mogu trajno naštetiti sustavu.

6.1. Izvještaj o fazi prikupljanja podataka

Prilikom ručnog prikupljanja podataka uočeno je korištenje HTTP protokola što ukazuje na nedovoljnu sigurnosnu zaštitu web stranice. HTTP je protokol aplikacijske razine koji omogućava prijenos datoteka koje u sebi sadrže veze na druge dokumente, a funkcionira na principu zahtjev-odgovor u okviru klijent-poslužitelj arhitekture. Postoji veliki broj ranjivosti HTTP protokola koji se mogu iskoristiti s ciljem pristupanja osjetljivim podacima ili nanošenje štete sustavima. Jedna od ranjivosti koja se može iskoristiti je presretanje HTTP poruka s ciljem prikupljanja kritičnih podataka poput: korisničkog imena, lozinke, lokacije, informacija s kreditnih kartica ili slično. Zaglavlje i tijelo HTTP poruka koriste se za pohranu podataka koji nisu kriptirani, čime se svakim pristupanjem HTTP zahtjevu ili odgovoru omogućuje pristupanje podacima unutar HTTP poruke, ovakav način najčešće se koristi prilikom napada trećih strana odnosno metodom napada čovjeka u sredini (engl. *Man in the Middle* – MITM). Osim presretanja HTTP poruka, moguće je korištenje log datoteka koje u većini slučajeva koriste poslužitelji s ciljem praćenja zapisa o posluženom HTTP prometu. Pristupanju HTTP log zapisima moguće je otkriti informacije o korisnicima ili podatke od značaja za određene sustave. Također, ranjivosti protokola su HTTP zaglavlja te ranjivosti prilikom analize URL-a ukoliko se koriste HTTP GET zahtjevi. Većina ranjivosti protokola mogu dovesti do neovlaštenog pristupanja informacijama o korisnicima, sustavu ili poslužitelju [50].

Nad HTTP protokolom moguće je izvesti više vrsta napada. Napad nad datotečnim sustavom jedan je od najčešće korištenih metoda kod penetracijskog testiranja web aplikacija s HTTP protokolom, a korišten je i u praktičnom dijelu uz pomoć automatiziranog DirBuster alata. Način zaštite virtualnog okruženja od pristupanju direktoriju moguća je ograničavanjem URL-a, na način da se ne dopuštaju modifikatori putanja unutar URL-a onim direktorijima koji su od značaja za sigurnost sustava. Nadalje, moguće je ostvariti MITM i DDoS napade koji se izvode slanjem velikih količina zahtjeva s ciljem preopterećenja poslužitelja. Također, moguće je izvesti napad prelijevanjem spremnika kao i SQL injection. Napad prelijevanjem spremnika izvodi se na način da se putem web formi ili URL-a pošalje zahtjev na poslužitelj s većom količinom podataka od one koju ima određeni spremnik. Napad temeljen na SQL injekcijama provodi se na način da se korištenjem URL-a traže zahtjevi iz baza podataka, a ispravni kod na poslužitelju prikazat će bazi podataka naredbu potrebnu za dohvat podataka.

HTTPS (engl. *HyperText Transfer Protocol Secure*) je sigurna inačica HTTP protokola korištena za pružanje sigurnosti na webu. Omogućuje autentifikaciju i kriptiranu komunikaciju te umjesto komuniciranja nezaštićenim tekstom, HTTPS protokol kriptira podatke tijekom korisničke sesije, koristeći pritom TLS protokol čime ostvaruje veću sigurnosnu zaštitu. Razina zaštite koju pruža HTTPS protokol ovisi o ispravnosti implementacije u web pregledniku, vrsti i obilježjima poslužiteljskog softvera i o podržanim kriptografskim algoritmima. HTTPS protokol provjerava identitet na web stranicama ili web poslužiteljima na koje se klijent želi spojiti te kriptira gotovo sve informacije koje se šalju između korisnika i web stranice ili usluge [51].

