Analiza ranjivosti jezgre Linux operativnog sustava kroz simulaciju kibernetičkih napada

Pribanić, Marija

SVEUČILIŠTE U ZAGREBU

FAKULTET PROMETNIH ZNANOSTI

Marija Pribanić

ANALIZA RANJIVOSTI JEZGRE LINUX OPERATIVNOG SUSTAVA KROZ SIMULACIJU KIBERNETIČKIH NAPADA

DIPLOMSKI RAD

Zagreb, 2022.

SVEUČILIŠTE U ZAGREBU
FAKULTET PROMETNIH ZNANOSTI
POVJERENSTVO ZA DIPLOMSKI ISPIT

Zagreb, 6. lipnja 2022.

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

DIPLOMSKI ZADATAK br. 6973

Pristupnik: **Marija Pribanić (0036481337)**
Studij: Promet
Smjer: Informacijsko-komunikacijski promet

Zadatak: **Analiza ranjivosti jezgre Linux operativnog sustava kroz simulaciju kibernetičkih napada**

Opis zadatka:

U okviru diplomskog rada potrebno je pružiti pregleda relevantnih istraživanja u području sigurnosti Linux operativnog sustava. Nadalje potrebno je analizirati sigurnosne prijetnje i ranjivosti Linux sustava te ih kategorizirati prema stupnju rizika. Potrebno je dizajnirati simulaciju kibernetičkih napada na Linux sustava iskorištavanjem prethodno identificiranih ranjivosti. Konačno, potrebno je predložiti smjernice unaprjeđenja razine sigurnosti Linux sustava temeljem analiza rezultata prethodno provedenih simulacija.

Mentor:

dr. sc. Ivan Cvitić

Predsjednik povjerenstva za diplomski ispit:

Sveučilište u Zagrebu

Fakultet prometnih znanosti

DIPLOMSKI RAD

ANALIZA RANJIVOSTI JEZGRE LINUX OPERATIVNOG SUSTAVA KROZ SIMULACIJU KIBERNETIČKIH NAPADA

CYBER ATTACKS SIMULATION FOR VULNERABILITIES ANALYSIS OF LINUX KERNEL

Mentor: dr. sc. Ivan Cvitić

Student: Marija Pribanić
JMBAG: 0036481337

Zagreb, rujan 2022.

ANALIZA RANJIVOSTI JEZGRE LINUX OPERATIVNOG SUSTAVA KROZ SIMULACIJU KIBERNETIČKIH NAPADA

SAŽETAK

U ovom diplomskom radu proveden je pregled i analiza ranjivosti Linux operativnog sustava. Ranjivosti jezgre Linux-a mogu omogućiti zlonamjernim korisnicima da zaobiđu sve zaštitne mehanizme jezgre i ugroze cijeli sustav. U svrhu prikaza eksploatacije ranjivosti Linux operativnog sustava, provedena je simulacija kibernetičkih napada. Temeljem rezultata provedenih simulacija predložene su smjernice poboljšanja sigurnosti sustava.

KLJUČNE RIJEČI: Linux operativni sustav; jezgra; ranjivost; kibernetički napad; eksploatacija

CYBER ATTACKS SIMULATION FOR VULNERABILITIES ANALYSIS OF LINUX KERNEL

SUMMARY

In this master thesis, an overview and analysis of the Linux operating system vulnerabilities were carried out. Vulnerabilities in the Linux kernel can allow malicious users to bypass all kernel protection mechanisms and compromise the entire system. In order to demonstrate the exploitation of the vulnerabilities of the Linux operating system, a simulation of cyber attacks was carried out. Based on the results of the simulations, guidelines for improving system security were proposed.

KEY WORDS: Linux operating system; kernel; vulnerability; cyber attack; exploit

SADRŽAJ

1. Uvod

Linux je jedan od najpopularnijih operativnih sustava na svijetu. Zahvaljujući svojoj proširivosti i prirodi otvorenog koda, može se koristiti za širok raspon različitih sustava, od radijski upravljanog modela helikoptera, preko mobilnih telefona, do većine najvećih superračunala na svijetu. Međutim, kada je nešto otvorenog koda, upravljanje time može često biti problematično. Zbog velike rasprostranjenosti na uređajima diljem svijeta, sve su učestalije kibernetičke prijetnje i napadi na Linux operativni sustav.

Kibernetičke prijetnje bilježe kontinuirani porast na globalnoj razini, a različite vrste napada u kibernetičkom prostoru postaju sve sofisticiranije i složenije te utječu na svakodnevni život i poslovanje. Ranjivosti operativnog sustava i same jezgre mogu se iskoristiti za dobivanje pristupa sustavu i vršenje raznih zlonamjernih radnji poput krađe povjerljivih informacija. Upravo iz tog razloga vrlo je bitna svijest o mogućim kibernetičkim ugrozama i kako se od njih zaštititi.

Ovaj diplomski rad sastoji se od osam poglavlja:

1. Uvod
2. Pregled dosadašnjih istraživanja
3. Arhitektura Linux operativnog sustava
4. Analiza ranjivosti Linux operativnog sustava
5. Metodologija provedbe istraživanja
6. Simulacija kibernetičkih napada na jezgru Linux operativnog sustava
7. Analiza rezultata istraživanja i prijedlozi unaprjeđenja razine sigurnosti Linux operativnog sustava
8. Zaključak

U drugom poglavlju predstavljen je pregled dosadašnjih istraživanja ranjivosti jezgre Linux operativnog sustava. Mnogo je istraživanja provedeno u proučavanje ranjivosti jezgre te smanjivanje mogućnosti iskorištavanja ranjivosti, što je u ovom poglavlju opisano.

U trećem poglavlju opisana je arhitektura Linux operativnog sustava. Prikazane su njegove najbitnije komponente te podsustavi od kojih je građen. Detaljno su opisane funkcionalnosti

svakog od podsustava. Nabrojane su najbitnije značajke Linux operativnog sustava i objašnjen je pojam Linux distribucija.

U četvrtom poglavlju je provedena analiza ranjivosti Linux operativnog sustava. Objašnjen je pojam ranjivosti te kako se one klasificiraju zatim su objašnjene ranjivosti same jezgre Linux-a te su sustavno kategorizirane.

Peto poglavlje predstavlja metodologiju provedbe istraživanja. Sustavno su prikazani i objašnjeni alati pomoću kojih je provedeno istraživanje.

U šestom poglavlju prikazan je postupak provedbe kibernetičkih napada realiziranih eksploatacijom ranjivosti Metasploitable sustava. Provedena je analiza ranjivosti sustava pomoću OpenVAS alata te na temelju identificiranih ranjivosti, simulirani su kibernetički napadi na sustav primjenom alata dostupnih u Kali Linux-u.

U sedmom poglavlju provedena je analiza prethodno izvedenih simulacija napada te su izneseni prijedlozi unaprjeđenja razine sigurnosti Linux operativnog sustava u svrhu zaštite od kibernetičkih napada na temelju provedenog istraživanja.

Osmo poglavlje daje pregled svih saznanja i informacija stečenih izradom ovog rada u vidu jedinstvenog i subjektivnog zaključka.

2. Pregled dosadašnjih istraživanja

Linux je obitelj od tisuće operativnih sustava (engl. *operating system* – OS) temeljenih na Linux jezgri. Jezgra Linux-a pokreće mnoge uređaje: računala, mobilne telefone, tablete, pametne satove, kućanske uređaje, automobile, televizije. Velika većina Linux OS-ova (ili Linux distribucija) instalirana je na poslužiteljima diljem svijeta. Koristi se na web poslužiteljima, superračunalima i radnim računalima burzi, vladinih organizacija, obrazovnih institucija, znanstvenih institucija i velikih tehničkih kompanija kao što su Google, Facebook ili Apple.

Za razliku od Microsoft Windows OS-a, distribucije Linux-a informatičkim inženjerima nude potpunu kontrolu nad poslom koji obavljaju, i mogućnost prilagođavanja svakog dijela koda prema potrebama. Različite napredne tehnologije mogu se učinkovito implementirati na Linux zbog njegove stabilnosti, visoke učinkovitosti i prirode otvorenog koda. Međutim, sam Linux OS i neke od usluga temeljenih na Linux-u neizbježno imaju svoje ranjivosti. Ranjivosti u samoj jezgri mogu dopustiti zlonamjernim korisnicima da zaobiđu sve mehanizme zaštite jezgre i ugroze sustav. Zbog toga je važno pratiti znanstvena istraživanja i radove kako bi se mogli bilježiti trendovi i davati preporuke sigurnosnih rješenja. Mnogo je istraživanja provedeno u ublažavanje ranjivosti jezgre i što je još važnije, smanjivanje mogućnosti iskorištavanja ranjivosti. Osim toga postoje i mnogobrojni alati koji pokušavaju riješiti pitanje sigurnosti.

U radu Shuangxia, N., Jiansong, M., Zhigang, Z., Zhuo, L. *Overview of Linux Vulnerabilities* dan je pregled ranjivosti Linux-a, koje potječu iz samog Linux OS-a. Ranjivosti su podijeljene u tri vrste prema posljedicama uzrokovanim iskorištavanjem tih ranjivosti, a to su ranjivost eskalacije ovlasti, ranjivost uskraćivanja usluge te ranjivost IP lažiranja. U radu je predstavljena metodologija poboljšanja sigurnosti Linux-a koja se može podijeliti u dvije glavne skupine na temelju učvršćivanja OS-a i proširene kontrole pristupa.

U radu Dmitry, M., Elena, P., 2020. *Linux Privilege Increase Threat Analysis* autori analiziraju sigurnosne probleme Linux OS-a povezane s povećanjem ovlasti neovlaštenih korisnika, uslijed čega napadač može dobiti potpunu kontrolu nad OS-om. U tom je smislu vrlo bitna analiza prijetnji sigurnosti Linux OS-a. Zbog činjenice da je Linux višekorisnički OS, osiguran je mehanizam za diskrecijsko razgraničenje prava pristupa. Ovaj mehanizam implicira da svaka datoteka ili direktorij u sustavu ima vlasnika korisnika i vlasničku grupu. Također, svaka

datoteka ili direktorij ima tri grupe prava pristupa: vlasnik, grupa vlasnika i svi ostali korisnici OS-a. Svaka grupa ima prava čitanja, pisanja i izvršavanja. Ovaj mehanizam razgraničenja ovlasti omogućuje pojedinačnim korisnicima sustava stvaranje osobnih datoteka i direktorija koji su nedostupni drugim korisnicima. U radu su razmatrani drukčiji autorizacijski pristupi u Linux-u te njihove prednosti i nedostaci.

U radu Alam, D., Zaman, M., Farah, T., Rahman R., Hosain, M. S., 2017. *Study of the Dirty Copy on Write, a Linux Kernel memory allocation vulnerability* se istražuju tehnike iskorištavanja *Dirty COW* ranjivosti te njen utjecaj na poslužitelje s Linux OS-om. *Dirty Copy on Write* također poznat kao *Dirty COW* ranjivost je ranjivost jezgre Linux-a koja je utjecala na sve OS-e temeljene na Linux jezgri. Ova ranjivost omogućuje napadačima da eskaliraju zaštitu datotečnog sustava Linux jezgre, dobiju *root* ovlasti te tako ugroze cijeli sustav. Autori provode analizu nad skupom podataka koji opisuju utjecaj napada iskorištavanjem ove ranjivosti na poslužiteljima u Bangladešu.

Rad Shameli-Sendi, A. *Understanding Linux kernel vulnerabilities* izvješće je o analizi 1858 ranjivosti jezgre Linux-a koje su pronađene u razdoblju od siječnja 2010. godine do siječnja 2020. godine. Ranjivosti su klasificirane s gledišta napadača na sustav, koristeći različite kriterije kao što su:

- Vrsta napada- analiza ranjivosti pokazuje da velik broj njih utječe na kriterije dostupnosti koji dovode do napada uskraćivanja usluge (engl. *Denial of Service* – DoS)
- Ciljani podsustavi jezgre- ova kategorija označava komponente jezgre Linux-a koje su na meti napadača kao što su *arch, net, fs, crypto*
- Porijeklo napada- lokacija odakle se ranjivosti mogu iskoristiti (lokalno ili udaljeno),
- Utjecaj napada na povjerljivost, cjelovitost i dostupnost sustava te
- Razina složenosti izvođenja napada.

U zaključku je navedeno kako otkrića ukazuju na prisutnost velikog broja ranjivosti niske razine složenosti. Većina njih može se iskoristiti iz lokalnog sustava, što dovodi do napada koji mogu ozbiljno ugroziti kvalitetu usluge jezgre te omogućiti napadačima da dobiju ovlašteni pristup.

3. Arhitektura Linux operativnog sustava

Linux je razvio Linus Torvalds 1991. godine i pustio u distribuciju kao OS otvorenog koda. Linux je izveden iz Unix-a i nastavak je osnove Unix dizajna. Unix i OS-i njemu slični su obitelj računalnih OS-a koji potječu od originalnog Unix sustava iz Bell Labs-a koji se može pratiti unatrag do 1965. godine. Unix je obitelj višezadaćnih, prijenosnih, višekorisničkih računalnih OS-a, koji ujedno imaju konfiguracije dijeljenja vremena. Unix sustavi koriste centraliziranu OS jezgru koja je odgovorna za upravljanje cijelim sustavom. Linux distribucije najpoznatiji su i najzdraviji primjer izravnih izvedenica Unix-a, [1].

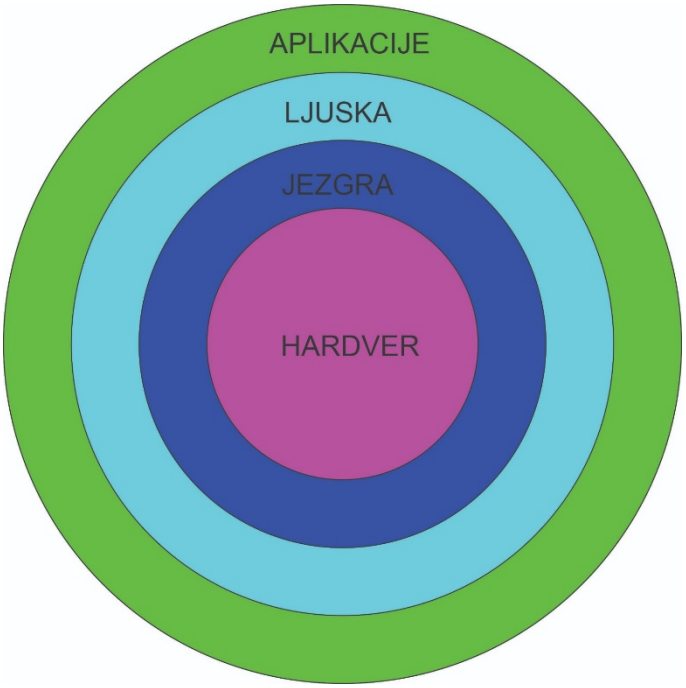
Pojam Linux ima dva različita značenja. Iako se obično koristi kao naziv za cijeli OS, Linux je samo naziv jezgre, dijela OS-a zaduženog za upravljanje interakcijama između hardvera i korisničkih aplikacija. Izraz Linux distribucija, s druge strane, odnosi se na kompletan OS izgrađen na temelju Linux jezgre, obično uključujući instalacijske programe i mnoge aplikacije, koje su ili unaprijed instalirane ili pakirane tako da ih se može lako instalirati, [2]

Kod Linux jezgre u potpunosti je napisan od nule. Dizajnirana je na takav način da se ponaša kao Unix, ali nema izvorni Unix kod u sebi. Linux jezgra je licencirana pod GNU *General Public License*, koja određuje da se softver licenciran pod njom može mijenjati, prilagođavati, pa čak i prodavati, ali prilikom redistribucije tog istog softvera, ista prava moraju imati i drugi korisnici, [3].

Velika prednost Linuxa je njegov modularan dizajn. To znači da se na osnovne dijelove koda, naknadno, instaliraju samo oni moduli koji su potrebni za određene zadatke. Tako instalacija uvijek radi sigurno, brzo i stabilno, nakon inicijalnog postavljanja na neki uređaj. Otvoreni kod još je jedna velika prednost Linux-a. Time se otvara mogućnost prilagođavanja svakog dijela koda prema potrebama korisnika. Upravo je to i glavni razlog što je Linux toliko zastupljen u modernim uređajima kao što su pametni telefoni, moderni automobili, Smart TV uređaji i drugi periferni uređaji koji trebaju softver za rad, [4].

3.1. Komponente Linux operativnog sustava

Arhitektura Linuxa, prikazana na slici 1, sastoji se od jezgre, ljuske i aplikacijskih programa koji predstavljaju softver, [5].



Slika 1 Opća arhitektura Linux-a

Izvor: [5]

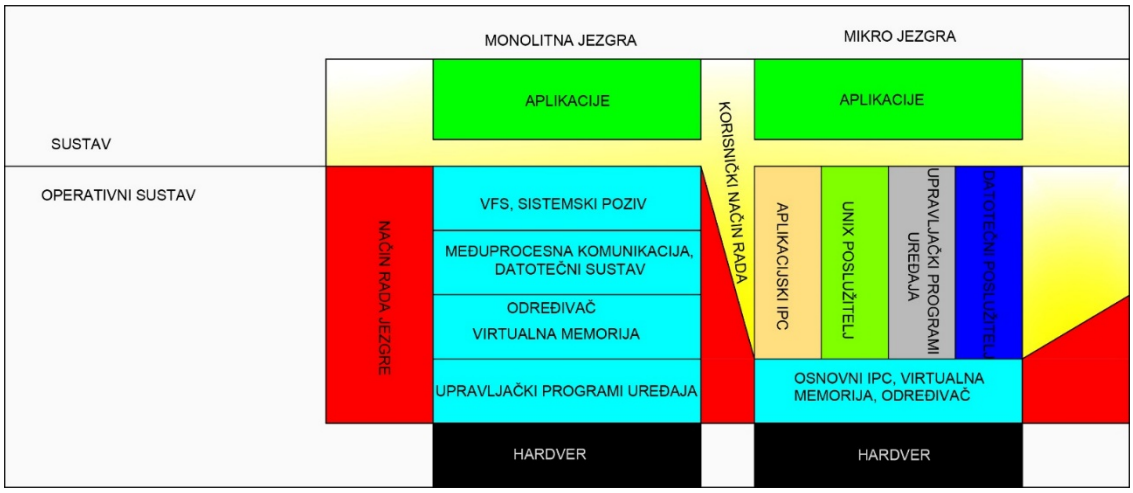
- **Hardver** predstavlja fizičke dijelove računala, kao što su središnja procesorska jedinica (engl. *central processing unit,* CPU), memorija, monitor, miš, tipkovnica, tvrdi disk i drugi uređaji povezani s CPU-om.
- **Jezgra** (engl. Kernel) je računalni program koji je središnji, sastavni dio svakog OS-a. Ona upravlja operacijama hardvera (procesor, memorija, i periferni uređaji), te njihovoj međusobnoj komunikaciji. Linux jezgra je pisana u programskom jeziku C i asembleru.
- **Ljuska** (engl. Shell) je sučelje između korisnika i jezgre. To je okruženje u kojemu se mogu pokretati naredbe, programi i skripte ljuske, to jest korisničko sučelje za pristup uslugama OS-a kao što su izvršavanje programa korisničkog sučelja, manipulacija datotečnim sustavom, ulazno/ izlazne operacije, komunikacija, dodjela resursa, otkrivanje grešaka, sigurnost i zaštita.
- **Aplikacije** su programi koji korisniku daju većinu funkcionalnosti operacijskog sustava.