Tablica 4. *Razlika između HTTP i HTTPS protokola*

HTTP (HyperText Transfer Protocol)	HTTPS (HyperText Transfer Protocol Secure)
Ne kriptira tekst	Kriptira tekst
Nudi sporije pretraživanje	Nudi kvalitetnije pretraživanje
Djeluje na aplikacijskom sloju	Djeluje na transportnom sloju
Koristi port 80	Koristi port 443
Ne zahtijeva sigurnosne certifikate	Zahtijeva sigurnosne certifikate

U tablici 4 prikazane su prednosti koje nudi HTTPS u odnosu na HTTP protokol. HTTPS pruža veću sigurnost i manju vjerojatnost od napada te kvalitetnije pretraživanje. Također, potrebno je napomenuti kako HTTPS koristi port 443 dok HTTP protokol koristi port 80. Iako HTTPS također ima ranjivosti, znatno je sigurniji od HTTP protokola, stoga se preporučuje nadogradnja web aplikacija sigurnijim protokolom. Za nadogradnju je potrebno kupiti i instalirati SSL (engl. *Secure Socket Layer*) certifikat na web aplikaciju te postaviti 301 preusmjeravanje koje omogućuje automatsko preusmjeravanje svog prometa na siguran HTTPS protokol.

Robots.txt je tekstualna datoteka pomoću koje se daju određene instrukcije pretraživačima ili web robotima koje stranice je potrebno indeksirati, a koje ne. Korištenje datoteke robots.txt sama po sebi nije sigurnosna ranjivost. Međutim, podaci unutar datoteke mogu pomoći napadaču u prikupljanju informacija o web stranici, osobito ako neke od identificiranih lokacija nije moguće pronaći drugačijim načinom. Robots.txt može predstavljati sigurnosnu prijetnju ukoliko se ne provodi pravilna kontrola pristupa i otkriva se put do administrativnih podataka. Zaštita od prijetnji nad robots.txt datotekom omogućuje se kontrolom podataka na način da se izbjegava postavljanje osjetljivih informacija unutar robots.txt datoteke.

6.2. Izvještaj o fazi mapiranja mreže

Mapiranje mreže korištenjem Nmap alata rezultiralo je saznanjima da su portovi 20, 80, 3306 otvoreni. Otvoreni portovi općenito nisu prijetnja po sustavu, ukoliko su ispravno postavljeni. Otvoreni portovi mogu predstavljati prijetnju ukoliko su servisi korišteni na tim portovima pogrešno konfigurirani, neažurirani te ranjivi na određene napade ili jednostavno imaju slabu mrežnu zaštitu. U praktičnom dijelu rada iskorištene su ranjivosti SSH i mysql servisa otvorenih portova. SSH je mrežni protokol koji korisnicima omogućuje uspostavu sigurnog komunikacijskog kanala između dva računala putem računalne mreže. Veliki je broj ranjivosti koje proizlaze nesigurnim korištenjem otvorenog SSH porta, a u praktičnom radu iskorištena je ranjivost neovlaštenog pristupa, nastala pogrešnom konfiguracijom porta. Načini kako zaštititi SSH portove su sljedeći [52]:

1. Postaviti SSH na nasumični port s ciljem zavaravanja napadača iz razloga što tada SSH servis nije na svom standardnom portu.
2. Postaviti *TCP Wrrapers* kojim se omogućuje sigurnosna kontrola te mogućnost sortiranja i filtriranja pristupa SSH poslužitelju.
3. Potrebno je onemogućiti root prijavu.
4. Onemogućiti *brute force* napade korištenjem vatrozida ili alata poput *SSHGuard.*
5. Filtrati otvoreni port uz pomoć vatrozida.

Mysql je sustav otvorenog koda za upravljanje bazom podataka. Sustav za upravljanje čini mysql poslužitelj koji je dostupan kao program za različitu upotrebu te se može implementirati u zasebne aplikacije, prema zadanim postavkama pokreće se na portu 3306. Ranjivosti s kojima se susreće ovaj sustav su *brute force* napadi, pokretanje upita bez prijave u mysql, prikupljanje podataka kroz *schema-dump* ili mysql poslužitelja te enumeracija direktorija i datoteka. Kroz testiranje virtualnog okruženja omogućen je pristup informacijama iz baze podataka pomoću sqlmap alata korištenjem SQL injection-a uz metodu *brute force* napada. Načini zaštite mysql baze podataka su sljedeći [53]:

1. Zaštita porta uz pomoć filtriranja.
2. Izmjena mysql servisa sa standardnog 3306 porta.
3. Izbjegavanjem davanja informacija o pristupu tablicama i bazama ne autoriziranim korisnicima.
4. Koristiti enkripciju podataka unutar baze.
5. Koristiti INVOKE i GRANT naredbe za kontrolu mysql sustava kako bi se odredile privilegije korisnika.
6. Koristiti lozinke koje se ne nalaze unutar standardnih rječnika.
7. Ispravno konfigurirati sigurnosne aspekte zaštite poput vatrozida.
8. Onemogućiti prijenos podataka putem interneta ukoliko informacije nisu kriptirane.