3.1.1. Jezgra

Jezgra je jedan od ključnih dijelova OS-a. Postoji mnoštvo Linux distribucija, ali postoji samo jedna stvar koja im je zajednička, a to je Linux jezgra. Ona je odgovorna za svaku od glavnih akcija Linux OS-a. Postoji nekoliko važnih vrsti jezgra koje su navedene u nastavku, [1]:

- Monolitna jezgra (engl. *Monolithic kernel*)
- Mikro jezgra (engl. *Microkernel*)
- Hibridna jezgra (engl. *Hybrid kernel*)- kombinacija monolitne i mikro jezgre

Arhitekture monolitne i mikro jezgre prikazane su na slici 2, te su na njoj naznačene najbitnije razlike.



Slika 2 Vrste arhitektura jezgri OS-a

U monolitnoj arhitekturi jezgre, cijeli OS radi u jednom jezgrenom prostoru. Samostalno definira virtualno sučelje visoke razine povrh računalnog hardvera. Linux jezgra, kao i većina Unix jezgri, je monolitna: svaki sloj jezgre je integriran u cijeli program jezgre i radi u načinu rada jezgre u ime trenutnog procesa.

U mikro arhitekturi jezgre, osnovne usluge OS-a pokreću se u jednom procesu, dok se ostale usluge pokreću u različitim procesima. U način rada jezgre, uključena je gotovo minimalna količina mehanizama, a to su osnovna komunikacija među procesima IPC (engl. *inter-process*

communication), raspoređivanje i upravljanje adresnim prostorom niske razine. Što se tiče veličine izvornog koda, mikro jezgra je općenito manja od monolitne jezgre, [6].

Iako su akademska istraživanja OS-a usmjerena na mikro jezgre, takvi OS-i općenito su sporiji od monolitnih. Međutim, OS-i koji koriste mikro arhitekturu jezgre mogu imati neke teoretske prednosti u odnosu na monolitne. Mikro jezgre prisiljavaju sistemske programere da usvoje modularni pristup, jer je svaki sloj OS-a relativno neovisan program koji mora komunicirati s ostalim slojevima kroz dobro definirana i čista softverska sučelja. Štoviše, postojeći OS s mikro jezgrom može se lako prenijeti na druge arhitekture, jer su sve komponente ovisne o hardveru obično enkapsulirane u kodu jezgre. Konačno, OS-i s mikro jezgrom imaju tendenciju boljeg iskorištavanja memorije s izravnim pristupom (engl. *Random access memory* - RAM) od monolitnih, jer sistemski procesi koji ne implementiraju potrebne funkcije mogu biti zamijenjeni ili izbačeni, [6].

Jezgra je zadužena za upravljanje zadacima u četiri opća područja sustava, [7]:

- Procesi – jezgra je odgovorna za određivanje kojim procesima je dopušteno koristiti CPU.
- Memorija – jezgra mora pratiti svu memoriju, što je trenutno dodijeljeno određenom procesu, što se može dijeliti između procesa i što je slobodno.
- Upravljački programi uređaja – jezgra djeluje kao sučelje između hardvera i procesa. Obično je posao jezgre da upravlja hardverom.
- Sistemski pozivi i podrška – procesi obično koriste sistemske pozive za komunikaciju s jezgrom.

Gotovo sve što jezgra radi vrti se oko glavne memorije. Jedan od zadataka jezgre je podijeliti memoriju na mnoge potpodjele i održavati određene informacije o stanju tih potpodjela u svakom trenutku. Svaki proces dobiva svoj vlastiti udio memorije, a jezgra mora osigurati da svaki proces zadrži svoj udio.

Postoji znatna razlika između načina na koji se jezgrini i korisnički procesi izvode: jezgra radi u načinu rada jezgre, a korisnički procesi se izvode u korisničkom načinu rada. Kod koji radi u načinu rada jezgre ima neograničen pristup procesoru i glavnoj memoriji. Ovo je moćna, ali opasna

ovlast koja omogućuje procesu da lako sruši cijeli sustav. Područje kojem samo jezgra može pristupiti naziva se jezgrin prostor, [7].

S druge strane, korisnički način rada ograničava pristup (obično vrlo malom) podskupu memorije i sigurnim CPU operacijama. Korisnički prostor odnosi se na dijelove glavne memorije kojima korisnički procesi mogu pristupiti. Ako proces pogriješi i sruši se, posljedice su ograničene i jezgra ih može popraviti, [7].

3.1.2. Ljuska

U Linux-u i Unix-u, ljuska se odnosi na program koji se koristi za tumačenje unesenih naredbi. Omogućuje korisniku izvršavanje naredbi njihovim ručnim upisivanjem u terminalu ili automatski u programima koji se nazivaju skripte ljuske. Najbliža analogija u sustavu Windows je *Command Prompt*. Međutim, za razliku od Windows-a, Linux i Unix dopuštaju korisniku da odabere koju ljusku želi koristiti. U tablici 1 je dan popis najpopularnijih programa ljuske u Linux-u, [8]:

Tablica 1 Programi ljuske Linux OS-a

Akronim	Puni naziv	Opis
sh	The Bourne shell	Originalna ljuska u Unix-u
csh	The C shell	Bolja verzija od originalne
ksh	The Korn shell	Prvi napredniji program ljuske
bash	GNU Bourne-Again shell	Trenutno najčešći program ljuske

Bourne ljuska (engl. *The Bourne shell* - sh) nazvana je po svom programeru Stephenu Bourne-u iz *AT&T Bell Laboratories*, a izdana je 1977. godine u verziji Unix 7, izdanje koje je distribuirano fakultetima i sveučilištima. To je originalna Unix ljuska. Brza je i poželjna, ali joj nedostaju značajke za interaktivnu upotrebu poput mogućnosti prisjećanja prethodnih naredbi, ugrađene aritmetike i rukovanja logičkim izrazima. I dalje je popularna zadana ljuska za Unix sustave.

C ljuska (engl. *The C shell* - csh), kao što joj naziv može implicirati, dizajnirana je da omogući korisnicima pisanje skripti ljuske koristeći sintaksu vrlo sličnu onoj u programskom jeziku C. Uključuje značajke kao što su aliasi i prisjećanja prethodnih naredbi.

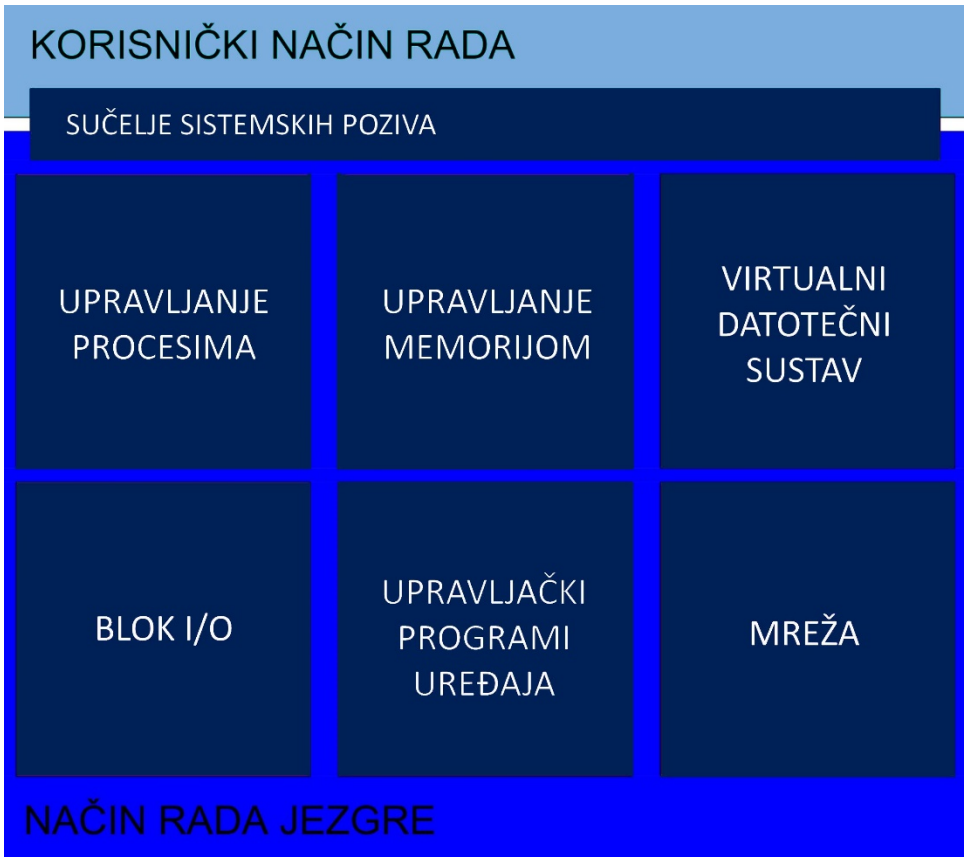
Korn ljuska (engl. The Korn shell - ksh) je Unix ljuska koju je razvio David Korn isto iz *AT&T Bell Laboratories* ranih 1980-ih. Kompatibilna je s Bourne ljuskom i također uključuje mnoge značajke C ljuske, poput prisjećanja prethodnih naredbi. Glavna prednost Korn u odnosu na tradicionalnu Unix ljusku je njegova upotreba kao programskog jezika. Uključuje mogućnost za programere da kreiraju nove naredbe ljuske prema potrebi, uz zadržavanje snažne kompatibilnosti s prethodnim verzijama ljuski.

Bourne Again ljuska (engl. GNU Bourne Again shell - bash) je ljuska napisana kao besplatna zamjena za standardnu Bourne ljusku. Ima sve njene značajke, uključujući dodatke koji olakšavaju programiranje i korištenje iz naredbenog retka. Budući da je besplatni softver, prihvaćen je kao zadana ljuska na većini Linux sustava.

Uz ljuske naredbenog retka, postoje i grafičke ljuske kao što su Windows Desktop, MacOS Finder ili Linux Gnome koje većini korisnika pojednostavljuju korištenje računala. Međutim, ove grafičke ljuske nisu zamjena za ljuske naredbenog retka za napredne korisnike koji žele izvršavati složene nizove naredbi više puta ili s parametrima koji nisu dostupni u prijateljskim, ali ograničenim grafičkim dijalozima i kontrolama, [9].

3.2. Podsustavi Linux operativnog sustava

Linux je podijeljen na nekoliko podsustava koji se bave specifičnim zadacima.



Slika 3 Pregled Linux podsustava

Izvor: [10]

Slika 3 predstavlja pregled najčešćih podsustava, a to su, [10]:

- Sučelje sistemskih poziva (engl. *system call interface*)
- Upravljanje procesima (engl. *process management*)
- Upravljanje memorijom (engl. *memory management*)
- Virtualni datotečni sustav (engl. *virtual file system - VFS*)
- Blok I/O (engl. *block I/O*)
- Upravljački programi uređaja (engl. *device drivers*)
- Mreža

U nastavku će biti opisano više o svakom pojedinom podsustavu.

3.2.1. Sučelje sistemskih poziva

Za povezivanje s korisničkim prostorom, Linux koristi koncept sistemskih poziva. To su definirane C funkcije, koje se interno prevode u specifične asemblerske pozive, kako bi se procesor doveo u način nadzora.

OS ne treba vjerovati nijednom programu koji se izvodi na procesoru, osim samom sebi. Sustav se stoga ponaša kao stražar koji s jedne strane ima potpunu kontrolu, a s druge strane ograničava pristup resursima i drugim procesima. Da bi se to postiglo, procesor mora hardverski podržavati nekoliko razina privilegija. OS tada obično radi na najvišoj razini, a sav ostali kod na nižim razinama. Međutim, ponekad programi (ili točnije procesi) moraju pristupiti resursima koje kontrolira OS, trebaju komunicirati s jezgrom. Za to je potrebno sučelje i uveden je koncept sistemskih poziva (engl. *System Calls* ili skraćeno *Syscalls*). Sistemski pozivi predstavljaju jasan, strogo definiran način za traženjem/ zahtijevanjem funkcija jezgre iz konteksta procesa, [10].

3.2.2. Upravljanje procesima

Temeljna zadaća OS-a je upravljanje procesima. Proces je program u izvođenju. Sastoji se od koda programa i od svakog resursa potrebnog za izvođenje. To uključuje memorijski adresni prostor, stanje procesora, otvorene datoteke kojima proces pristupa, signale i interne podatke jezgre.

OS pruža transparentan sloj apstrakcije između hardvera i programa, tj. svaki proces ima svoj virtualni procesor i virtualnu memoriju što znači da sa stajališta jednog procesa, on je jedini koji posjeduje resurse procesora. Izvršavanje više procesa u sustavu i prebacivanje između njih odvija se transparentno za svaki proces. Ovaj sloj apstrakcije osigurava da svaki proces ima kontrolirani pristup hardveru dok je izoliran od drugih procesa, [10].

3.2.3. Upravljanje memorijom

Baš kao i korisnički procesi, procesi jezgre trebaju memoriju za rad. No, unutar jezgre ne postoje unaprijed definirane funkcije koje se mogu pozvati za dodjelu i oslobađanje memorije. Dakle, jezgra mora definirati vlastitu funkcionalnost upravljanja memorijom tako da budu ispunjeni sljedeći uvjeti, [7]:

- Jezgra mora imati svoje vlastito područje u memoriji kojem korisnički procesi ne mogu pristupiti.
- Svaki korisnički proces treba svoj vlastiti dio memorije.
- Jedan korisnički proces ne smije pristupiti memoriji drugog procesa.
- Korisnički procesi mogu dijeliti memoriju.
- Neka memorija u korisničkim procesima može biti samo za čitanje.
- Sustav može koristiti više memorije nego što je fizički prisutno korištenjem prostora na disku kao pomoćnog.

Moderni CPU-i uključuju jedinicu za upravljanje memorijom (engl. *memory management unit* – MMU) koja omogućuje shemu pristupa memoriji koja se zove virtualna memorija. Koristeći virtualnu memoriju, proces ne pristupa izravno memoriji putem svoje fizičke lokacije u hardveru, nego jezgra pokazuje svakom procesu kao da ima cijeli stroj za sebe. Kada proces pristupi dijelu svoje memorije, MMU presreće pristup i koristi mapu memorijske adrese za prevođenje memorijske lokacije iz procesa u stvarnu fizičku memorijsku lokaciju na stroju. Jezgra i dalje mora inicijalizirati i kontinuirano održavati i mijenjati ovu mapu memorijske adrese. Ukratko, MMU prevodi virtualne memorijske adrese koje koriste procesi u stvarne dok jezgra pomaže MMU-u razbijanjem memorije koju koriste procesi u manje dijelove koji se nazivaju stranicama, [10].

3.2.4. Virtualni datotečni sustav

Tvrdi disk je velika hrpa blokova, gdje se podaci mogu pohraniti. Za pristup tim podacima potrebna je neka struktura, koja se naziva datotečni sustav. Postoji mnogo datotečnih sustava i Linux podržava mnoge od njih. VFS je softverski sloj u jezgri koji pruža sučelje datotečnog sustava programima korisničkog prostora. Također pruža apstrakciju unutar jezgre koja omogućuje koegzistiranje različitih implementacija datotečnog sustava, [10].

3.2.5. Blok I/O

Cilj ovog podsustava je upravljanje blok uređajima i zahtjevima prema njima. Blok uređaj je hardverski uređaj koji upravlja svojim podacima u dijelovima podataka fiksne veličine, koji se nazivaju blokovi. Podacima se zatim pristupa (ne nužno) sekvencijalnim pristupom. Redoslijed blokova nije bitan, što opet zahtijeva skuplje upravljanje blok uređajem jer mora imati mogućnost navigacije s jedne lokacije na drugu na disku. Uobičajeni primjer za blok uređaj je tvrdi disk. I/O-

određivač pokušava obraditi zahtjeve i staviti ih u razuman redoslijed obrade kako bi minimizirali pomicanje glave diska. Naravno, uzastopni blokovi su brže dostupni nego nasumični blokovi zato jezgra pokušava prikupiti pristupe za blok uređaje, predmemorirati već pristupljene podatke i promijeniti njihov redoslijed, [10].

3.2.6. Upravljački programi uređaja

Dobar OS podržava mnoštvo hardvera. Korisnici žele imati mogućnost korištenja nekoliko WLAN čipova, TV kartica, Bluetooth uređaja, itd. Kako bi to podržao, Linux definira sučelje za upravljačke programe uređaja koje omogućuje pisanje specifičnog koda za određeni hardver, ali i opći način rada s njima.

3.2.7. Mreža

Mrežni pristup važan je dio računala, njime upravlja jezgra jer je to jedino mjesto gdje je rukovanje mrežom dovoljno brzo. Fizički i mrežni sloj moraju biti povezani tako da mrežni sloj zadrži svoju neovisnost o hardveru. Linux jezgra održava svoju vlastitu podjelu između dva sloja i pruža komunikacijske standarde za njihovo povezivanje koji se nazivaju mrežnim sučeljem jezgre, [7].

3.3. Značajke Linux operativnog sustava

Linux je danas potpuno konkurentan komercijalnim OS-ima. Komercijalni OS-ovi često uvode nove značajke kako bi osvojili veći dio tržišta, ali te značajke nisu nužno korisne, stabilne ili produktivne. S druge strane, Linux ne trpi ograničenja i uvjete koje nameće tržište, stoga se može slobodno razvijati prema zamislima svojih dizajnera. Konkretno, Linux nudi sljedeće prednosti u odnosu na svoje komercijalne konkurente, [11]:

- Linux je besplatan. Kompletan OS je moguće instalirati bez ikakvih troškova osim troškova hardvera.
- Linux je u potpunosti prilagodljiv u svim svojim komponentama. Zahvaljujući opcijama kompilacije, jezgra se može prilagoditi svakom korisniku odabirom stvarno potrebnih značajki. Zahvaljujući GPL-u[1] dopušteno je slobodno čitanje i mijenjanje izvornog koda jezgre i svih sistemskih programa, što znači da korisnici Linux-a mogu odabrati po volji

[1] *General Public License*

osnovne komponente sustava, kao što je sustav za prikaz grafike/slike? kao i druge komponente korisničkog sučelja.

- Linux može raditi na jeftinim hardverskim platformama. Na primjer, postoje inačice Linux-a koje će raditi na mrežnim poslužiteljima koji koriste stari Intel 80386 sustav s 4 megabajta (engl. *megabyte* – MB) RAM-a.
- Linux je moćan. Linux sustavi su vrlo brzi jer u potpunosti iskorištavaju značajke hardverskih komponenti. Glavni cilj Linux-a je učinkovitost, te su programeri Linux-a odbacili mnoge izbore dizajna komercijalnih varijanti, zbog njihovog lošeg utjecaja na performanse sustava.
- Linux programeri su izvrsni programeri. Linux sustavi su vrlo stabilni; imaju vrlo nisku stopu kvarova i vrijeme održavanja sustava.
- Linux jezgra može biti vrlo malena i kompaktna. Moguće je staviti sliku jezgre, uključujući nekoliko sistemskih programa, na samo jedan disk od 1,44 MB. Zato je Linux popularan izbor OS-a za mnogi uređaje, kao što su Android telefoni i tableti, uređaji za digitalnu pohranu, videorekorderi, kamere, nosivi uređaji pa čak i neki automobili rade na Linux-u.
- Linux je vrlo kompatibilan s mnogim uobičajenim OS-ima. Linux omogućuje izravnu ugradnju datotečnih sustava za sve verzije MS-DOS i Microsoft Windows i mnogih drugih OS-a. Linux također može raditi s mnogim mrežnim slojevima, kao što su *Ethernet* (kao i *Fast Ethernet*, *Gigabit Ethernet* i 10 *Gigabit Ethernet*), FDDI (engl. *Fiber Distributed Data Interface*), HIPPI (engl. *High Performance Parallel Interface*), IEEE 802.11 (*Wireless LAN*) i IEEE 802.15 (*Bluetooth*). Korištenjem odgovarajućih biblioteka, Linux sustavi čak mogu izravno pokretati programe napisane za druge OS-e.
- Linux je jako dobro podržan. Puno je lakše napraviti zakrpe i ažuriranja za Linux nego za bilo koji vlasnički OS. Zbog velike zajednice i mnoštva razvojnih programera koji rade na sustavu, prijavljeni problemi se rješavaju unutar nekoliko sati ili dana, ovisno o kompleksnosti problema. Štoviše, upravljački programi za Linux obično su dostupni nekoliko tjedana nakon što su novi hardverski proizvodi predstavljeni na tržištu, dok proizvođači hardvera izdaju upravljačke programe za samo nekoliko komercijalnih OS-a, obično Microsoft-ove. Zato sve komercijalne varijante Unix-a rade na ograničenom podskupu hardverskih komponenti.

Otprilike 3 do 3,5 milijarde ljudi koristi Linux, na ovaj ili onaj način. Nije jednostavno definirati točan broj korisnika Linux-a. Kada je riječ o računalnom OS-u, 85% tržišnog udjela pripada Microsoft Windows-u, međutim, svaki Android telefon koristi Linux jezgru kao i mnogi skeneri, usmjerivači, pisači i modemi vjerojatno temeljeni na manjem Linux sustavu. Većina web stranica također se temelji na Linux-u, [12].

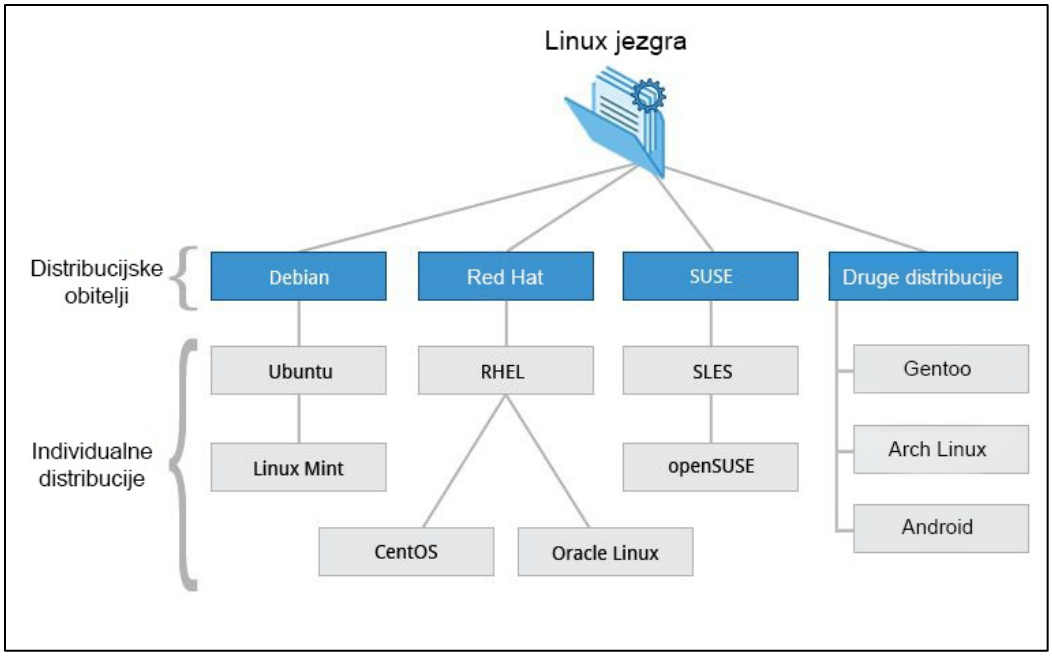
Korisnici su se navikli na određene značajke koje su standardne na drugim OS-ima te počinju očekivati isto od Linux-a. Prema tome, Linux je razvijen pod vodstvom Linus-a i njegovih održavatelja podsustava kako bi se prilagodio potrebama masa.

3.4. Linux distribucije

Linux je razvijen kao besplatan softver, dopuštajući korisnicima da stvaraju njegove razne varijacije. Linux distribucija je OS sastavljen od komponenti koje su razvili različiti projekti otvorenog koda i programeri. Svaka distribucija uključuje određenu verziju Linux jezgre, programe ljuske (terminal i naredbe), X poslužitelj (za grafičku radnu površinu), okruženje radne površine, sustav za upravljanje paketima, instalacijske programe i druge usluge. Mnoge komponente razvijene su neovisne jedna o drugoj i distribuirane su u obliku izvornog koda. Distribucije također uključuju internetski preglednik, alate za upravljanje i drugi softver kao što je KVM (engl. *Kernel-based Virtual Machine*) hipervizor. Jedna distribucija Linuxa može sadržavati tisuće softverskih paketa, programa i aplikacija, [13]. hipervizor. Jedna distribucija Linuxa može sadržavati tisuće softverskih paketa, programa i aplikacija, [13].

Distribucije Linux-a kompiliraju kod iz projekata otvorenog koda i kombiniraju ga u jedan OS koji se može instalirati i pokrenuti. Budući da se radi o softveru otvorenog koda, svatko može napraviti vlastitu distribuciju Linux-a tako što će je sam sastaviti iz izvornog koda ili modificirati postojeću distribuciju. Trenutno se aktivno održava više od 300 Linux distribucija. Postoje tri glavne distribucijske obitelji, prikazane na slici 4, iz kojih su se razvile gotovo sve ostale Linux distribucije, a to su:

- Debian (Ubuntu, Linux Mint, Kali Linux)
- Red Hat (Fedora, Red Hat Enterprise Linux- RHEL, CentOS)
- SUSE (openSUSE, SUSE Linux Enterprise Server- SLES)



Slika 4 Linux distribucije

Izvor: [14]

Postoje komercijalno podržane distribucije te distribucije kojima u potpunosti upravlja zajednica. Komercijalne Linux distribucije su one koje su dostupne uz naknadu, kao što su Red Hat Enterprise Linux, SUSE Enterprise Linux Server i Mandriva (bivši Mandrake). Oni obično uključuju proizvod s podrškom, kao i licencirane aplikacije ili upravljačke programe koji nisu otvorenog koda. Nekomercijalne ili besplatne distribucije uključuju Debian, Ubuntu i Gentoo koji su dostupni kao nekomercijalni proizvodi. Red Hat (kroz svoj projekt Fedora Core), SUSE i Mandriva također nude nekomercijalne verzije, [14].

Prema posljednjim istraživanjima, razvojni programeri ne vjeruju da postoje bitne prednosti komercijalne verzije u odnosu na nekomercijalnu te smatraju da su najveće prednosti nekomercijalnih distribucija Linux-a jednostavnost korištenja i troškovi održavanja i nadogradnje, [15].

4. Analiza ranjivosti Linux operativnog sustava

Ranjivost (eng. *Vulnerability*) je slabost sustava koja može biti iskorištena u svrhu uzrokovanja gubitka informacija ili nanošenja štete sustavu. Ranjivosti mogu biti različite, kao i način njihovog iskorištavanja. To je stanje, nedostatak ili slabost u sigurnosnim procedurama, tehničkim kontrolama, fizičkim i drugim kontrolama sustava, dizajnu i implementaciji tih kontrola i procedura koje je moguće iskoristiti. Slučajno ili namjerno iskorištavanje može prouzrokovati operativne i financijske gubitke sustavu, [16].

Međunarodna organizacija za standardizaciju (engl. *The International Organization for Standardization* – ISO) definira sigurnosnu ranjivost kao slabost imovine ili grupe imovine koju može iskoristiti jedna ili više kibernetičkih prijetnji, pri čemu je imovina sve što ima vrijednost za organizaciju, njezino operativno poslovanje i njegov kontinuitet, uključujući informacijske resurse koji podržavaju misiju organizacije.

Četiri glavne vrste ranjivosti u informacijskoj sigurnosti su mrežne ranjivosti, ranjivosti OS-a, ranjivosti procesa (proceduralne ranjivosti) i ljudske ranjivosti, [17]:

- Mrežne ranjivosti su slabosti infrastrukture organizacije (hardverske ili softverske) koje zlonamjernim korisnicima omogućuju pristup i nanošenje štete. Ta područja izloženosti mogu varirati od slabo zaštićenog bežičnog pristupa pa sve do pogrešno konfiguriranog vatrozida koji ne štiti mrežu u cjelini.
- Ranjivosti OS-a su izloženosti unutar OS-a koje zlonamjernim korisnicima omogućuju da izazovu štetu na bilo kojem uređaju na kojem je instaliran. Primjer napada koji iskorištava ranjivosti OS-a je napad uskraćivanja usluge (DoS), gdje ponovljeni lažni zahtjevi preopterećuju sustav tako da postaje neuporabljiv. Nezakrpani i zastarjeli softver također stvara ranjivosti OS-a, jer je sustav koji pokreće aplikacije nesiguran te tako ugrožava cijelu mrežu.
- Ranjivosti procesa nastaju kada su procedure koje bi trebale djelovati kao sigurnosne mjere neefikasne. Jedna od najčešćih ranjivosti procesa je slabost autentifikacije, gdje korisnici, pa čak i IT administratori, koriste slabe lozinke.
- Ljudske ranjivosti stvaraju pogreške korisnika koje mogu izložiti mreže, hardver i osjetljive podatke zlonamjernim korisnicima. Oni nedvojbeno predstavljaju najveću prijetnju, osobito zbog povećanja udaljenog i mobilnog rada. Primjeri ljudske ranjivosti su

otvaranje privitka elektroničke pošte zaražene zlonamjernim softverom ili ne ažuriranje softvera na mobilnim uređajima.

Nadalje ranjivosti OS-a moguće je kategorizirati na sljedeći način, [18]:

1. Ranjivosti uzrokovane programskim nedostacima u isporučenim programskim paketima
   - otkrivaju se uporabom skenera (najčešće mrežnih) zatim se eliminiraju
2. Konfiguracijske ranjivosti
   - rizične vrijednosti konfiguracijskih postavki otklanjaju se pravilnim upravljanjem sigurnosnim postavkama (konfiguracija korisničkih računa, pristupna prava)
3. Ranjivosti web aplikacija
   - rješavaju se implementacijom sigurnosnih zaštita za sprječavanje napada i uklanjanja ranjivosti

Mnogo je institucija koje svoju aktivnost usmjeravaju na otkrivanje i uklanjanje ranjivosti. Oni nedvojbeno uključuju dobavljače softvera kao i brojne vladine i neovisne međunarodne organizacije, komercijalna poduzeća pa čak i pojedince. Većina tih institucija daje javno dostupne skupove podataka o ranjivostima. Najpoznatiji i najpouzdaniji su navedeni u nastavku.

4.1. Pregled koncepata ranjivosti – CVE

U prošlosti, javne baze podataka o ranjivostima nisu imale standardna pravila za opis ranjivosti te su uzrokovale poteškoće u komunikaciji i razmjeni informacija o ranjivostima, sve do 1999. godine kada je tvrtka MITRE predstavila koncept *Common Vulnerability Enumeration* (CVE) kao mehanizma koji ima za cilj riješiti ovaj problem i pružiti standardni način za opisivanje karakteristika ranjivosti.

MITRE je američka, neovisna, neprofitna tvrtka koja radi u javnom interesu i pruža tehničku podršku vladi. Koriste se raznim alatima za procjenu sigurnosnog stanja, uključujući nekoliko alata za procjenu mreže i sustave za otkrivanje upada (engl. *intrusion detection system* - IDS) raznih dobavljača. Svi ovi alati obrađuju ranjivosti te se pomoću njih gradi sustav koji može integrirati i upravljati informacijama o ranjivostima iz različitih izvora u centraliziranu bazu podataka. Kako bismo postigli interoperabilnost sigurnosnih alata i dijelili informacije o ranjivostima, potreban nam je CVE, standardizirani popis koji, [19]:

- nabraja i razlikuje sve poznate ranjivosti
- dodjeljuje standardno, jedinstveno ime svakoj ranjivosti (engl. *CVE Identifier* – CVE ID)

- postoji neovisno o višestrukim perspektivama onoga što je ranjivost
- javno je "otvoren" i može se dijeliti bez ograničenja

Zajednica koja se bavi kibernetičkom sigurnošću potvrdila je važnost CVE-a. Brojni veliki dobavljači OS-a i druge organizacije iz cijelog svijeta uključuju CVE ID u svoja upozorenja kako bi osigurali da međunarodna zajednica ima koristi od njih čim se pojavi problem. Osim toga, CVE zapisi se koriste za jedinstvenu identifikaciju ranjivosti na javnim popisima za praćenje kao što je OWASP (engl. *Open Web Application Security*) Top 10 sigurnosnih problema web aplikacija, [19].

Između ostalog, CVE se ocjenjuju prema ozbiljnosti po *Common Vulnerability Scoring System* (CVSS), otvorenom okviru za procjenu karakteristika i ozbiljnosti softverskih ranjivosti. CVSS se sastoji od tri skupine mjernih podataka: osnovna, vremenska i skupina koja se odnosi na okolinu. Osnovna skupina predstavlja svojstvene kvalitete ranjivosti koje su konstantne tijekom vremena i u različitim korisničkim okruženjima, vremenska skupina odražava karakteristike ranjivosti koje se mijenjaju tijekom vremena, a treća skupina predstavlja karakteristike ranjivosti koje su jedinstvene za korisničko okruženje. Osnovna skupina mjernih podataka daje rezultat u rasponu od 0 do 10, koji se zatim može modificirati bodovanjem vremenske skupine i skupine okoline, [20].

Slika 5 CVSS rezultat

Izvor: [20]

Kao što je to vidljivo na slici 4, numerička ocjena se zatim može prevesti u kvalitativni prikaz (kao što je niska, srednja, visoka i kritična) kako bi se organizacijama pomoglo da ispravno procijene i daju prioritet svojim procesima upravljanja ranjivostima, [21].

4.2. Baza podataka o ranjivostima – NVD

National Vulnerability Database (NVD) nacionalna je baza podataka o ranjivostima koju održava Nacionalni institut za standarde i tehnologiju SAD-a (engl. *National Institute of Standards and Technology* - NIST), nadograđuje se na CVE i sinkronizirana je s njim. Za razliku od CVE-a, on kategorizira ranjivosti prema vrsti i ozbiljnosti, pruža određeni popis ranjivih softverskih proizvoda uključujući baze podataka sigurnosnih kontrolnih popisa, sigurnosnih softverskih nedostataka, pogrešnih konfiguracija i naziva proizvoda.

CVE lista sadržana je u bazi podataka NVD, koja se zatim nadovezuje na informacije uključene u CVE zapise kako bi pružila proširene informacije za svaki zapis kao što su informacije o popravcima, rezultati ozbiljnosti i ocjene utjecaja. Kao dio svojih proširenih informacija, na NVD web stranici se mogu pronaći napredne značajke pretraživanja kao što su pretraživanje po OS-u, prema nazivu dobavljača, nazivu proizvoda i/ili broju verzije te prema vrsti ranjivosti, ozbiljnosti, povezanom rasponu iskorištavanja i utjecaju, [22].

Štoviše, vidljivo je da MITRE i NIST, koji upravljaju CVE i NVD bazama, ulažu relativno jednake napore u ispitivanje ranjivosti različitih softverskih proizvoda i pružaju pouzdane informacije koje se mogu koristiti kao pokazatelj sigurnosti ili kvalitete softvera.

4.3. Pregled koncepata slabosti – CWE

Common Weakness Enumeration (CWE) popis je uobičajenih tipova slabosti softvera i hardvera koje imaju sigurnosne posljedice. Slabosti su nedostaci, mane ili pogreške u softverskoj ili hardverskoj implementaciji, kodu, dizajnu ili arhitekturi koje bi, ako se ne riješe, mogle dovesti do toga da sustavi, mreže ili hardver budu ranjivi na napad.

CWE nastoji upravljanje ranjivostima učiniti jednostavnijim i pristupačnijim. CWE lista sastavljena je kako bi pomogla i razvojnim inženjerima i sigurnosnim stručnjacima te joj je glavni cilj zaustaviti ranjivosti na izvoru, edukacijom programera softvera i hardvera, dizajnera i dobavljača o tome kako eliminirati najčešće pogreške prije isporuke proizvoda. U konačnici, korištenje CWE liste pomaže u sprječavanju vrsta sigurnosnih ranjivosti koje su predstavljale prijetnju industriji softvera i hardvera te dovodile poduzeća u opasnost, [23].