6.3. Izvještaj o fazi identificiranja ranjivosti

Jedan od načina identificiranja ranjivosti u praktičnom dijelu rada bilo je korištenjem Nessus alata kojim su pronađene ranjivosti sustava različitih razina. Većina informacija pronađenih Nessus alatom daju uvid u opće informacije sustava koje se također mogu iskoristiti prilikom testiranja sustava. Opće informacije prikazuju podatke vezane o robots.txt datoteci, SSH protokolu, Nessus skeniranju, PHP i Apache verziji te detekciji servisa, *JQuerry*-a, OS-a i *tracerout*-a. Među ranjivostima koje je pronašao Nessus alat su korištenje HTTP Track metode i napad korištenjem *XSS* metode koje su definirane kao ranjivosti srednje razine te korištenje *CBC* metode unutar SSH sustava prikazane kao ranjivost niske razine (Slika 20).

MEDIUM	5.0	11213	HTTP TRACE / TRACK Methods Allowed
MEDIUM	4.3	136929	JQuery 1.2 < 3.5.0 Multiple XSS
LOW	2.6	70658	SSH Server CBC Mode Ciphers Enabled
INFO	N/A	10114	ICMP Timestamp Request Remote Date Disclosure
INFO	N/A	18261	Apache Banner Linux Distribution Disclosure
INFO	N/A	48204	Apache HTTP Server Version
INFO	N/A	84574	Backported Security Patch Detection (PHP)
INFO	N/A	39520	Backported Security Patch Detection (SSH)
INFO	N/A	39521	Backported Security Patch Detection (WWW)
INFO	N/A	45590	Common Platform Enumeration (CPE)
INFO	N/A	54615	Device Type
INFO	N/A	10107	HTTP Server Type and Version
INFO	N/A	24260	HyperText Transfer Protocol (HTTP) Information
INFO	N/A	106658	JQuery Detection
INFO	N/A	117886	Local Checks Not Enabled (info)
INFO	N/A	11219	Nessus SYN scanner
INFO	N/A	19506	Nessus Scan Information
INFO	N/A	11936	OS Identification
INFO	N/A	48243	PHP Version Detection
INFO	N/A	70657	SSH Algorithms and Languages Supported
INFO	N/A	149334	SSH Password Authentication Accepted
INFO	N/A	10881	SSH Protocol Versions Supported
INFO	N/A	10267	SSH Server Type and Version Information
INFO	N/A	22964	Service Detection
INFO	N/A	25220	TCP/IP Timestamps Supported
INFO	N/A	110723	Target Credential Status by Authentication Protocol - No Credentials Provided
INFO	N/A	10287	Traceroute Information
INFO	N/A	10302	Web Server robots.txt Information Disclosure

Slika 20. *Rezultati skeniranja Nessus alatom*

Trace i *Track* su HTTP metode koje se koriste prilikom ispravljanja pogrešaka veza web poslužitelja. Neovlašteni korisnici mogu zloupotrijebiti ovu metodu s ciljem pristupa osjetljivim podacima unutar HTTP zaglavlja prilikom postavljanja HTTP zahtjeva. Kako bi se zaštitio sustav od potencijalne ranjivosti *Trace* i *Track* metoda potrebno je onemogućiti *Trace* metodu unutar web poslužitelja ili ažurirati pakete web poslužitelja kako bi prema zadanim postavkama onemogućili *Trace* metodu. Druga ranjivost srednje razine upozorava da je sustav ranjiv na *Cross Site Scripting* napade, a kako bi se osiguralo od te ranjivosti potrebno je ažurirati *JQuerry* verziju. *SSH Server CBC Mode Ciphers Enabled* je ranjivost koja iskorištava CBC (engl. *CBCher Block Chaining*) enkripciju na SSH poslužitelju, što omogućuje napadaču da oporavi *plaintext* podatke, odnosno ne kriptirane podatke koje čekaju za unos u kriptografske algoritme. Kako bi se sustav zaštitio potrebno je onemogućiti šifriranje CBC načinom i omogućiti CTR ili GCM šifriranje [54].

Joomla 3.7.0 'com_fields' SQL injection je ranjivost koja je iskorištena prilikom provođenja testiranja nad virtualnim okruženjem, a rezultirala je dobivanjem korisničkih podataka za prijavu kao administrator. Komponenta *com_fields* omogućuje posuđivanje pogleda iz baze na

administratorskoj strani. Korištenjem URL-a u kodu sqlmap alata prikazanog u praktičnom dijelu rada omogućuje se pristup popisu svih prilagođenih polja dostupnih na web mjestu koje se mogu zatražiti na temelju različitih parametara. Iskorištavanjem ove ranjivosti omogućuje se napadaču da kompromitira aplikaciju te pristupi ili izmijeni podatke unutar baze podataka. Kako bi se zaštitilo od prijetnje potrebno je ažurirati Joomla verziju na 3.7.1 ili kasniju.

6.4. Izvještaj o fazi iskorištavanja ranjivosti

Nakon analiziranja ranjivosti bilo je potrebno provesti SQL injection napad kako bi se prikupili podaci o korisniku. Osim korištenja SQLmap alata moguće je provođenje napada korištenjem Joomblah skripte, a rezultati napada prikazani su na slici 21. Korištenjem John The Ripper alata omogućilo se razbijanje *bcrypt* hash zapisa. Nakon pristupanja administratorskim ovlastima na web stranici koristio se *php-reverse shell* koji je zamijenjen s *error.php* kodom te uz korištenje Netcat alata i pokretanje shell-a uspostavila se sesija s udaljenim računalom. Sljedeći korak bio je održati stabilnu konekciju s poslužiteljem te povećati privilegije unutar sustava. Korištenjem Linpeas skripte omogućeno je pronalaženje korisničkog imena i lozinke unutar *configuration.php* direktorija, čime je omogućen pristup SSH portu s podacima korisnika.

Slika 21. *Rezultati korištenja Jooblah skripte*

Nakon pristupanja sustavu kroz SSH port potrebno je bilo dodatno povećati ovlasti što je omogućeno korištenjem pomoćnog programa za upravljanje datotekama yum (engl. *The Yellowdog Updater Modified)*. Tehnika povećanja privilegija na root korisnika omogućena je zbog pogrešne konfiguracije koja je omogućila pokretanje yum naredbi kroz sudo funkciju. Na slici 20 prikazan je kod koji omogućuje pristupanje root privilegijama, na način da je upisanim kodom unutar shell-a omogućeno stvaranje privremenog direktorija unutar kojeg su napravljene tri datoteke. Unutar svake datoteke unesen je sadržaj potreban za ostvarivanje pristupa te je kroz naredbu: *sudo yum -c $TF/x --enableplugin=y* omogućeno pokretanje privremenog direktorija unutar kojeg se nalazi upisani *payload* pomoću kojeg se ostvaruje interaktivni root shell. Kako bi se povećala sigurnost i otklonile ranjivosti virtualnog okruženja testiranog u ovom radu potrebno je ažurirati Joomla CMS sustav na najnoviju verziju, zatim koristiti kontrolu pristupa i vatrozid s ciljem zaštite portova te izmijeniti pogrešnu konfiguraciju sustava koja omogućuje prikupljanje lozinke kroz *configuration.php* datoteku te mogućnost korištenja *yum* programa za upravljanje datotekama kroz sudo naredbe.