CWE isto kao i CVE, ima sustav za bodovanje *Common Weakness Scoring System* (CWSS) koji služi za određivanje prioriteta softverskih slabosti. Slično kao i CVSS, CWSS organiziran je

u tri skupine mjernih podataka: osnovna, skupina površine napada i skupina koja se odnosi na okolinu. Svaka grupa sadrži više faktora koji se koriste za izračunavanje CWSS rezultata, [24]:

- Osnovna skupina mjernih podataka obuhvaća svojstven rizik slabosti, povjerenje u točnost nalaza i snagu kontrola.
- Skupina mjernih podataka površine napada predstavlja barijere koje napadač mora svladati kako bi iskoristio slabost.
- Skupina mjernih podataka okoline karakteristike slabosti koje su specifične za određeno okruženje.

Razlika između CVE-a i CWE-a je u tome što CWE kategorizira vrste softverskih ranjivosti dok se CVE odnosi na određenu instancu unutar proizvoda ili sustava, a ne s temeljnim nedostacima. Ugrubo, možemo reći da je CWE uzrok, a CVE njegova posljedica. CWE se fokusira na vrstu pogreške ili slabosti koja se može iskoristiti uz odgovarajuće uvjete za stvaranje ranjivosti u proizvodu, ali on nije usredotočen na ranjivosti, već ima glavni fokus na pogreške koje se mogu pojaviti u implementaciji, dizajnu ili drugim fazama životni ciklus proizvoda, [25].

4.4. Ranjivosti jezgre Linux operativnog sustava

Primarni izvor ranjivosti softvera su slabosti i greške u dizajnu i implementaciji softvera. OS-i kao i aplikacijski softveri mogu sadržavati ranjivosti, no sigurnosni nedostaci u OS-u su bez sumnje najkritičniji jer ako ih zlonamjerni korisnik iskoristi, sve usluge i procesi koje izvršava OS mogu biti ugroženi te je moguće da korisnik dobije nedozvoljen pristup svim podacima koji su pohranjeni na uređaju. Štoviše, prijetnje koje oni predstavljaju pouzdanosti i sigurnosti sustava puno su većih razmjera od klasičnih pogrešaka koje postoje u svim aplikacijskim softverima, [26].

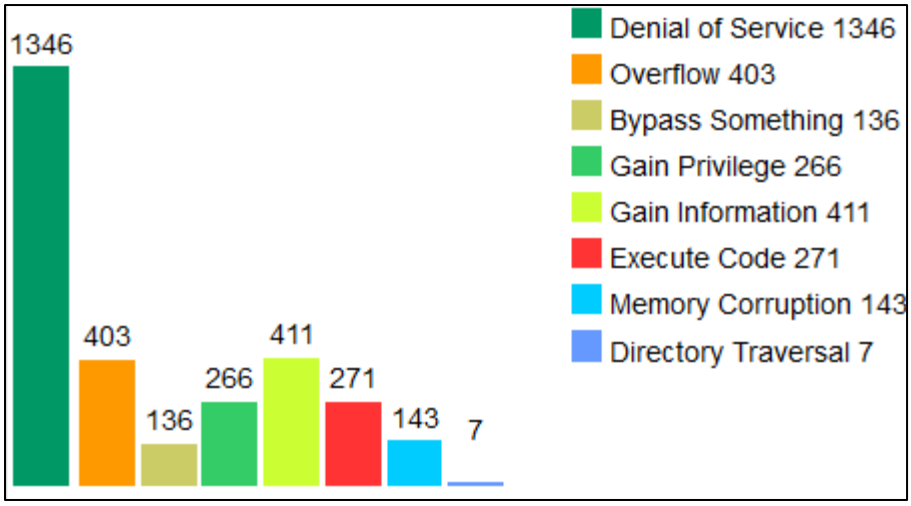
Ranjivosti Linux OS-a mogu se kategorizirati na mnogo načina jer svaka ranjivost posjeduje sljedeće atribute, [28]:

- jedinstveni ID poznat kao CVE ID,
- kratki opis koji sadrži informacije o zahvaćenom softveru,
- vektor napada, te vrsta napada koji se može provesti uspješnim iskorištavanjem ranjivosti,
- uzrok ranjivosti poznat kao CWE ID,

- ozbiljnost ranjivosti izračunata pomoću standardiziranog mehanizma bodovanja poznatog kao CVSS,
- potencijalni utjecaj na povjerljivost, cjelovitost i dostupnost ako se ranjivost iskoristi te
- razina složenosti za pristup i iskorištavanje ranjivosti.

Međutim, treba napomenuti da neke ranjivosti mogu imati više uzroka ranjivosti što dodatno otežava njihovu kategorizaciju.

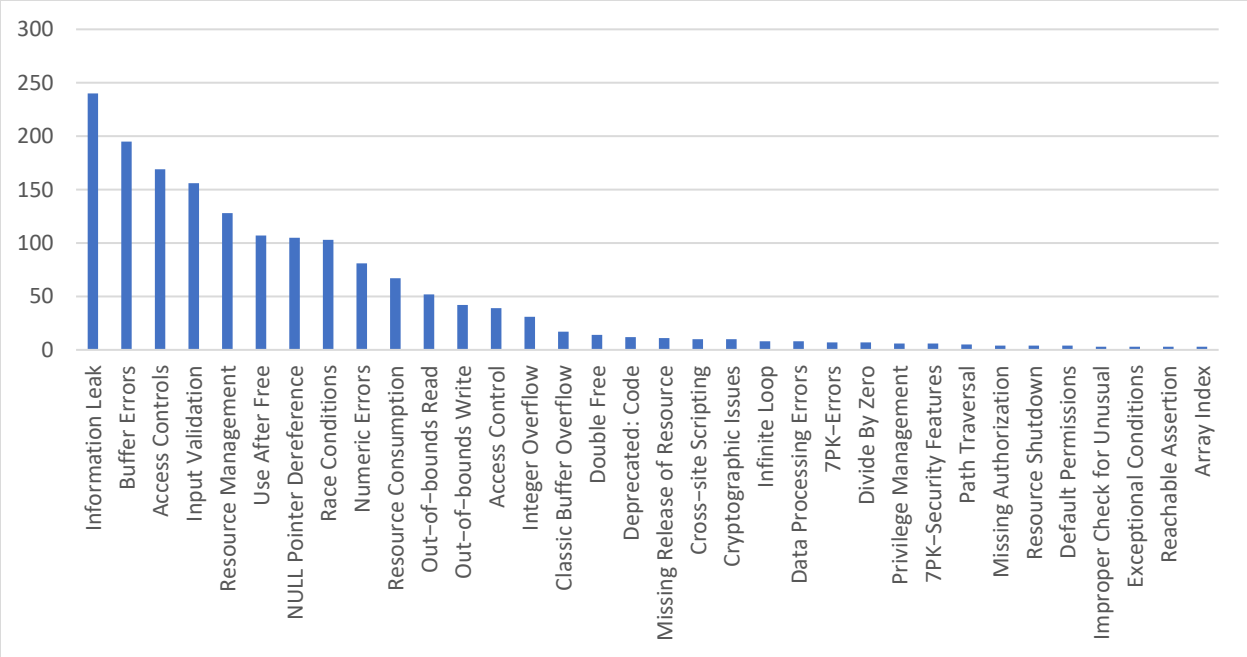
Internet stranica www.cvedetails.com nudi jednostavno sučelje za podatke o ranjivostima. Ovdje se mogu pretraživati dobavljači, proizvodi i verzije softvera te pregledavati CVE zapisi povezani s njima. Također se može vidjeti statistika o ranjivostima određenih dobavljača te njihovih proizvoda. Na slici 5 prikazana je statistika Linux ranjivosti uzeta iz vremenskog perioda od 1999. godine do 2022. godine, kategorizirana prema vrsti napada.



Slika 6 Kategorizacija ranjivosti prema vrsti napada, [27]

Iz statistike je vidljivo da većina ranjivosti Linux OS-a (45%) može dovesti do napada uskraćivanja usluge, dok su drugi najčešći prekoračenje međuspremnika te prikupljanje informacija.

Autori u radu [28], analizirali su 1858 ranjivosti Linux jezgre (verzije 1.2.0- 5.4) od siječnja 2010. godine do siječnja 2020. godine. Grupirali su vrste ranjivosti prema uzroku ranjivosti (CWE). Slika 7 prikazuje rezultate klasificiranja ranjivosti Linux jezgre podijeljenih u 34 kategorije gdje su se najčešće javljale ranjivosti tijekom tih deset godina.



Slika 7 Klasifikacija ranjivosti Linux jezgre

Izvor: [28]

Iz slike 7 moguće je razlučiti najučestalije vrste ranjivosti koje su navedene u nastavku.

4.4.1. Curenje informacija

Curenje informacija (CWE-200, *Exposure of Sensitive Information to an Unauthorized Actor*) je namjerno ili nenamjerno otkrivanje ovlaštenih informacija zlonamjernom korisniku. Ova slabost bi mogla biti rezultat problema koji uključuju izloženost osjetljivim informacijama. Može se dogoditi u mnogim situacijama kao što je slanje podataka, spremanje podataka u predmemoriju ili indeksiranje privatnih podataka. Ne postoji specifično rješenje za ovu vrstu ranjivosti. Trebalo bi se osigurati da osjetljivi podaci nikad ne izađu izvan perimetra sigurnosti, [28].

4.4.2. Ranjivosti međuspremnika

Ranjivosti u ovoj kategoriji (CWE-119, *Improper Restriction of Operations within the Bounds of a Memory Buffer*) mogu dovesti do koda koji može čitati ili pisati na memorijsku lokaciju koja je izvan predviđene granice međuspremnika. Međuspremnik je kontinuirani niz memorije koju koristi program. Ova ranjivost omogućuje napadaču umetanje i izvršavanje vlastitog zlonamjernog koda koji pomaže u lakom dobivanju ovlasti. Ne postoje poznati načini za potpuno sprječavanje

pogrešaka međuspremnika ni na softverskoj ni na hardverskoj razini. Većina široko predloženih rješenja imaju velike memorijske troškove, [28].

4.4.3. Kontrola pristupa

Ranjivost kontrole pristupa (CWE-264, *Permissions, Privileges, and Access Controls*) omogućuje napadaču da dobije ovlašteni put do važnih sredstava sustava obično zaobilazeći dopuštenja ili ovlasti. Napadači iskorištavaju ovu vrstu ranjivosti obično za otkrivanje osjetljivih informacija, izmjenu datoteka ili iniciranje drugih mogućih napada. Mnogi dijelovi Linux jezgre bi se trebali poboljšati kako bi se ublažio ovaj tip napada, uključujući, [29]:

1) *root* račun- tradicionalno ime za povlašteni korisnički račun na Unix-ovim OS-ima,
2) autentifikaciju korisnika i atribute korisničkog računa,
3) udaljenu autentifikaciju,
4) datotečni sustav
5) konfiguraciju usluga.

Postoji nekoliko rješenja za poboljšanje sigurnosti Linuxa koja se odnose na upravljanje putanjom. Postojeći alati, poput *up2date*, *YaST* i *apt-get*, koji mogu automatski preuzeti i instalirati sigurnosna ažuriranja, trebaju se koristiti za pružanje zakrpa.

4.4.4. Potvrda valjanosti ulaznih podataka

Ranjivost potvrde valjanosti unosa (CWE-20, *Improper Input Validation*) odnosi se na neispravnu provjeru valjanosti ulaznih podataka koja može utjecati na tok kontrole ili tok podataka programa. Ova slabost može se nalaziti u širokom rasponu Linux modula, kao što su USB, upravljačkim programima uređaja te mnogim funkcijama u bibliotekama. Kako bi se ublažila ova ranjivost, treba ispraviti odgovarajuće funkcije, [28].

4.4.5. Upravljanje resursima

Ranjivost upravljanja resursima (CWE-399, *Resource Management Errors*) odnosi se na nepravilno upravljanje resursima sustava. Ova se ranjivost može dogoditi u mnogim podsustavima Linux-a, kao što je curenje memorije u upravljačkom programu grafičke kartice, NFSv4 implementacija datotečnog sustava, TCP vezi u mreži, implementaciji utičnice i drugim. Ne postoji sveobuhvatno rješenje za sve ranjivosti ove vrste i treba ih korigirati jednu po jednu, [28].

4.5. Kategorizacija ranjivosti prema stupnju rizika

Kategorizacija ranjivosti, osim po vrstama napada te po uzroku ranjivosti, može se provesti na temelju ozbiljnosti ranjivosti koja je definirana CVSS ocjenom. Što je ocjena veća, to je ranjivost opasnija. Kompanija MITRE pruža statistički pregled podataka za pojedine dobavljače prema broju različitih ranjivosti u njihovim proizvodima. Na slici 8 može se vidjeti popis određenih dobavljača prema ukupnom broju različitih ranjivosti distribuiranih po CVSS ocjeni. Sa slike je vidljivo da je najveći broj ranjivosti, čak 8802 pronađeno u proizvodima kompanije Microsoft, dok je Linux tek 11-i na listi sa svega 2891 ranjivosti.

	Naziv dobavljača	Ukupan broj ranjivosti	Broj ranjivosti									
			0-1	1-2	2-3	3-4	4-5	5-6	6-7	7-8	8-9	9+
1	Microsoft	8802	2	111	634	247	1730	984	949	1915	39	2191
2	Oracle	8670	64	146	425	563	2628	2466	1007	743	41	587
3	Google	7327	1	54	737	99	1989	690	1249	1338	37	1133
4	Debian	6735	2	81	374	172	1877	1376	1396	1155	21	281
5	Apple	5680	1	58	395	52	1125	707	1521	785	17	1019
6	IBM	5481	6	64	369	988	1490	1050	550	539	27	398
7	Redhat	4344	1	66	339	192	1142	781	720	732	16	355
8	Cisco	4263	2	6	95	193	958	911	565	987	47	499
9	Fedoraproject	3457		34	189	98	1018	770	808	440	11	89
10	Canonical	3452		48	224	112	1042	612	539	611	9	255
11	Linux	2891	4	106	472	84	925	165	231	764	9	131

Slika 8 Distribucija po CVSS ocjeni

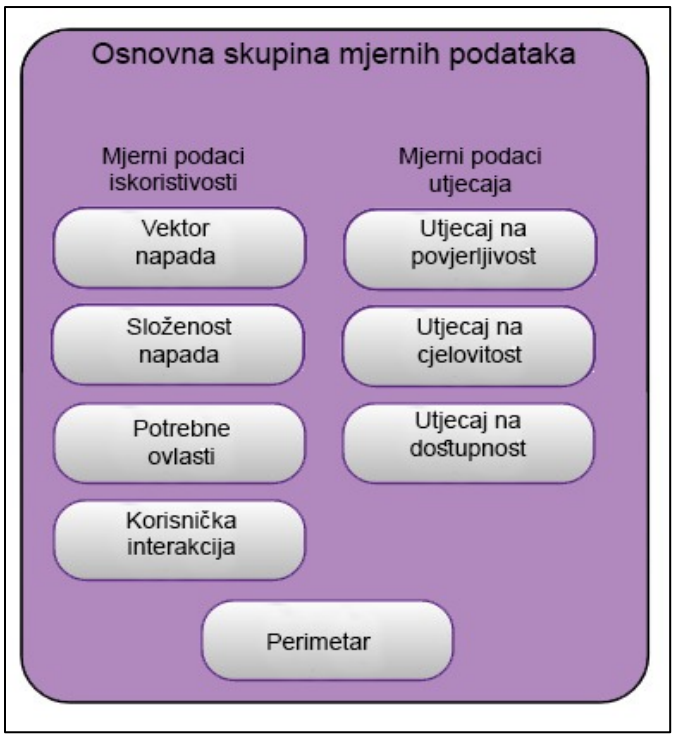
Izvor: [30]

U poglavlju 4.1. opisano je kako se numerička ocjena može prevesti u kvalitativni prikaz:

- Nula: 0.0
- Niska: 0.1- 3.9
- Srednja: 4.0- 6.9
- Visoka: 7.0- 8.9
- Kritična: 9.0- 10.0

S obzirom na to, može se zaključiti kako Linux ima 4 ranjivosti vrijednosti nula, 662 ranjivosti niske ozbiljnosti, 1321 ranjivost srednje ozbiljnosti, 773 ranjivosti visoke ozbiljnosti te 131 kritičnu.

Da bi razumjeli kako se kategoriziraju ranjivosti po CVSS ocjeni, potrebno je razumjeti kako se CVSS ocjena računa. U poglavlju 4.1. objašnjeno je da se CVSS sastoji od tri skupine mjernih podataka: osnovna, vremenska i skupina koja se odnosi na okolinu. Osnovna skupina mjernih podataka najbitnija je komponenta CVSS-a. Ona ne podliježe promjenama tijekom vremena niti ovisi o korisničkoj okolini. Na slici 9 prikazana je podjela osnovne skupine mjernih podataka. Sastoji se od dva skupa mjernih podataka, prvi se odnose na iskorištavanje ranjivosti, a drugi na utjecaj na sustav.



Slika 9 Osnovna skupina mjernih podataka CVSS verzije 3

Izvor: [31]

Trenutno korištena verzija CVSS-a je verzija tri, a u nastavku je detaljno objašnjeno kako se računa ocjena osnovne skupine mjernih podataka, [31].

4.5.1. Mjerni podaci iskoristivosti ranjivosti

Mjerni podaci iskoristivosti odražavaju karakteristike ranjive komponente, stoga bi se svaka od niže navedenih skupina podataka iskoristivosti trebala ocijeniti u odnosu na ranjivu

komponentu i odražavati svojstva ranjivosti koja dovode do uspješnog napada. Mjerni podaci iskoristivosti dijele se na, [31]:

- Vektor napada (engl. *Attack Vector*) odražava kako se ranjivost iskorištava: fizički, lokalno, preko susjedne mreže ili preko mreže. Ovaj mjerni podatak (a time i osnovni rezultat) bit će veći što je napadač udaljeniji od ranjive komponente. Pretpostavka je da je broj potencijalnih napadača na ranjivost koji bi mogli iskoristiti ranjivost preko cijele mreže veći od broja potencijalnih napadača koji bi mogli iskoristiti ranjivost koja zahtijeva fizički pristup uređaju, te stoga jamči veću vrijednost osnovnog rezultata.
- Složenost napada (engl. *Attack Complexity*) može biti visoka, srednja ili niska. Ona opisuje prepreke koje napadač treba zaobići da bi napad bio uspješan. Takve prepreke mogu zahtijevati prikupljanje više informacija o meti ili računalne iznimke. Osnovni rezultat je najveći za najmanje složene napade.
- Potrebne ovlasti (engl. *Privileges Required*) odražavaju razinu ovlasti koju napadač mora posjedovati prije nego što uspješno iskoristi ranjivost. Moguće vrijednosti su: neovlaštena te niska i visoka razina ovlasti. Osnovni rezultat je najveći ako nije potrebna nikakva autentifikacija.
- Korisnička interakcija (engl. *User Interaction*) može biti potrebna ili ne. Ova skupina odražava doprinos zasebnog korisnika u uspješnoj kompromitaciji ranjive komponente, to jest određuje može li se ranjivost iskoristiti isključivo po volji napadača ili mora na neki način sudjelovati korisnik. Osnovni rezultat je najveći kada nije potrebna interakcija korisnika.