7. Zaključak

Svakim danom se povećava broj kibernetičkih napada kao i razvoj novih prijetnji po sustav, stoga zaštita informacijskog sustava postaje imperativ za sve organizacije. Osim računalnih i mrežnih sustava ciljane mete kibernetičkih napada su i osobe kao i nove vrste tehnologija, poput IoT uređaja. Sve veća prisutnost novih uređaja zahtijeva nove mehanizme zaštite, a jedan od primjera je i korištenje umjetne inteligencije prilikom zaštite sustava. Potrebno je naglasiti važnost sigurnosti te predočiti osobama i organizacijama određene mehanizme zaštite kako bi uvelike osigurali svoju imovinu. Statistički podaci u ovom radu prikazuju kako su najveće interesne skupine kibernetičkih napada organizacije od velikih značaja, poput vladinih organizacija i zdravstva. Zbog toga je potrebna preventivna zaštita sustava korištenjem sigurnosnog mehanizma kao što je penetracijsko testiranje. Penetracijski testovi zahtijevaju stručne certificirane osobe koje će svojim znanjem simulirati napad s ciljem zaštite informacijskog sustava. Razvojem novih tehnologija, ujedno i novih prijetnji stvara se potreba za konstantnim unaprjeđenjem znanja ispitivača kako bi u što većoj mjeri mogao zaštiti sustav. Svrha metodologije penetracijskog testiranja je pružiti ispitivaču uvid u korake i postupke prilikom izvođenja testiranja. Standardizacija i pravni aspekti nisu u potpunosti regulirani, a taj problem se nastoji riješiti korištenjem standardizirane metodologije. Pozitivna strana standardizacije je što se u posljednje vrijeme daje veliki značaj u odabiru ispravne metodologije s kvalitetnim alatima i metodama u cilju pružanja kvantitativne i opsežne zaštite informacijskog sustava.

Svaki sustav podložan je ranjivostima i ne može se u potpunoj mjeri zaštiti, međutim to nije razlog da se sustavi ne postave na maksimalnu razinu sigurnosti. Primjer penetracijskog testiranja izvedeno je u ovom radu, a za uspješnost provođenja bilo je potrebno koristiti metodologiju i automatizirane alate. Alati su opisani kroz rad, a njihova primjena prikazana je kroz napad nad virtualnim okruženjem. Automatizirani alati omogućili su prikupljanje podataka i mapiranje mreže ciljnog sustava te u fazi penetracije sustava koristili su se kao vektori napada. Praktični dio rada uspješno je izveden te je pronašao ranjivosti koji su došli do udaljenog pristupa sustavu. Ispitivači nakon završetka testiranja nude organizacijama izvještaj i prijedloge za rješavanje sigurnosnih propusta, a takav pristup proveden je i u ovom radu. Do ranjivosti sustava došlo je zbog pogrešne konfiguracije i nepotpunog ažuriranja sustava, a različita rješenja ranjivosti nude se kroz kvalitetniju konfiguraciju vatrozida, primjenom kontrole pristupa, korištenjem najnovijih inačica sustava te zaštitom od SQLi napada korištenjem parametriziranih upita i provjere unosa. Penetracijsko testiranje od velikog je značaja prilikom pružanja sigurnosnih mehanizama i njegova primjena sve je učestalija u svijetu. U budućnosti, potrebno je provoditi testiranje s oprezom kako ne bi došlo do negativnih posljedica po organizaciju, stoga je potrebno pristupati testiranju u skladu s metodologijom penetracijskog testiranja.

LITERATURA

[1] Cvitić I, Peraković D, Periša M, Botica M. An Overview of the Cyber Security Strategic Management in Republic of Croatia, RCITD - Proceedings in Research Conference in Technical Disciplines, Mokrys, Michal ; Badura, Stefan ; Peraković, Dragan (ur.). Zilina, Slovakia: EDIS - Publishing Institution of the University of Zilina, 2017. str. 13-18 doi:10.18638/rcitd.2017.5.1

[2] Weidman G. Penetration Testing: A Hands-On Introduction to Hacking 1st Edition. No Starch Press; 2014

[3] Peraković D, Cvitić I, Kuljanić T, Brletić L. Analysis of Wireless Routers Vulnerabilities Applied in the Contemporary Networks. Proceedings of The 6th International Virtual Research Conference in Technical Disciplines (RCITD-2018), Mokrys, Michal ; Badura, Stefan ; Peraković, Dragan (ur.). Zilina: Publishing Society, 2018. str. 31-37 doi:10.18638/rcitd.2018.6.1.123

[4] Gupta B, Tewari A, Cvitić I, Peraković D, Chang X. Artificial intelligence empowered emails classifier for Internet of Things based systems in industry 4.0. Wireless networks (2021) doi:10.1007/s11276-021-02619-w

[5] The Canadian Institute of Chartered Accountants Information Technology Advisory Committee. Using an Ethical hacking Technique to Assess Information Security Risk, Toronto, 2003.

[6] Statista. Preuzeto sa: https://www.statista.com/statistics/267132/total-damage-caused-by-by-cyber-crime-in-the-us/ [Pristupljeno: srpanj 2021.].

[7] Ptsecurity. Preuzeto sa: https://www.ptsecurity.com/ww-en/analytics/cybersecurity-threatscape-2018/ [Pristupljeno: lipanj 2021.].

[8] Appiah JK. Network and Systems Security Assessment using penetration testing in a university environment: The case of Central University College. Kwame Nkrumah Universityy of science and technology, 2014.

[9] Infosec. Preuzeto sa: https://resources.infosecinstitute.com/topic/what-are-black-box-grey-box-and-white-box-penetration-testing/ [Pristupljeno: lipanj 2021.].

[10] Securityboulevard. Preuzeto sa: https://securityboulevard.com/2021/04/what-is-black-box-grey-box-and-white-box-penetration-testing/ [Pristupljeno: lipanj 2021.].

[11] Utor. Preuzeto sa: https://u-tor.com/topic/white-box-penetration-testing [Pristupljeno: lipanj 2022.].

[12] Federal Office for Information Security. A Penetration Testing Model. Preuzeto sa: www.bsi.bund.de [Pristupljeno: lipanj 2021.].

[13] PCI Security Standards Council. Information Supplement: Penetration Testing Guidance, 2017.

[14] Purplesec. Preuzeto sa: https://purplesec.us/firewall-penetration-testing/ [Pristupljeno srpanj 2021.].

[15] Janić, D., Peraković, D. & Remenar, V. An Analysis of Wireless Network Security in the City of Zagreb and the Zagreb and Karlovac Counties. U: 7th International Conference on Ports and Waterways – POWA 2012, Zagreb: Fakultet prometnih znanosti, 2012. str. 216-223

[16] He-Jun L, Yang Y. Research on WiFi Penetration Testing with Kali Linux, 2021.

[17] Cvitić, I., Peraković, D., Periša, M. & Jurcut, A. Methodology for Detecting Cyber Intrusions in e-Learning Systems during COVID-19 Pandemic. *Mobile networks & applications* (2021) doi:10.1007/s11036-021-01789-3.

[18] Utwente. Preuzeto sa: https://research.utwente.nl/en/publications/physical-penetration-testing-a-whole-new-story-in-penetration-tes [Pristupljeno: svibanj 2021.].

[19] Geer D, Harthorne J. Penetration testing: a duet, Annual Computer Security Applications Conference. 2002;18(1):185-195.

[20] Hadnagy C, Wilson P. Social Engineering: The Art of Human Hacking, WileyPublishing, Indiana, 2010.

[21] Peraković D., Cvitić I.: Sigurnost i zaštita informacijsko komunikacijskog sustava, Fakultet prometnih znanosti, Zagreb, 2020., nastavni tekst, publicirano u digitalnom obliku na Internet poslužitelju na adresi: https://moodle.srce.hr/

[22] Agarwal A. International Journal of Computer Applications in Engineering Sciences. The Security Risks Associated with Cloud Computing. 2011;1: 257-259.

[23] Geric S, Hutinski Z. Information system security threats classifications. Journal of Information and Organizational Sciences, 2007;31: 51.

[24] Jouini M, Rabai L, Aissa A. Classification of security threats in information systems. ScienceDirect, 2014;32: 489-496.

[25] Icove D, Seger K, VonStroch W. Computer Crime: A Crimefighter's Handbook, O'Reilly Media; 1995.

[26] Boban M. Biometrija u sustavu sigurnosti, zaštite i nadzora informacijskih sustava. Zbornik radova Veleučilišta u Šibeniku, 2015;1: 115-148.

[27] Zakon HR. Preuzeto sa: https://www.zakon.hr/z/218/Zakon-o-informacijskoj-sigurnosti [Pristupljeno: srpanj 2021.].

[28] Gudac N. Sigurnosni aspekt informacijskih sustava. Veleučilište u Karlovcu, 2019.

[29] Baloch R. Ethical Hacking and Penetration Testing Guide. Auerbach Publications, 2014.

[30] Nmap. Preuzeto sa: https://nmap.org/book/port-scanning.html [Pristupljeno: kolovoz 2021.].