4.5.2. Opseg

Opseg (engl. *Scope*) odražava utječe li ranjivost u jednoj ranjivoj komponenti na resurse u komponentama izvan njenog sigurnosnog opsega, to jest može li napadač utjecati na druge resurse iz ugroženog resursa. Sigurnosni mehanizmi definiraju i kontroliraju načine na koji određeni subjekti/ akteri (npr. korisnici, procesi) mogu pristupiti određenim objektima/ resursima (npr. datoteke, CPU, memorija). Svi subjekti i objekti koji su u nadležnosti jednog sigurnosnog mehanizma smatraju se jednim sigurnosnim opsegom. Ako ranjivost u ranjivoj komponenti može utjecati na komponentu koja je u drugom sigurnosnom opsegu od ranjive komponente, dolazi do promjene opsega. Osnovni rezultat je najveći kada dođe do promjene opsega, [31].

4.5.3. Mjerni podaci utjecaja ranjivosti

Mjerni podaci utjecaja obuhvaćaju učinke uspješno iskorištene ranjivosti na komponentu sustava koja trpi najgori ishod povezan s napadom. Prilikom bodovanja skupina utjecaja trebalo bi ograničiti utjecaje na razuman, konačan ishod za koji je sigurno da se može postići. Mjerni podaci utjecaja dijele se po tri osnovna načela informacijske sigurnosti, a to su, [31]:

- Povjerljivost (engl. *Confidentiality*)- mjeri se utjecaj na povjerljivost informacijskih resursa kojima upravlja komponenta pogođena uspješnim iskorištavanjem ranjivosti. Povjerljivost se odnosi na ograničenje pristupa te otkrivanje informacija samo ovlaštenim korisnicima, kao i sprječavanje pristupa te otkrivanja informacija neovlaštenim korisnicima.
- Cjelovitost (engl. *Integrity*)- mjeri se utjecaj na cjelovitost informacijskih resursa nakon uspješnog iskorištavanja ranjivosti. Cjelovitost se odnosi na pouzdanost i istinitost informacija, te podrazumijeva zaštitu informacija od namjerne ili slučajne neovlaštene modifikacije uzrokovane ljudskim utjecajem ili pogreške u radu sustava.
- Dostupnost (engl. *Availability*)- mjeri se utjecaj na dostupnost komponente pogođene uspješnim iskorištavanjem ranjivosti. Dok se utjecaj na povjerljivost i cjelovitost odnosi na gubitak povjerljivosti ili cjelovitosti podataka (npr. informacija, datoteka) koje koristi pogođena komponenta, dostupnost se odnosi na gubitak dostupnosti same pogođene komponente, kao što je mrežna usluga (npr. web, baza podataka, e-pošta). Budući da se dostupnost odnosi na dostupnost informacijskih resursa, napadi koji pogađaju propusnost mreže, cikluse procesora ili prostor na disku utječu na dostupnost pogođene komponente.

Utjecaj na povjerljivost, integritet i dostupnost može biti nikakav, djelomičan ili potpun. Neke ranjivosti utječu samo na jedno načelo informacijske sigurnosti, dok druge mogu dovesti do proboja u dva ili sva tri.

5. Metodologija provedbe istraživanja

Procjene ranjivosti neophodne su za otkrivanje potencijalnih ranjivosti u okruženju. Dostupni su mnogi alati koji automatiziraju ovaj proces tako da čak i neiskusni sigurnosni stručnjaci ili amateri mogu učinkovito odrediti sigurnosno stanje svog okruženja.

Za provedbu ovog istraživanja koristit će se VirtualBox 6.1 hipervizor na kojemu će biti pokrenute virtualne mašine. Na jednoj virtualnoj mašini bit će pokrenut OS Kali Linux koji služi za penetracijska testiranja, dok će na drugoj biti pokrenuta ranjiva virtualna mašina Metasploitable 2. Pomoću alata OpenVAS, ranjiva virtualna mašina bit će skenirana te će njene ranjivosti biti identificirane potom iskorištene za simulaciju kibernetičkih napada.

5.1. Kali Linux

Kali Linux je najmoćnija i najpopularnija svjetska platforma u kontekstu kibernetičke sigurnosti i penetracijskog testiranja koju koriste sigurnosni stručnjaci u širokom rasponu specijalizacija, uključujući penetracijsko testiranje, forenziku, obrnuto inženjerstvo i procjenu ranjivosti. Kali Linux nije samo zbirka alata, već fleksibilan okvir koji profesionalni sigurnosni stručnjaci, entuzijasti, studenti i amateri mogu prilagoditi svojim specifičnim potrebama, [2].

Kali Linux inačica 1.0 puštena je u distribuciju početkom 2013. godine, razvijena pod okriljem organizacije *Offensive Security*. Glavne značajke Kali Linux distribucije, [2]:

- temeljena je na Debian Linux distribuciji- većina paketa dostupnih u Kali Linux-u dolazi izravno iz Debian repozitorija,
- može se koristiti na različitim uređajima: prijenosnim računalima, stolnim računalima, poslužiteljima, također se može implementirati u oblaku,
- podržava sustave temeljene na ARM tehnologiji, ARM uređaji savršeni su za provođenje napada zbog svog malog oblika i malih zahtjeva za napajanjem,
- podržava veliki broj bežičnih mrežnih kartica,
- korisnici ga mogu u potpunosti prilagoditi svojim potrebama je uključuje mnoge značajke za izmjenu instaliranog sustava, instaliranje dodatnih datoteka, dodatnih paketa, pokretanje proizvoljnih naredbi te promjenu unaprijed definiranih vrijednosti u konfiguraciji.

Kali sadrži stotine programskih alata koji se mogu podijeliti u sljedeće skupine, [16]:

- Prikupljanje informacija: programski alati koji se primjenjuju za prikupljanje podataka o ciljnoj mreži i njezinoj strukturi, identificiranje računala, njihovih OS-a i usluga koje pokreću.
- Procjena ranjivosti: alati za identifikaciju ranjivosti ciljanog sustava poput skenera ranjivosti koji koriste baze podataka koje sadrže tisuće zapisa za prepoznavanje potencijalnih ranjivosti.
- Analiza web aplikacija: alati za identificiranje pogrešnih konfiguracija i sigurnosnih slabosti u web aplikacijama, alati za iskorištavanje ranjivosti baza podataka te skeneri ranjivosti web aplikacija.
- *Password* napadi: alati korišteni za izvođenje napada s ciljem otkrivanja zaporki te alati za napade na enkripcijske ili kriptografske sustave
- Alati za iskorištavanje ranjivosti: alati primjenjivi pri iskorištavanju ranjivosti otkrivenih u ciljanom sustavu na mrežnoj, web i razini baza podataka. Unutar navedene kategorije nalaze se i alati za provođenje socijalnog inženjeringa.
- Alati za prikupljanje prometa i lažiranje informacija: alati za prikupljanje mrežnog i web prometa te za lažiranje informacija o napadaču.
- Održavanje pristupa: alati koji pomažu u održavanju pristupa ciljanom sustavu, *backdoor* alati za OS-e i web aplikacije.
- Alati za izvješća: alati za dokumentiranje procesa penetracijskog testiranja i dobivenih rezultata.
- Servisi: servisi koji mogu biti korisni tijekom procesa penetracijskog testiranja, poput Apache servisa, MySQL servisa, SSH servisa, itd.

Kali Linux je razvio mali tim iskusnih programera koji rade transparentno i slijede najbolje sigurnosne prakse kao što je postavljanje potpisanih izvornih paketa u repozitorij.

5.2. Metasploitable

Metasploitable virtualni stroj namjerno je ranjiva verzija Ubuntu Linux-a dizajnirana za provođenje sigurnosne obuke, testiranje sigurnosnih alata, demonstraciju uobičajenih ranjivosti te

vježbanje uobičajenih tehnika napada. Ovaj virtualni stroj kompatibilan je s VMWare, VirtualBox i drugim uobičajenim platformama za virtualizaciju, [32].

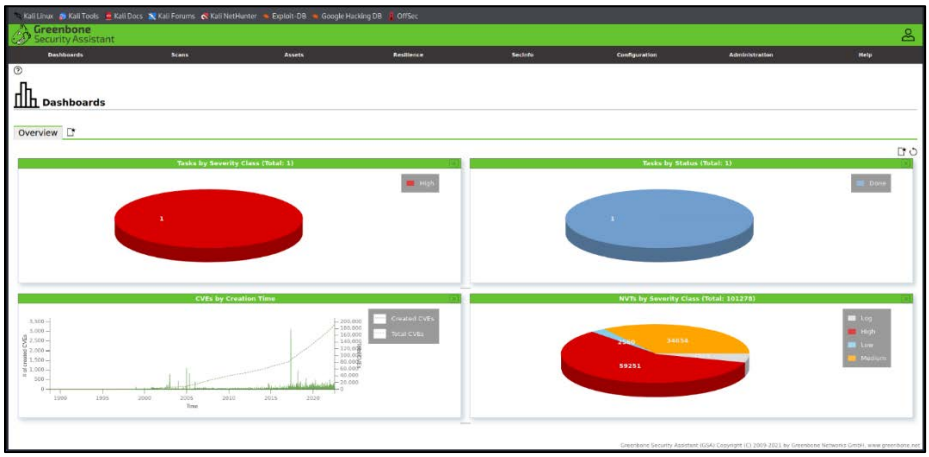
U ovom radu koristit će se Metasploitable 2, druga verzija ovog virtualnog stroja koja sadrži puno više ranjivosti od izvornog.

5.3. OpenVAS

OpenVAS (*Open Vulnerability Assessment Scanner*) je mrežno temeljeni skener ranjivosti. Razvila ga je tvrtka *Greenbone Networks* sa sjedištem u gradu Osnabrück u Njemačkoj 2006. godine kao dio komercijalne obitelji proizvoda za upravljanje ranjivostima *Greenbone Enterprise Appliance*, temeljeni na modulima otvorenog koda.

OpenVAS je razvijen za ispitivanje mreže i generiranje izvještaja vezanih uz sigurnost. Namijenjen je za administratore sigurnosti, ali svatko ga može besplatno preuzeti i koristiti. Njegove funkcionalnosti uključuju neautentificirano i autentificirano testiranje, razne industrijske protokole visoke i niske razine, podešavanje performansi za skeniranje velikih razmjera i snažan interni programski jezik za implementaciju bilo koje vrste testa ranjivosti. To je sjajan alat, ne samo za pronalaženje ranjivosti, već i za upravljanje njima, njihovo ispravljanje te generalno upravljanje projektima, [33].

Greenbone Security Assistant je OpenVAS web sučelje, dostupno lokalno na računalu na adresi https://localhost:9392. Nakon prihvaćanja samopotpisanog certifikata, prikazuje se stranica za prijavu i nakon provjere autentičnosti, vidi se glavna nadzorna ploča prikazana na slici 10.



Slika 10 Glavna nadzorna ploča *Greenbone Security Assistant*-a

Na nadzornoj ploči, u donjem redu s lijeve strane može se vidjeti distribucija CVE-a po godini nastanka, a s desne strane ukupan broj testova ranjivosti mreže (engl. *Network Vulnerability Tests* – NVT) koje OpenVAS sadrži u svojoj bazi podataka za skeniranje ranjivosti mreže. CVE je, kao što je već rečeno, popis unosa za javno poznate kibernetičke sigurnosne ranjivosti dok je NVT skripta koja se izvršava prema ciljanom sustavu i vrši provjere ranjivosti (daljinski ili lokalno), što također uključuje ranjivosti kojima je dodijeljen CVE. Sa slike 10 može se uočiti kako je za skeniranje korišteno 101 278 testova od kojih je 59 251 visoke ozbiljnosti, 34 654 srednje ozbiljnosti i 2 560 niske. U gornjem redu slike prikazan je pregled obavljenih zadataka, s lijeve strane prikaz po stupnju razine ozbiljnosti, a s desne status zadatka (može biti izvršen ili u tijeku).

6. Simulacija kibernetičkih napada na jezgru Linux operativnog sustava

Ideja ovog dijela rada zasniva se na simulaciji kibernetičkih napada na ciljni sustav iskorištavanjem identificiranih ranjivosti pomoću OpenVAS alata te eksploataciju sustava izvedenu pomoću ostalih dostupnih alata u Kali Linux-u. Pri tome ispitana je jednostavnost iskorištavanja ranjivosti za krađu podataka bitnih za korisnike sustava kao što su autentifikacijski podaci. Simulacija je provedena kroz 4 koraka:

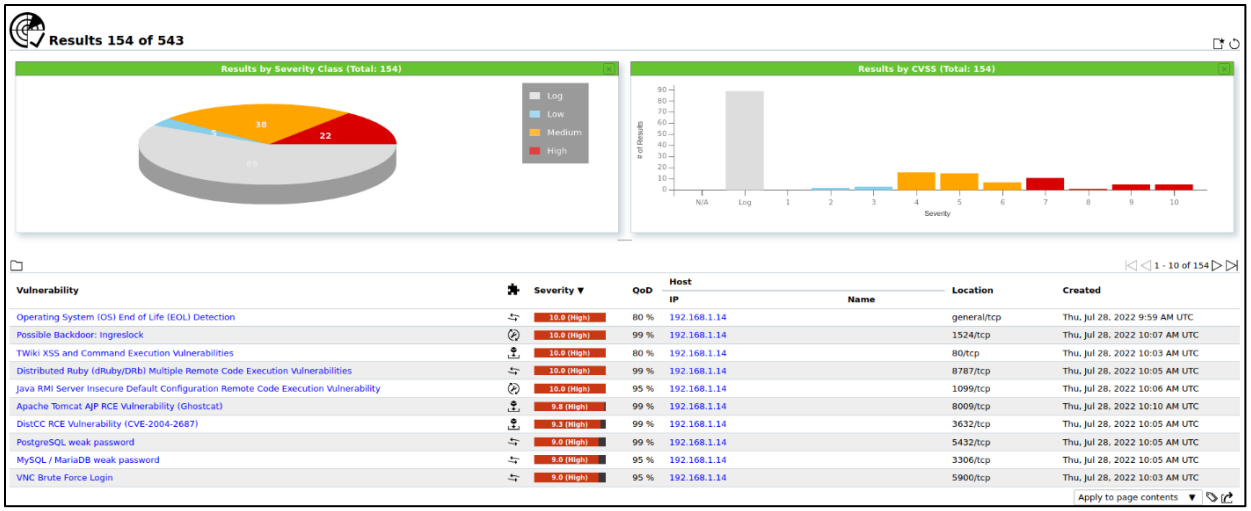
1. skeniranje sustava pomoću OpenVAS alata za traženje ranjivosti i otvorenih portova,
2. eksploatacija odabranih ranjivosti za dobivanje kontrole nad sustavom,
3. eskalacija ovlasti za dobivanje *root* pristupa te potpune kontrole nad sustavom,
4. dohvat osjetljivih podataka.

6.1. Skeniranje ranjivosti

OpenVAS pokrenut na Kali Linux-u, koristio se za pronalazak ranjivosti. Prije početka skeniranja na sučelju je bilo potrebno zadati parametre:

- naziv zadatka omogućuje da postavimo naziv pod kojim će skeniranje biti poznato,
- ciljni sustav može uključivati uređaje, portove i vjerodajnice,
- vrsta skenera može biti zadani OpenVAS skener ili CVE,
- konfiguracija skeniranja uključuje sedam različitih vrsta skeniranja koja se mogu odabrati i koristiti ovisno o tome koje informacije je potrebno prikupiti skeniranjem.

Nakon što je zadatak izrađen, moguće je na nadzornoj ploči pratiti napredak skeniranja. Ovisno o konfiguraciji skeniranja te raspoloživim resursima na uređaju na kojem je pokrenut OpenVAS, skeniranje može trajati par minuta ili par sati. U ovom slučaju skeniranje je završilo kroz tridesetak minuta. Rezultate skeniranja moguće je vidjeti na web sučelju, prikazanom na slici 11, ali i kroz generirano izvješće koje je dostupno u raznim formatima (PDF, XML, HTML, Latex).



Slika 11 Rezultati skeniranja *Metasploitable* virtualne mašine

Skeniranjem su prikupljene relevantne informacije o ciljnom sustavu koje će biti iskorištene za pronalaženje poznatih ranjivosti na mreži i izvan nje. Vidljivo je na slici 10 da je skeniranjem pronađeno 543 ranjivosti od kojih su 22 visoke, 38 srednje i samo 5 niske razine ozbiljnosti. Moguće je za svaku ranjivost vidjeti detalje: kratki sadržaj koji opisuje ranjivost, način na koji je otkrivena, njen potencijalni utjecaj na sustav, CVE identifikator, reference te način kako je riješiti. U tablici 2 prikazane su ranjivosti kritične razine ozbiljnosti, ocijenjene CVSS ocjenama 9 i 10 te u kojim se servisima nalaze, na kojem portu i kratki opis te ranjivosti.

Tablica 2 Kritične ranjivosti pronađene skeniranjem Metasploitable virtualne mašine

NVT	SERVIS	PORT	CVSS	Opis
Detekcija kraja životnog ciklusa OSa	Ubuntu Linux: verzija 8.04	-	**10.0**	Operativni sustav na sustavu došao je do kraja života i ne bi se trebao više koristiti.
Mogući backdoor[2]: Ingreslock	Metasploitable *root* ljuska	**1524**	**10.0**	Na sustavu je instaliran backdoor.
TWiki XSS I Ranjivosti u izvršavanju naredbi	Apache Http Server verzija 2.2.8	**80**	**10.0**	TWiki je sklon *Cross-Site Scripting* (XSS) napadima i ranjivostima u izvršavanju naredbi.
Ranjivosti višestrukog daljinskog	dRuby/DRb verzija 1.6	**8787**	**10.0**	Sustavi koji koriste distribuirani Ruby, koji je dostupan u Ruby verzijama 1.6 i novijim, mogu

[2] skrivena metoda za zaobilaženje provjere autentičnosti računalnih sustava za pristup sustavu bez znanja korisnika

izvršavanja distribuiranog Ruby koda				omogućiti neovlaštenim sustavima da izvršavaju distribuirane naredbe.
Apache Tomcat	Apache Jserv protocol (AJP) 1.3	**8180**	**9.8**	Apache Tomcat je sklon ranjivosti daljinskog izvršavanja koda u AJP konektoru.
DistCC ranjivost daljinskog izvršavanje koda	DistCC 2.x	**3632**	**9.3**	DistCC je sklon ranjivosti daljinskog izvršavanja koda.
PostgreSQL slaba lozinka	PostgreSQL 8.3.0 – 8.3.7	**5432**	**9.0**	Moguće je prijaviti se na udaljeni PostgreSQL koristeći slabe vjerodajnice.
MySQL/ MariaDB slaba lozinka	MySQL 5.0.51a	**3306**	**9.0**	Moguće je prijaviti se na udaljeni MySQL koristeći slabe vjerodajnice.
VNC Brute Force prijava	VNC protocol v1.3	**5900**	**9.0**	Moguće je prijaviti se s danim lozinkama putem VNC protokola.