[31] Parvin A, Ashikali H. Seven Phrase Penetration Testing Model. International Journal of Computer Applications, 2012;59(5): 16-20.

[32] Beyondtrust. Preuzeto sa: https://www.beyondtrust.com/blog/entry/privilege-escalation-attack-defense-explained [Pristupljeno: srpanj 2021.].

[33] CARNet. Metodologija penetracijskog testiranja, 2008; 219(2).

[34] Engebretson P. The Basics of Hacking and Penetration Testing: Ethical Hacking and Penetration Testing Made Easy 2nd Edition. Syngress Publishing, 2011.

[35] Herzog P. The Open Source Security Testing Methodology Manual: Contemporary Security Testing and Analysis. ISECOM, 2010.

[36] Curphey M, Cuthbert D, van der Stock A, Shields L. The OWASP Testing Project. The OWASP Foundation, 2004.

[37] OWASP Web Application Penetration Checklist. Preuzeto sa: https://csrc.nist.gov/publications/sp#sp800-30 [Pristupljeno: kolovoz 2021.].

[38] Scarfone K, Souppaya M, Cody A, Orebaugh A. Technical Guide to Information Security Testing and Assessment. Recommendations of the National Institute of Standards and Technology, 2008.

[39] Hertzog R, O'Gorman J, Aharoni M. Kali Linux Revealed. Mastering the Penetration Testing Distribution, 2017.

[40] Thacker BH, Riha DS, Fitch S, Huyse LJ, Pleming JB. Probabilistic engineering analysis using the Nessus software. Southwest Research Institute, Reliability and Materials Integrity, 2006;28: 83-107

[41] Anderson H. Introduction to Nessus. SecurityFocus Printable INFOCUS, 2003.

[42] Networkworld. Preuzeto sa: https://www.networkworld.com/article/3296740/what-is-nmap-why-you-need-this-network-mapper.html [Pristupljeno: kolovoz 20221.].

[43] Nmap. Preuzeto sa: https://nmap.org/book/man-nse.html [Pristupljeno: kolovoz 2021.].

[44] CARNet. Analiza NMAP alata, 2006;147(1).

[45] Sqlmap. Preuzeto sa: https://sqlmap.org/ [Pristupljeno: kolovoz 2021.].

[46] Div0. Preuzeto sa: https://www.div0.sg/post/netcat [Pristupljeno: kolovoz 2021.].

[47] Github. Preuzeto sa: https://github.com/OWASP/joomscan [Pristupljeno: kolovoz 2021.].

[48] Varonis. Preuzeto sa: https://www.varonis.com/blog/john-the-ripper/ [Pristupljeno: kolovoz 2021.].

[49] GTFOBins. Preuzeto sa: https://gtfobins.github.io/gtfobins/yum/ [Pristupljeno: kolovoz 2021.].

[50] w3. Preuzeto sa: https://www.w3.org/Protocols/rfc2616/rfc2616-sec15.html [Pristupljeno: srpanj 2021.].

[51] Zadro P. HTTPS protokol, Filozofski fakultet odsjek za informacijske i komunikacijske znanosti. Zagreb, 2017.

[52] Venafi. Preuzeto sa: https://www.venafi.com/blog/best-practices-ssh-key-management-what-are-your-ssh-security-risks [Pristupljeno: kolovoz 2021.].

[53] Mysql. Preuzeto sa: https://dev.mysql.com/doc/mysql-security-excerpt/8.0/en/security-guidelines.html [Pristupljeno: kolovoz 2021.].

[54] Tenable. Preuzeto sa: https://www.tenable.com/plugins/nessus/70658 [Pristupljeno: kolovoz 2021].

POPIS KRATICA

ENISA - European Network and Information Security Agency

IoT – Internet of Things

XSS – Cross Site Scripting

SQLi – SQL injection

DoS – Denial of Service

ACL - Access Control List

IP – Internet Protocol

SSH – Secure Shell

IDS – Intrusion Detection System

IPS - Intrusion Preventions System

IC3 - Internet Crime Complaint Center

CIA – Confidentiality Integrity Availability

CERT - Computer Emergency Response Team

VPN – Virtual Private Network

TCP - Transmission Control Protocol

UDP - User Datagram Protocol

CA - Code analysis

VA - Vulnerabilty analysis

TA - Total analysis

CVE - Common Vulnerabilities and Exposures

UAC - User Account Control

OSSTMM - Open Source Security Testing Methodology Manual

OWASP - Open Web Application Security Project

NIST - National Institute of Standards and Technology

OpSec – Operation Security

STAR - Security Test Audit Report

GUI - Graphical User Interface

NASL - Nessus Attack Scripting Language

SCTP - Stream Control Transmission Protocol

ICMP - Internet Control Message Protocol

CMS - Content Management System

NSE - Nmap Scripting Engine

JTR – John The Ripper

HTTP – HyperText Transfer Protocol

MITM – Man In The Middle

HTTPS - HyperText Transfer Protocol Secure

SSL - Secure Socket Layer

CBC - CBCher Block Chaining

YUM - The Yellowdog Updater Modified

POPIS SLIKA

Slika 1. Klasifikacija penetracijskog testiranja 8
Slika 2. Proces izvođenja mrežnih penetracijskih testova 13
Slika 3. Provođenje penetracijskog testiranja aplikacija 15
Slika 4. Klasifikacija prijetnji 18
Slika 5. Proces izvođenja penetracijskog testiranja 21
Slika 6. OWASP metodologija penetracijskog testiranja 28
Slika 7. Prikaz korisničkog sučelja DirbBuster alata 36
Slika 8. Izgled web stranice ranjivog sustava 39
Slika 9. Rezultati skeniranja Nmap alatom 40
Slika 10. Rezultati pretraživanja JoomScan alatom 41
Slika 11. Rezultati identificiranja ranjivosti 41
Slika 12. Rezultat pronalaska baza podataka korištenjem sqlmap alata 42
Slika 13. Cjelokupni rezultat provedbe napada SQLmap alatom 43
Slika 14. Rezultat razbijanja hash zapisa i pronalazak lozinke 43
Slika 15. Izmjena error.php koda web stranice s php reverse shell-om 44
Slika 16. Prikaz uspješno pokrenutog shell-a i stvaranje stabilne konekcije 45
Slika 17. Prikaz procesa preuzimanja i promjena ovlasti nad Linpeas skriptom 45
Slika 18. Rezultat korištenja Linpeas skripte i pronalazak lozinke 46
Slika 19. Prikaz ostvarivanja root pristupa 46
Slika 20. Rezultati skeniranja Nessus alatom 50
Slika 21. Rezultati korištenja Jooblah skripte 51

POPIS TABLICA

Tablica 1. Najučestaliji TCP i UDP portovi 23
Tablica 2. Vjerojatnost vektora napada nad operativnim sustavima 25
Tablica 3. Prikaz najučestalijih sintaksi Nmap alata 34
Tablica 4. Razlika između HTTP i HTTPS protokola 48

POPIS GRAFIKONA

Grafikon 1. Financijska šteta prouzročena kibernetičkim napadima (2001.-2020.) 5
Grafikon 2. Učestalost kibernetičkih napada nad organizacijama 6
Grafikon 3. Prikaz najčešćih vrsta napada i ciljnih sustava 7

Sveučilište u Zagrebu
Fakultet prometnih znanosti
10000 Zagreb
Vukelićeva 4

IZJAVA O AKADEMSKOJ ČESTITOSTI I SUGLASNOST

Izjavljujem i svojim potpisom potvrđujem kako je ovaj diplomski rad isključivo rezultat mog vlastitog rada koji se temelji na mojim istraživanjima i oslanja se na objavljenu literaturu što pokazuju korištene bilješke i bibliografija.

Izjavljujem kako nijedan dio rada nije napisan na nedozvoljen način, niti je prepisan iz necitiranog rada, te nijedan dio rada ne krši bilo čija autorska prava.

Izjavljujem također, kako nijedan dio rada nije iskorišten za bilo koji drugi rad u bilo kojoj drugoj visokoškolskoj, znanstvenoj ili obrazovnoj ustanovi.

Svojim potpisom potvrđujem i dajem suglasnost za javnu objavu diplomskog rada pod naslovom **Metodologija penetracijskog testiranja**

na internetskim stranicama i repozitoriju Fakulteta prometnih znanosti, Digitalnom akademskom repozitoriju (DAR) pri Nacionalnoj i sveučilišnoj knjižnici u Zagrebu.

U Zagrebu, 9/10/2021

Student/ica:

(potpis)