Mnoge od ovih ranjivosti mogu se iskoristiti za dobivanje pristupa, međutim potrebno je prikupiti informacije o tome kako se one mogu iskoristiti. Postoji nekoliko izvora koji se mogu koristiti za pretraživanje gotovih skripti za iskorištavanje ranjivosti. Najpopularniji i najpoznatiji izvori su *exploit-db* baza podataka od *Offensive Security*, *searchsloit* izvan mrežna baza podataka te *Metasploit Framework,* radni okvir također razvijen od *Offensive Security* uključen u Kali Linux.

Metasploit je modularna platforma temeljena na programskom jeziku *Ruby* koja sadrži jedinstvenu zbirku eksploatacija. On je jedan od najkorisnijih alata za sigurnosnu reviziju koji su danas besplatno dostupni i može se lako prilagoditi korisničkim potrebama. Metasploit pruža impresivno radno okruženje širokog spektra eksploatacija komercijalne razine i opsežnog razvojnog okruženja eksploatacija, sve do mrežnih alata za prikupljanje informacija i skeniranje mrežnih aplikacija. Omogućuje automatizaciju testiranja, reviziju lozinki, socijalni inženjering, post eksploataciju, prikupljanje dokaza i izvješćivanje, [34].

6.2. Eksploatacija ranjivosti

U ovom dijelu rada prikazano je iskorištavanje dvije kritične ranjivosti za dobivanje kontrole nad sustavom. Na VirtualBox hipervizoru pokrenute su dvije virtualne mašine:

- Kali Linux sustav s kojega se provode napadi – napadač,
- Metasploitable sustav čije su ranjivosti iskorištene – žrtva.

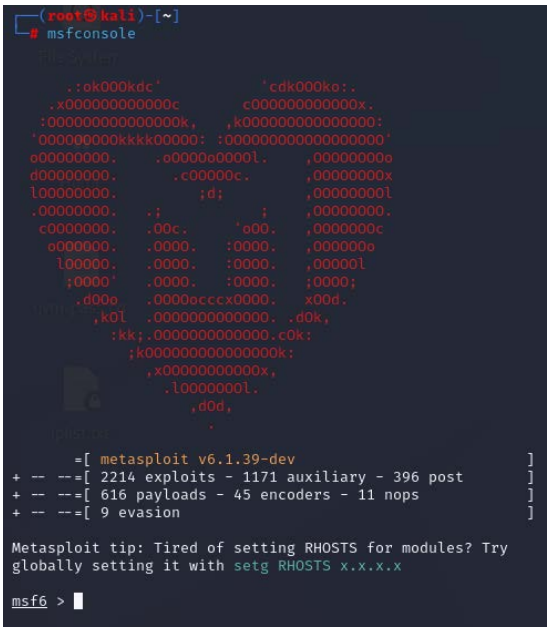
Prvi napad prikazuje iskorištavanje DistCC ranjivosti, a drugi ranjivosti Apache Tomcat poslužitelja.

6.2.1. Eksploatacija DistCC ranjivosti

DistCC (Distributed C/C++ Compiler) daemon je poslužitelj koji prihvaća i izvodi poslove kompilacije za mrežne klijente. *Daemon* je naziv za programe ili procese u Linux-u koji rade u pozadini, ali ostaju neaktivni dok se ne pozovu. DistCC dizajniran je da ubrza kompilaciju iskorištavanjem neiskorištene procesorske snage na drugim računalima. Uređaj s instaliranim DistCC *daemon*-om može poslati kod koji se kompajlira preko mreže na uređaj koji ima instaliran DistCC *daemon* i kompatibilni kompajler.

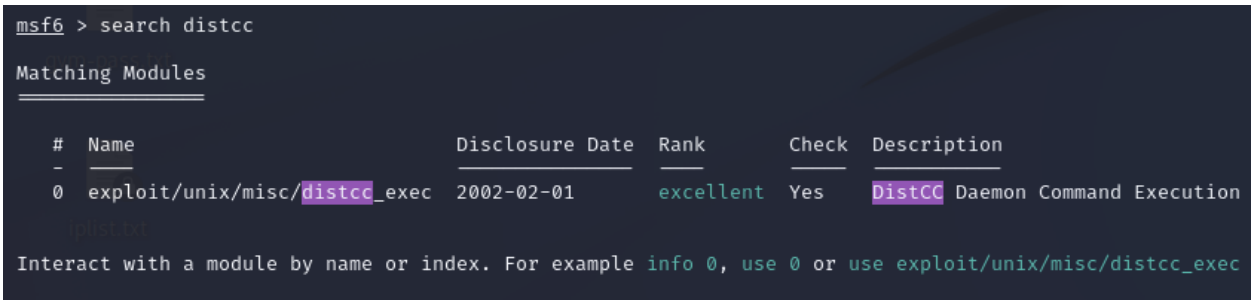
Iz OpenVAS izvješća jasno je da žrtva ima DistCC ranjivost (CVE 2004- 2687) na otvorenom portu 3632, preko kojeg će se odviti napad. DistCC 2.x, kada nije konfiguriran za ograničavanje pristupa portu poslužitelja, omogućuje udaljenim napadačima izvršavanje proizvoljnih naredbi putem poslova kompilacije, koje poslužitelj izvršava bez provjere autorizacije.

Na slici 12 prikazano je otvaranje terminala na Kali Linux-u sa *root* ovlastima i pokretanje Metasploit sučelja unošenjem naredbe *msfconsole*. *Msfconsole* pruža centralizirano sučelje zasnovano na konzoli koje omogućuje učinkovit pristup svim opcijama dostupnim u Metasploit-u. Pruža sve što je potrebno kako bi se pokrenula eksploatacija, učitali pomoćni (engl. *auxiliary*) moduli, izvela prebrojavanja ili stvorili slušatelji. To je jedini podržani način kako se pristupa većini značajki unutar Metasploita i to je najstabilnije Metasploit sučelje, [36].

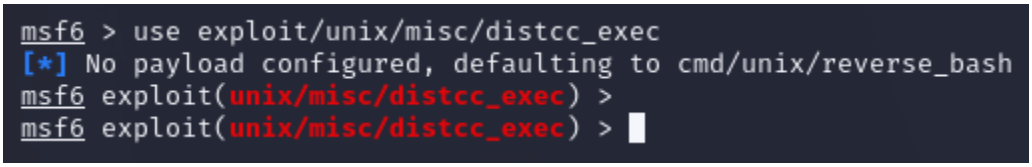


Slika 12 Pokretanje Metasploit Framework-a

Unošenjem naredbe *search distcc* radi se provjera Metasploit-a za modulima koji sadrže tu ključnu riječ. Kao što je vidljivo sa slike 13, modul za iskorištavanje ranjivosti ovog servisa je pronađen te ga je moguće iskoristiti za prodor u sustav.



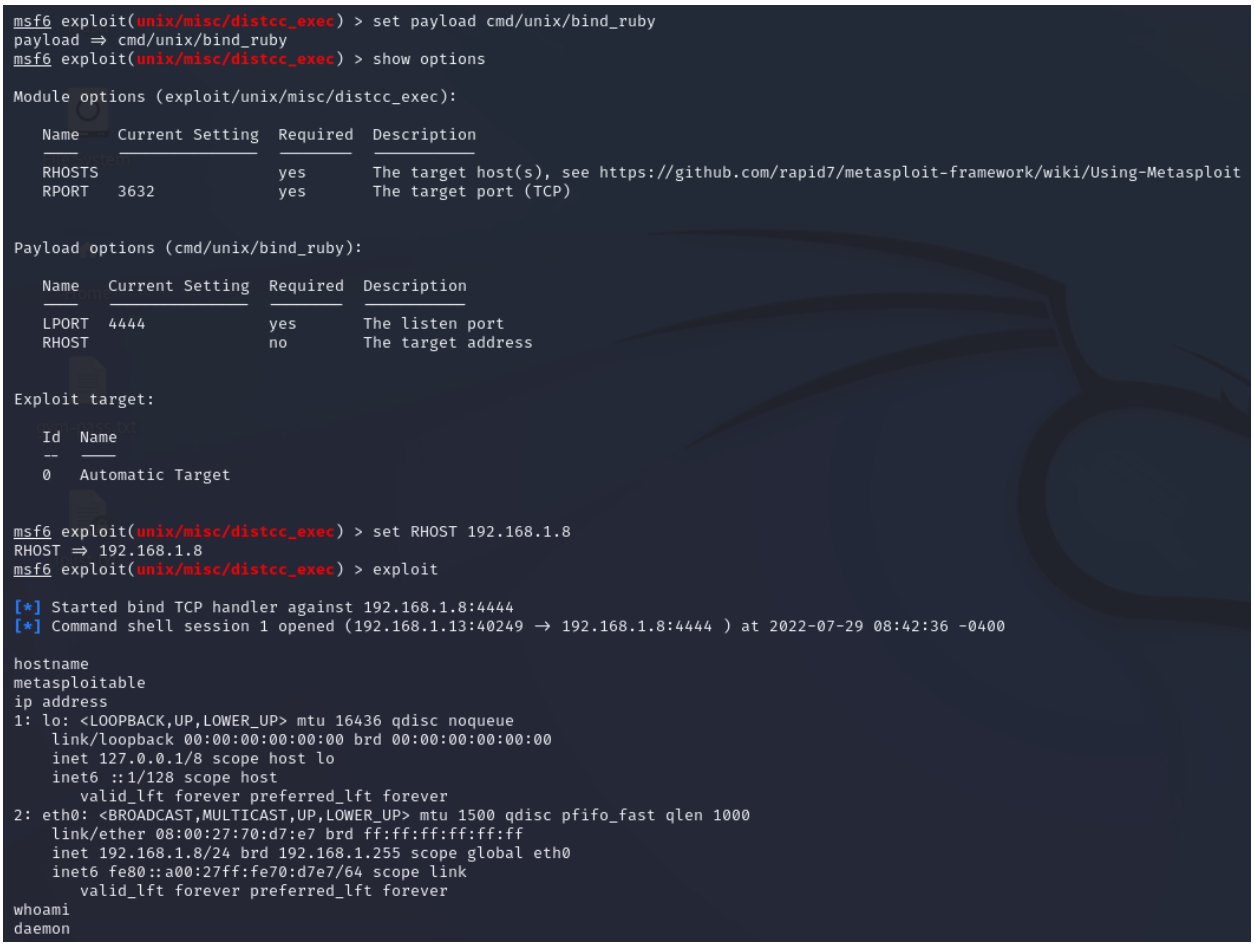
Slika 13 Pretraga Metasploit-a za DistCC modulom



Slika 14 Odabir DistCC modula

Na slici 14 prikazano je kako konzola javlja poruku da *payload* nije konfiguriran. *Payload*-i su proizvoljni kodovi koji se izvršavaju prilikom uspješne eksploatacije. Metasploit dozvoljava kombiniranje bilo kojih modula s bilo kojim *payload*-om i to je glavna prednost samog okvira jer

olakšava posao i napadaču i programerima modula i *payload*-a. Kao što je vidljivo na slici 15, u ovom je slučaju odabran modul *cmd/unix/bind_ruby*. Naredba *show options* prikazuje dostupne parametre za modul. Parametar RPORT je automatski postavljen na ciljani port 3632, a kao parametar RHOST postavljena je IP adresa žrtve koja je u ovom slučaju 192.168.1.8. Unosom naredbe *exploit* probija se u ljusku ciljanog sustava i uspostavlja se veza između napadača i žrtve.



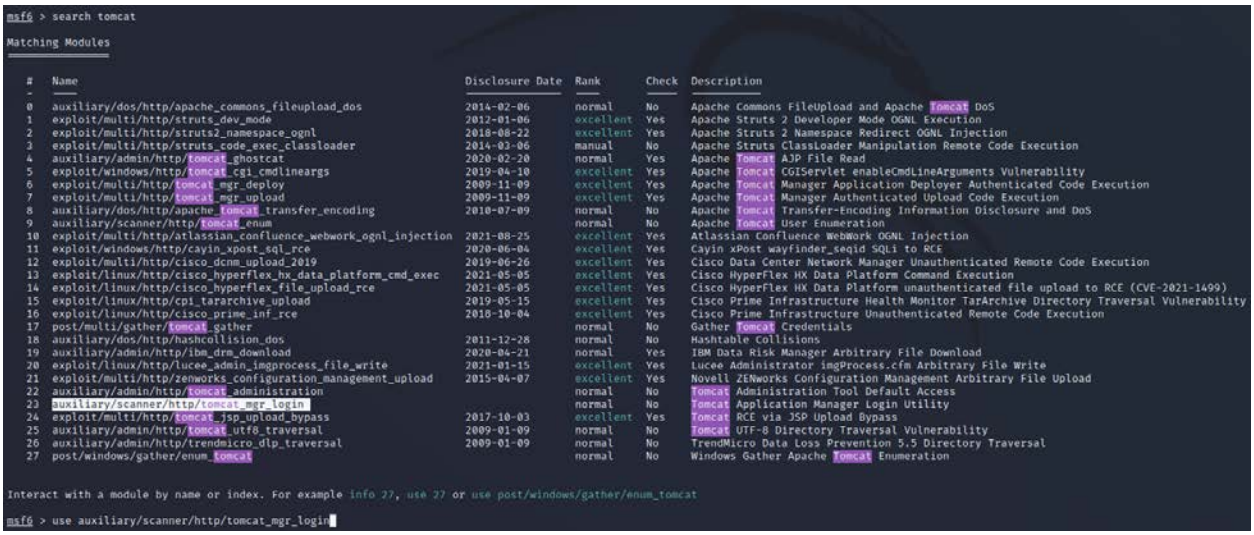
Slika 15 Konfiguracija i eksploatacija DistCC modula

Za provjeru uspjeha napada, upisuje se naredba *hostname* koja pokazuje da je ostvarena kontrola nad Metasploitable sustavom i naredbom *whoami* može se primijetiti da je korisnik *daemon* što znači da napadač ima pristup svim datotekama i direktorijima tog korisnika.

6.2.2. Eksploatacija Apache Tomcat ranjivosti

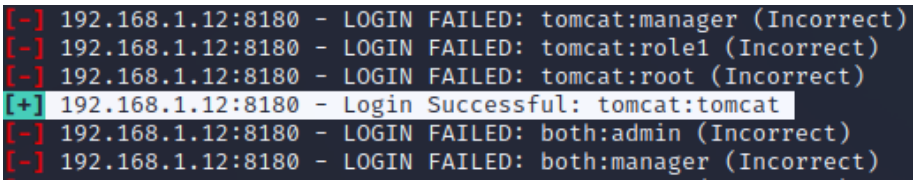
Apache je besplatan web poslužitelj otvorenog koda, odgovoran za prihvaćanje HTTP (engl. *The Hypertext Transfer Protocol*) zahtjeva od korisnika Interneta i slanje povratnih željenih informacija u obliku datoteka i web stranica. Apache je najčešće korišten web poslužitelj i radi na 67% svih web poslužitelja u svijetu jer se može lako prilagoditi okruženju, brz je, pouzdan i vrlo siguran te je zbog toga čest izbor mnogih tvrtki, [36]. Apache Tomcat pruža softver za pokretanje Java apleta na web pretraživačima.

Iz prethodno napravljenog OpenVAS skena vidljivo je da žrtva ima Apache Tomcat ranjivost (CVE 2020-1938) na otvorenom portu 8180 što znači da se preko tog porta može provesti napad korištenjem Metasploit-a i msf konzole. Naredbom *search tomcat* pretražuju se moduli koji sadrže ključnu riječ Tomcat, kao što je vidljivo sa slike 16.



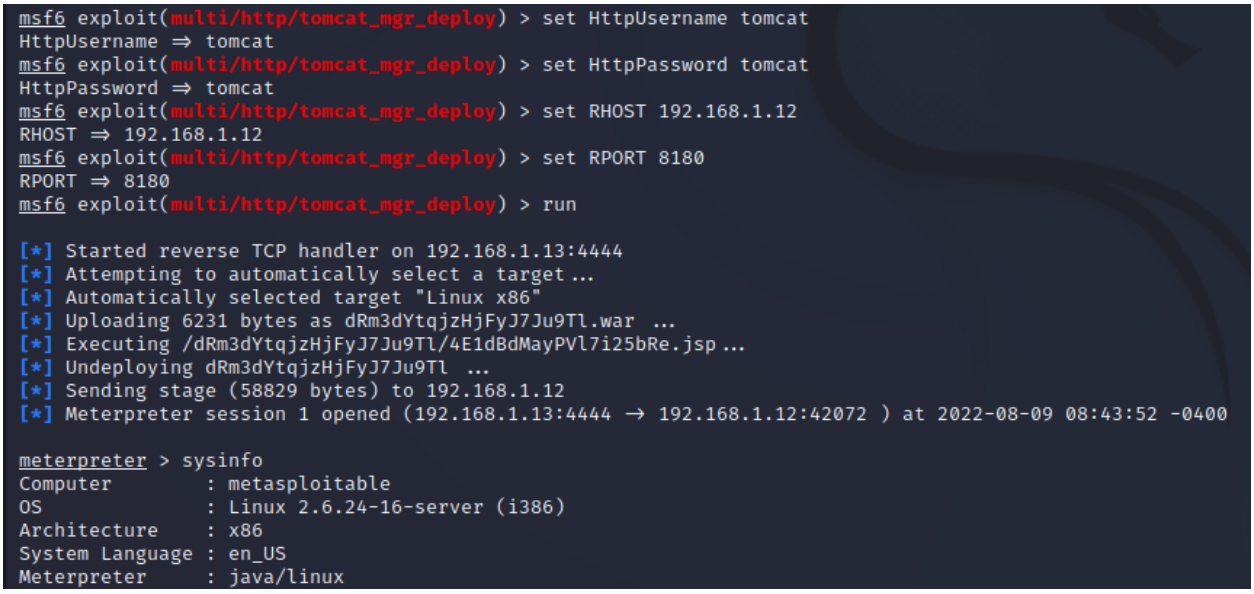
Slika 16 Pretraga Metasploit-a za Tomcat modulom

Odabirom modula *auxiliary/scanner/http/tomcat_mgr_login* pokušat će se eksploatirati sustav radi pronalaska vjerodajnica za prijavu. Ovaj modul jednostavno se pokušava prijaviti na instancu *Tomcat Application Manager*-a koristeći određeni spoj korisnika i lozinke. Slika 17 prikazuje ispis nekoliko isprobanih spojeva korisnika i korisničkih lozinki na ekranu što dovodi do zaključka da je ovaj modul jedan oblik brute force alata. Vidljivo je sa slike da je prijava bila uspješna s korisničkim imenom *tomcat* te lozinkom *tomcat*.



Slika 17 Vjerodajnice za prijavu

Nakon prikupljanja podataka za prijavu moguće je iskoristiti drugi modul dostupan na msf konzoli *exploit/multi/http/tomcat_mgr_deploy*. Ovaj modul iskorištava ranjivost autentificiranog daljinskog izvršenja koda. Ova ranjivost dopušta izvršavanje određenog *payload*-a na poslužitelju, koji je prethodno u njega učitan kao .war datoteka. Za izvođenje navedenog modula potrebno je postaviti korisničko ime i lozinku zajedno s ostalim opcijama, prikazanima na slici 18.



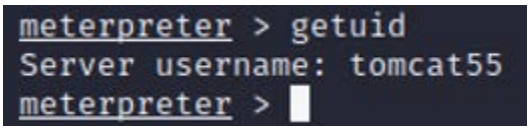
Slika 18 Eksploatacija ranjivosti daljinskog izvršenja koda

Ova eksploatacija omogućuje napadaču udaljeni pristup ljusci (engl. *remote shell*) sustava. Na slici 18 je vidljivo kako je pokrenuta Meterpreter sesija između napadača i žrtve te kako naredbom *sysinfo* Meterpreter konzola ispisuje informacije o eksploatiranom sustavu, poput imena sustava, imena i verzije OS-a, arhitekture i jezika.

Meterpreter je napredni posteksploatacijski program. Značajke su mu da ima naredbenu povijest, završetak kartice (engl. *tab completion*) i mnoge druge funkcionalnosti. Kada se pokušava eksploatirati udaljeni sustav, napadač obično pokušava pristupiti naredbenoj ljusci udaljenog

sustava, čime mu je omogućeno pokretanje proizvoljnih naredbi na tom sustavu. Napadač pokušava ostati neprimijećen, kao i izbjeći bilo kakve sustave za otkrivanje upada (IDS). Ako je eksploatacija uspješna, ali naredbena ljuska prestane raditi, njegove opcije bit će jako ograničene. To znači da pokretanje novog procesa na udaljenom sustavu predstavlja visoki rizik napadaču, gdje bi dobar administrator ili forenzički analitičar, koji bi prvo provjerio listu pokrenutih procesa na sumnjivom sustavu, primijetio novi proces. Meterpreter ima svoju vlastitu naredbenu ljusku, koja napadaču pruža široki spektar aktivnosti koje se mogu izvršiti na eksploatiranom sustavu, kao što su učitavanje i izvršavanje vlastitih programa na udaljenim sustavima. Povrh toga, Meterpreter radi tako da sam sebe ubrizga u ranjive pokrenute procese na udaljenom sustavu, kada se pojavi eksploatacija što znači da sve naredbe idu kroz Meterpreter i također se izvršavaju unutar konteksta pokrenutog procesa. To omogućuje izbjegavanje otkrivanja napada od bilo kakvog antivirusnog sustava ili osnovnih forenzičkih ispitivanja, [36].

Podaci o trenutno prijavljenom korisniku dohvaćaju se naredbom *getuid* te se sa slike 19 može vidjeti da je napadač prijavljen kao korisnik *tomcat55* što znači da ima sve ovlasti tog korisnika.



Slika 19 Meterpreter ljuska

6.3. Eskalacija ovlasti

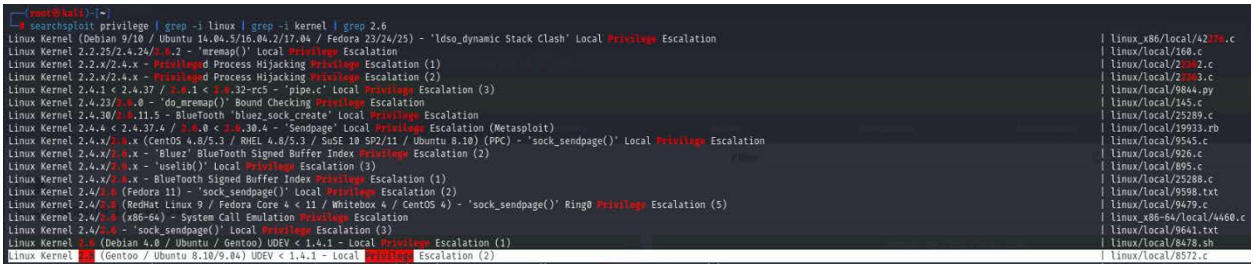
Eksploatacijom ranjivosti ostvaren je pristup Metasploitable-u, ali za dobivanje potpune kontrole nad sustavom potrebno je ostvariti takozvani *root* pristup, pristup visoke razine. To se može postići eskalacijom ovlasti s korisnika na superkorisnika. Korijenski korisnik (engl. *root user*), superkorisnik (engl. *super user*) ili administrator (engl. *admin*) je poseban korisnički račun u OS-u koji ima neograničene ovlasti.

Ranjivosti u jezgri Linux-a otkrivene su s vremena na vrijeme. Napadači mogu iskoristiti ove ranjivosti kako bi dobili *root* pristup Linux sustavu. U nastavku će biti prikazana dva načina eskalacije ovlasti iskorištavanjem ranjivosti jezgre Linux-a.

6.3.1. Netlink ranjivost

Korištenjem *Metasploit Framework*-a ostvaren je pristup naredbenoj ljusci na ciljnom sustavu. Za eskalaciju ovlasti potrebno je pronaći i iskoristiti ranjivosti jezgre koja se nalazi na ciljnom sustavu. Sa slike 18 vidljivo je da sustav ima jezgru verzije 2.6.24 i pokreće Ubuntu verzije 8.04.

Kali sadrži lokalnu verziju *exploit-db* kojoj se može pristupiti pomoću alata pod nazivom *SearchSploit*. Pokretanjem naredbe *searchsploit* s terminala sa zadanim parametrima *privilege*, *linux*, *kernel*, 2.6 dobiva se popis skripti za lokalnu eskalaciju ovlasti ove verzije jezgre. Za provedbu napada odabrana je 8572.c skripta, kao što se može vidjeti na slici 20.



Slika 20 Ispis naredbe *searchsploit*

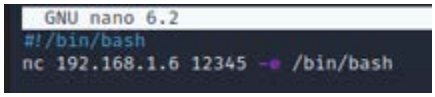
Ova eksploatacija iskorištava grešku u *udev* upravitelju uređaja, dopuštajući izvršavanje koda putem neprovjerene Netlink poruke (CVE 2009-1185). Netlink se koristi za prijenos informacija između procesa jezgrinog i korisničkog prostora, a *udev* je sustav korisničkog prostora koji upravlja uređajima. Glavna svrha *udev*-a je djelovati na otkrivanje uređaja i uključivanje u radno stanje, uključujući akcije koje vraćaju kontrolu jezgri, npr. učitavanje modula jezgre ili ugrađenog programa (engl. firmware) uređaja, [44].

Za uspješnu eksploataciju ove ranjivosti, potrebno je postaviti 8572.c skriptu na ciljani sustav i izvršiti je. Slika 21 prikazuje kopiranje skripte u direktorij poslužitelja koji je pokrenut na Kali-u. Tako će skripta biti dostupna za preuzimanje na ciljni sustav.

Slika 21 Kopiranje skripte naredbom cp

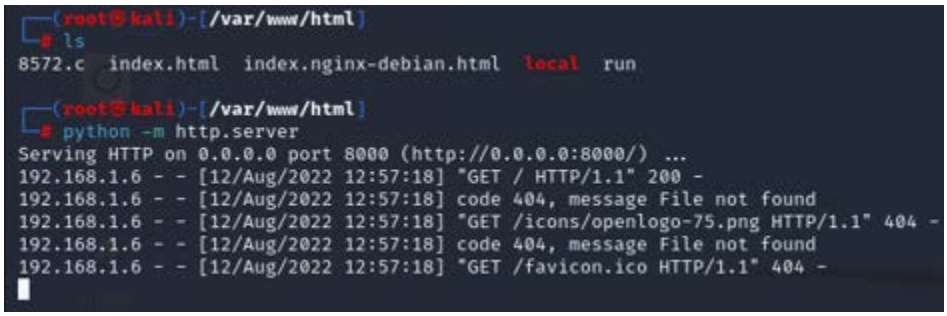
Ova skripta će izvršiti datoteku /tmp/run na žrtvi, tako da je potrebno stvoriti tu datoteku. Slika 22 prikazuje kod izvršne datoteke, spremljena pod nazivom run. Kada se ova datoteka izvrši,

pokrenut će se Netcat, uslužni program naredbenog retka koji čita i piše podatke putem mrežnih veza. Netcat se koristi za stvaranje *backdoor* pristupa preko porta 12345. Opcija *-e /bin/bash* služi za otvaranje ljuske nakon uspostavljanja veze s Kali mašinom.



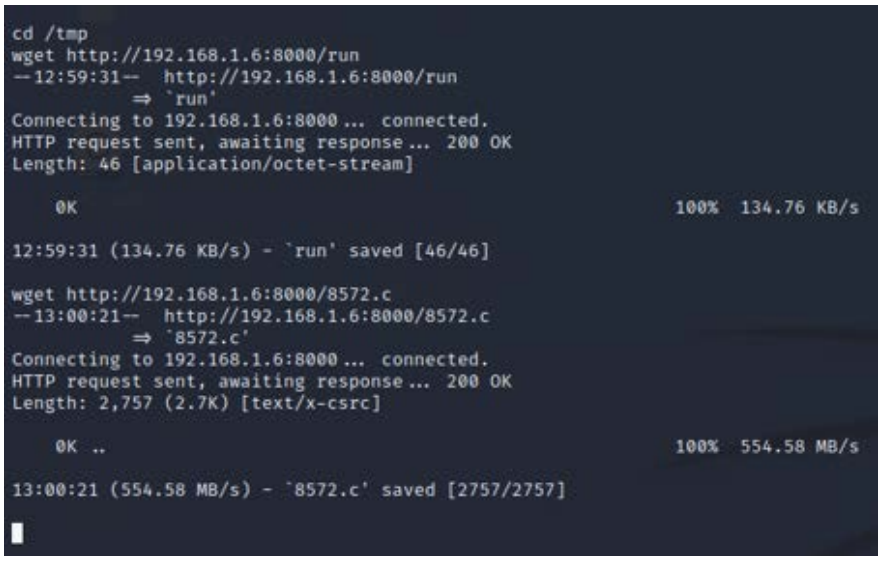
Slika 22 Kod skripte za Netcat povezivanje

Skripte 8572.c i run potrebno je postaviti na mjesto s kojeg ih žrtva može preuzeti stoga će na Kali-u biti pokrenut web poslužitelj. Na slici 23 vidljivo je stvaranje jednostavnog HTTP poslužitelja pomoću *Python*-a na Kali-u u direktoriju koji sadrži prethodno spomenute skripte.



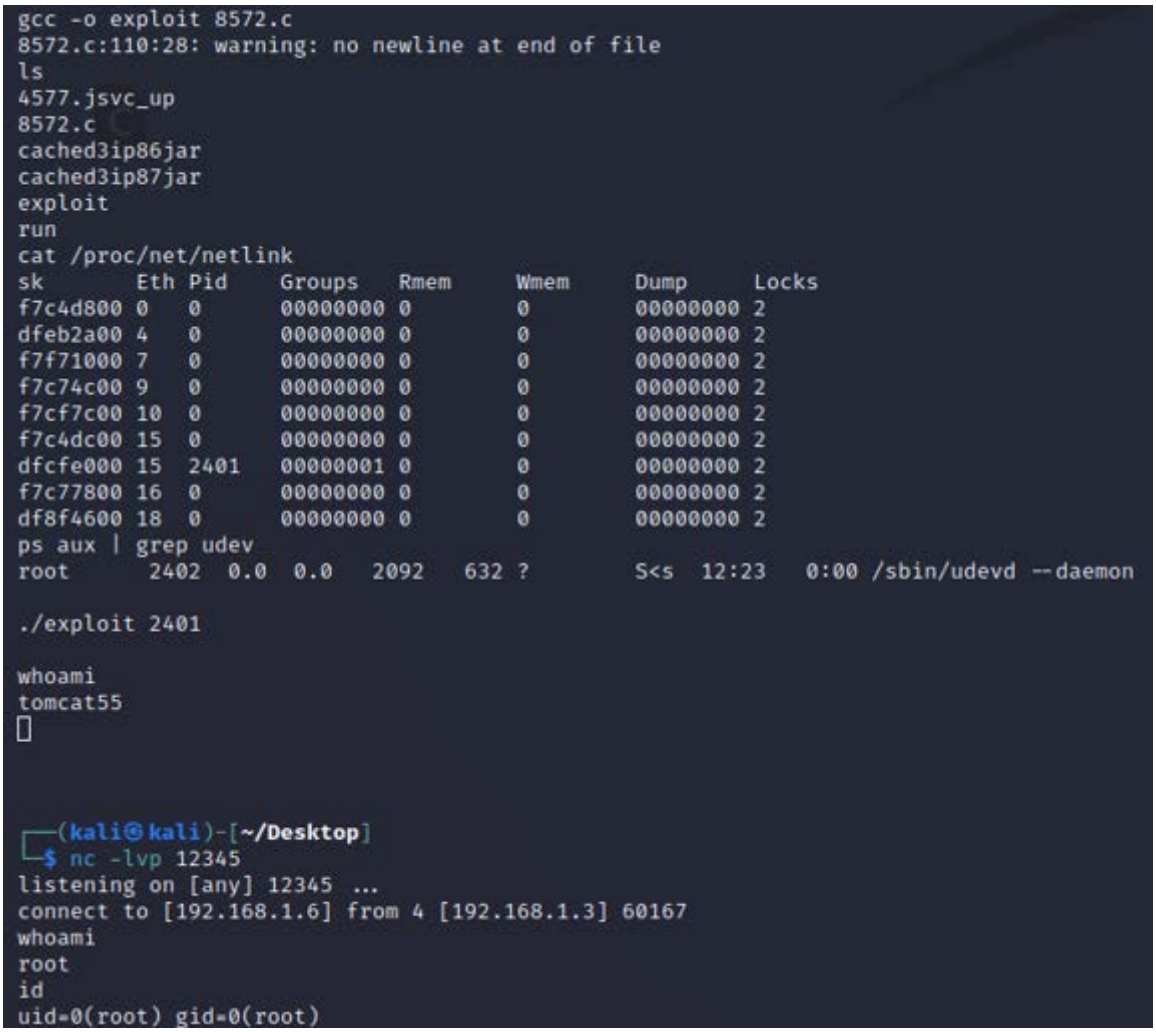
Slika 23 Pokretanje web poslužitelja

Prethodnom eksploatacijom ranjivosti Apache Tomcat, ostvaren je pristup naredbenoj ljusci na ciljnom sustavu. Unutar nje potrebno je pozicionirati se u direktorij */tmp* i upotrijebiti naredbu *wget* za povezivanje s poslužiteljem na Kali-u te za prijenos zlonamjernih skripti na ciljni sustav, kao što je prikazano na slici 24.



Slika 24 Preuzimanje zlonamjernih skripti na Metasploitable sustav

Skripte potrebne za provedbu napada uspješno su prenesene na žrtvu, no potrebno ih je izvršiti. Budući da je 8572.c datoteka kodirana u programskom jeziku C, potrebno ju je prevesti u izvršnu datoteku. To je moguće učiniti na Linux sustavima koristeći GCC (*GNU Compiler Collection*). Na slici 25 naredba *gcc* izvodi kompajliranje 8572.c datoteke u izvršnu datoteku, koristeći -o oznaku za određivanje naziva izlazne datoteke, u ovom slučaju *exploit*. Naredba *ls* koristi se za provjeru uspješne kompilacije izvršne datoteke. Prije izvršavanja skripte, potrebno je pronaći identifikator procesa (engl. process identifier – PID) Netlink utičnice, što je obično PID UDEVD procesa minus jedan, što se može učiniti naredbom *cat /proc/net/netlink*, a jedini PID različit od nule je taj broj. Radi se provjera ispravnosti pokretanjem naredbe *ps aux | grep udev* te bi taj broj trebao biti jedan broj veći.



Slika 25 Eksploatacija sustava i pokretanje Netcat sesije

Kako bi se uspostavila Netcat sesija, na Kali-u je potrebno postaviti slušatelja tako da kada se izvrši skripta run, napadač može doći do ljuske. U donjem terminalu na slici 25, naredba *nc -lvp* 12345 služi za slušanje dolaznih veza. Sada kada je slušatelj spreman, moguće je izvršiti exploit datoteku predajući joj PID Netlink-a. Nakon nekoliko trenutaka, sesija se pokreće na Netcat slušatelju i moguće je izvršiti naredbe kao što su *id* i *whoami* kako bi potvrdili uspješnost napada. Sa slike 25 je vidljivo da je postignut *root* pristup, i odavde, je u osnovi moguće raditi razne zlonamjerne aktivnosti na sustavu.

6.3.2. Dirty COW ranjivost

Dirty COW ranjivost (CVE 2016-5195) je ranjivost eskalacije ovlasti, a uzrokovana je stanjem nadmetanja koje se nalazi u načinu na koji memorijski podsustav Linux jezgre rukuje prekidom

kopiranja na pisanje (copy-on-write – COW) privatnih mapiranja memorije koja je samo za čitanje. Greška postoji od verzije jezgre 2.6.22 (objavljena 2007. godine) i popravljena je 2016. godine u verzijama 4.8.3, 4.7.9, 4.4.26, [38].

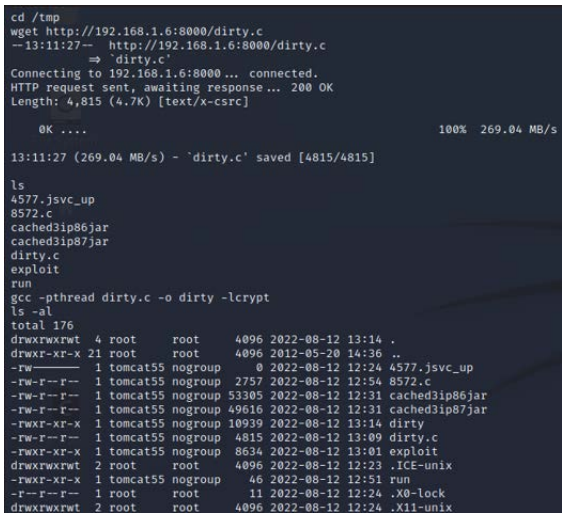
Neovlašteni lokalni korisnik mogao bi iskoristiti ovu grešku kako bi dobio pristup pisanju u dio memorije koja je inače samo za čitanje i tako povećao svoje ovlasti na sustavu. Napadači tako mogu dobiti *root* ovlasti iskorištavanjem ranjivosti te mogu modificirati bilo koju zaštićenu datoteku, iako su te datoteke samo za čitanje. U nastavku bit će prikazano kako se ova ranjivost može iskoristiti.

Potrebno je preuzeti skriptu dirty.c sa web stranice: https://github.com/FireFart/dirtycow/blob/master/dirty.c. Izvršavanjem ove skripte bit će moguće prijaviti se u sustav s novostvorenim korisnikom koji posjeduje *root* ovlasti.

Slika 26 Kopiranje dirty.c skripte

Nakon preuzimanja skripte potrebno ju je smjestiti u direktorij poslužitelja */var/www/html*. Taj postupak prikazan je na slici 26.

Da bi bio napad bio uspješan, napadač već mora imati ostvaren pristup sustavu prije nego što može iskoristiti ovu ranjivost. Prethodnom eksploatacijom Apache Tomcat ranjivosti je ostvaren pristup naredbenoj ljusci sustava. Putem nje, naredbom *wget* izvršeno je preuzimanje dirty.c datoteke, prikazano na slici 27. Potom je potrebno datoteku kompajlirati u izvršnu datoteku, koristeći naredbu *gcc -pthread dirty.c -o dirty -lcrypt*. Za potvrdu uspjeha kompilacije ispisuju se datoteke u direktoriju naredbom *ls -al*.



Slika 27 Preuzimanje i kompilacija dirty.c skripte

Pokretanjem izvršne datoteke *dirty* iniciraju se idući događaji, prikazani na slici 28:

- izvorna datoteka /etc/passwd se sigurnosno kopira u /tmp/passwd.bak,
- generira se novi redak u koji napadač upisuje novu lozinku,
- stvara se novi korisnik *firefart* koji ima *root* ovlasti i generiranu lozinku,
- ispis poruke koja govori da je stvoren novi korisnik i ispis njegovih vjerodajnica.

Slika 28 Eksploatacija Dirty-COW ranjivosti

Eksploatacijom ove ranjivosti omogućen je pristup sustavu sa *root* ovlastima. Napadač može kreirati trajni *backdoor* i time zadržati pristup kompromitiranom sustavu, čak i u slučaju prestanka rada eksploatirane usluge ili primjene sigurnosne zakrpe.

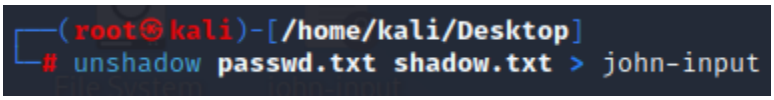
6.4. Dohvat osjetljivih podataka

Sada kada je ostvaren pristup žrtvi i to pristup visoke razine, kao superkorisnik moguće je dostupiti osjetljivim podacima korisnika poput autentikacijskih podataka. U nastavku će biti prikazan jedan od načina na koji je moguće izdvojiti i zatim razbiti *hashove* lozinki s lokalnog stroja. Za otkrivanje korisničkih računa i lozinki potrebne su dvije datoteke iz sustava žrtve:

- /etc/passwd - sadrži podatke o korisnicima,
- /etc/shadow - sadrži odgovarajuće hashove lozinki za korisnike.

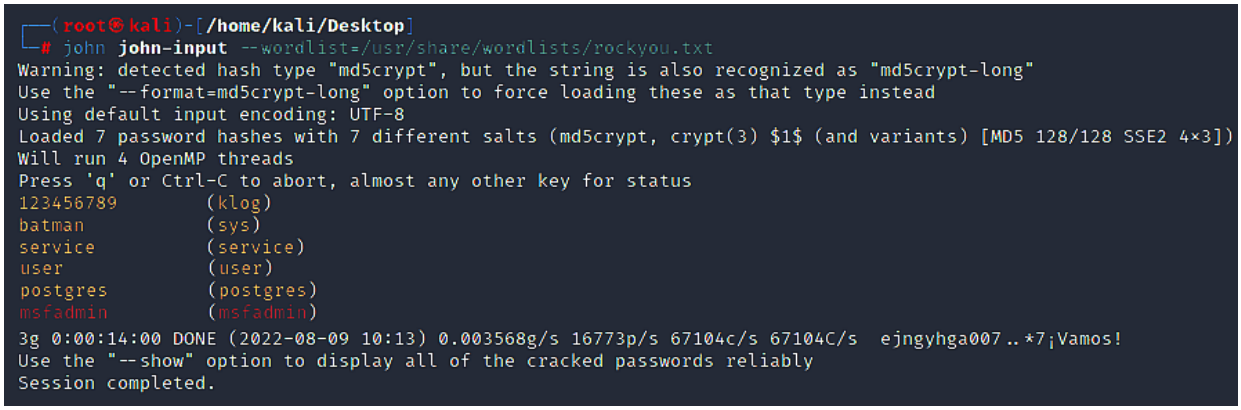
Superkorisnik ima ovlasti za dohvat i čitanje tih datoteka. Za probijanje lozinki koristit će se alat John the Ripper, unaprijed instaliran na Kali Linux-u. John the Ripper popularan je alat za otkrivanje lozinki temeljen na rječniku. Koristi popis riječi pun lozinki, a zatim pokušava probiti zadanu *hash* lozinku koristeći svaku od lozinki s popisa riječi. Drugim riječima, to je *brute force* način probijanja lozinki. To je također metoda koja troši najviše vremena i resursa. Što je više lozinki za isprobati, više je vremena potrebno za njihovo otkrivanje, [39].

Potrebno je spremiti *passwd* i *shadow* datoteke kao tekstualne te s naredbom *unshadow* spojiti podatke iz jedne i druge datoteke za stvaranje jedne datoteke s detaljima korisničkog imena i lozinke. Primjer korištenja *unshadow* naredbe prikazan je na slici 29.



Slika 29 Primjer korištenja *unshadow* naredbe

Naredbom john pokreće se alat te mu se kao parametar predaje *john-input* datoteka. Kao popis riječi koristi se popis lozinki *rockyou.txt* koji dolazi s alatom te je u naredbi potrebno samo navesti putanju do njega. Na slici 30 prikazano je pokretanje alata te otkrivanje lozinki i pripadnih korisničkih imena.



Slika 30 Primjer korištenja john alata za brute force lozinki

Sa slike 30 je vidljivo da je od sedam postojećih korisnika *john* otkrio njih šest, znači sve osim *root* lozinke.

7. Analiza rezultata istraživanja i prijedlozi unaprjeđenja razine sigurnosti Linux operativnog sustava

OS je najbitniji softver instaliran na uređaju. On upravlja njegovom memorijom i procesima, kao i svim njegovim hardverom tako da je bitno odabrati kvalitetan i siguran OS pošto je on ključna odrednica sigurnosti uređaja spojenih na mrežu. Desetljećima je Linux obitelj OS-a osvajala naklonost tehničkih stručnjaka zbog svoje jednostavnosti i funkcionalnosti. Opći konsenzus među stručnjacima je da je Linux vrlo siguran OS, nedvojbeno najsigurniji OS po dizajnu, ali nipošto nije u potpunosti siguran od kibernetičkih napada, [40].

7.1. Analiza rezultata istraživanja

U prethodnoj cjelini rada simulacijom su demonstrirani kibernetički napadi koji prikazuju iskorištavanje ranjivosti Linux sustava. OpenVAS alatom skeniran je sustav te su tako pronađene brojne ranjivosti različitog stupnja rizika.

Tablica 3 Sinteza rezultata istraživanja

Napad	Rezultat napada	Posljedice za žrtvu
Eksploatacija DistCC ranjivosti	Omogućen udaljeni pristup naredbenoj ljusci računala žrtve kao *daemon* korisnik	• Udaljeno izvršavanje naredbi od strane napadača • Uvid u informacije o sustavu žrtve • Udaljena pretraga sadržaja na računalu • Otuđivanje dostupnih podataka
Eksploatacija Apache Tomcat ranjivosti	Omogućen udaljeni pristup Meterpreter ljusci na računalu žrtve kao *tomcat* korisnik	• Uvid u trenutno aktivne procese na računalu • Neželjena izmjena pohranjenog sadržaja • Zrcaljenje zaslona žrtve

Eksploatacija Netlink ranjivosti	Omogućen udaljeni *root* pristup	• Izmjena dopuštenja u cilju brisanja ili krađe podataka • Dodavanje ili brisanje korisnika • Napadač ima pristup sistemskim datotekama i može uzrokovati smetnje u radu • Stvaranje *backdoor* pristupa za buduće napade • Implementacija malicioznog softvera
Eksploatacija Dirty COW ranjivosti	Omogućen udaljeni *root* pristup	

U tablici 3 prikazana je sinteza rezultata provedenih eksploatacija ranjivosti te su vidljivi ostvareni ishodi napada, kao i potencijalne posljedice za žrtvu ako se planirani napadi uspješno realiziraju.

Odabrane su dvije ranjivosti za eksploataciju: DistCC ranjivosti i ranjivost Apache Tomcat poslužitelja. Pokazano je kako je korištenjem Metasploit Frameworka moguće relativno lako iskoristiti ranjivosti te doći do kontrole nad sustavom kroz naredbenu ljusku. Eksploatacijom Apache Tomcat ranjivosti prikazana je kontrola nad sustavom preko Meterpreter sesije.

Nakon uspješne eksploatacije mete, ovisno o modulu koji se koristi, otvara se sesija kroz ljusku ili Meterpreter ljusku. Meterpreter ljuska daje pristup Metasploit modulima i drugim radnjama koje nisu dostupne u naredbenoj ljusci dok naredbena ljuska otvara standardni terminal na meti, omogućujući slične funkcije kao i terminal na OS-u. Funkcionalnost se može razlikovati ovisno o načinu eksploatacije.

Svaki račun u sustavu ima određenu razinu ovlasti. Standardni korisnici obično imaju ograničen pristup bazama podataka sustava, osjetljivim datotekama ili drugim resursima. Ostvarivanjem pristupa sustavu s korisničkim računom niske razine napadač ima polaznu točku u ranjivom sustavu. Idući korak je obično eskalacija ovlasti na višu razinu od računa koji je prvobitno

bio ugrožen jer to omogućuje izvedbu ozbiljnijih zlonamjernih aktivnosti. Na primjer, eskalacija ovlasti može transformirati jednostavnu infekciju zlonamjernim softverom u katastrofalnu povredu podataka.

Eskalacija ovlasti omogućuju napadačima otvaranje novih vektora napada na ciljni sustav. Na primjer, može uključivati, [41]:

- dobivanje pristupa drugim povezanim sustavima
- postavljanje dodatnih zlonamjernih sadržaja na ciljni sustav
- podešavanje sigurnosnih postavki ili ovlasti
- dobivanje pristupa aplikacijama ili podacima na sustavu izvan privilegija izvornog ugroženog računa
- dobivanje *root* pristupa ciljnom sustavu ili cijeloj mreži

U prethodnoj cjelini prikazana su dva načina eskalacije ovlasti kroz iskorištavanje ranjivosti Linux jezgre. U prvom napadu prikazana je eksploatacija Netlink ranjivosti pomoću skripti koje su prenesene i pokrenute na ciljnom sustavu, te je pomoću njih i netcat alata ostvaren *root* pristup sustavu. Drugi napad prikazuje eksploataciju Dirty COW ranjivosti koja je uzdrmala cijelu Linux zajednicu jer je postojala devet godina prije nego što je otkrivena i zakrpana. Ranjivost je utjecala na sve OS-e temeljene na Linux jezgri, uključujući Android, a njezina je posljedica vrlo ozbiljna. Napadači mogu dobiti *root* ovlasti iskorištavanjem ove ranjivosti, a način kojim se to postiže je prikazan u prethodnoj cjelini.

Na kraju prikazana je jedna od posljedica ovakvih kibernetičkih napada. Neovlašteni korisnik ima omogućen udaljeni pristup naredbenoj ljusci sustava, ima ovlasti superkorisnika te je u mogućnosti kompromitirati podatke na sustavu, bili oni samo za čitanje ili za pisanje. U ovom radu prikazan je dohvat zaporki iz ciljnog sustava te njihovo probijanje alatom John the Ripper. Time su dočarane potencijalne posljedice za žrtvu nakon uspješne realizacije napada. Osim dohvata osjetljivih podataka napadač je u mogućnosti istražiti datotečni sustav na meti, stvoriti ili manipulirati korisničkim računima, uređivati datoteke i zapisnike te mnoge druge zlonamjerne radnje.

7.2. Prijedlozi unaprjeđenja razine sigurnosti Linux operativnog sustava

Kibernetička sigurnost obuhvaća skup procesa, mjera i standarda kojima se jamči određena razina pouzdanosti pri korištenju proizvoda i usluga u kibernetičkom prostoru, pri čemu sustavna zaštita računala i računalnih mreža, informatičke i informacijske infrastrukture, mobilnih uređaja i podataka od malicioznih napada tome značajno pridonosi, [42].

Mnogo je istraživanja usmjereno na sigurnost Linux OS-a te je moguća kategorizacija na dva glavna pristupa, [43]:

1. Očvršćivanje OS-a (engl. *operating system hardening*)
2. Implementacija sigurnosnih poboljšanja (engl. *security enhancement*)

Kada je riječ o kibernetičkoj sigurnosti, korisnici Linux-a su u značajnoj prednosti u odnosu na korisnike Windows ili Mac OS-a. Za razliku od tih vlasničkih OS-a, Linux na mnogo načina ima sigurnost ugrađenu u svoj osnovni dizajn, zbog prirode otvorenog koda te visoke fleksibilnosti, podesivosti i raznolikosti. Transparentnost izvornog koda Linux-a znači da su ranjivosti u njemu gotovo uvijek kratkotrajne jer se on podvrgava stalnom, temeljitom pregledu od strane globalne zajednice te kao rezultat toga, sigurnosne ranjivosti Linux-a općenito se identificiraju i uklanjaju vrlo brzo, [40]. Stoga je najbolji način za očuvanje visoke razine sigurnosti praćenje sigurnosnih izvješća i instaliranje sigurnosnih ažuriranja i zakrpa pravovremeno. S vremena na vrijeme otkriju se ranjivosti u jezgri Linux-a i zato je važno pravovremeno reagirati.

Visoka razina raznolikosti slijedi kao rezultat mnogih dostupnih distribucija Linux-a te različitih arhitektura sustava i komponenti koje one sadrže. Ova raznolikost ne samo da pomaže zadovoljiti individualne zahtjeve korisnika, ona također pomaže u zaštiti od napada tako što zlonamjernim korisnicima otežava kreiranje učinkovitih eksploatacija koje mogu biti iskorištene protiv širokog spektra Linux sustava. Uz raznolikost dizajna koja se vidi u Linuxu, određene sigurne distribucije Linux-a razlikuju se tako da specifično rješavaju napredne probleme sigurnosti i privatnosti.

7.2.1. Očvršćivanje operativnog sustava

Proces stvaranja sigurnijeg sustava od razine koju nudi zadani instalirani OS poznat je kao očvršćivanje sustava. Sigurnost OS-a ovisi o velikom broju konfiguracijskih postavki na razini

OS-a i na aplikacijskoj razini. Nadalje, Linux i njegova jezgra vrlo su složeni i nije ih lako konfigurirati. Linux sustavi mogu se konfigurirati gotovo beskonačno, a suptilne promjene konfiguracije mogu imati značajne sigurnosne implikacije, [45].

Postoji mnoštvo tehnika kojima se može očvrstiti sustav, u nastavku će biti navedene neke od najčešćih.

1) Sigurnost pokretanja i pokretačkih programa

Većina Linux sustava koristi jedan od dva pokretača, *Linux Loader* (LILO) ili *Grub*. Ovi pokretački programi određuju koja se jezgra pokreće kada se sustav pokrene ili ponovno pokrene. Učitavaju se nakon što osnovni ulazno/ izlazni sustav (engl. *Basic Input/ Output System* – BIOS) inicijalizira sustav. Prema zadanim postavkama i LILO i Grub će omogućiti pokretanje u *single-user* načinu rada. U tom načinu rada korisnik ima *root* ovlasti bez potrebe za unosom *root* lozinke. Osim toga, zlonamjerni korisnik može unijeti niz drugih parametara u naredbeni redak oba pokretačka programa što mu može pružiti priliku da ugrozi sustav. Međutim i LILO i Grub imaju opciju zaštite lozinkama da bi se to spriječilo, [45].

2) Jake i jedinstvene lozinke

Gotovo svi korisnici imaju jedan cilj pri odabiru lozinke, odabrati onu koju će lako zapamtiti. Sigurnost se jednostavno ne uzima u obzir. Redovito mijenjanje lozinke za njih je neugodnost i muka, ali to je ključna aktivnost za stalnu sigurnost sustava. Osim poznatih pravila za generiranje jakih lozinki, u Linux-u postoji mogućnost kontrole karakteristika korisničkih lozinki putem PAM-a (engl. *Pluggable Authentication Modules*). Cilj PAM-a je pružiti fleksibilan mehanizam za autentikaciju korisnika kojeg konfigurira administrator sustava. Pri korištenju ovog pristupa, uslužni programi pozivaju različite module za provjeru autentičnosti tijekom izvođenja kako bi izvršili stvarni postupak provjere valjanosti korisnika, a uslužni programi zatim djeluju prikladno ovisno o rezultatima koje im moduli vraćaju, [46].

3) Sigurnost SSH (engl. *Secure Shell*).

SSH je najčešće korišten alat za udaljenu administraciju i upravljanje Linux poslužiteljima. Iako je izvrstan alat za administraciju i upravljanje, također može poslužiti napadačima da steknu potpunu kontrolu nad Linux poslužiteljem. Stoga je vrlo važno da SSH bude ispravno zaštićen.

Prema konfiguraciji SSH-a, on koristi zadani port 22, koji je poznat svim napadačima. Jedna od opcija je pokretanje SSH procesa na nekom drugom proizvoljno odabranom portu. Također SSH koristi autentikaciju lozinkom, tako da je dobra praksa koristiti jake lozinke kao zaštitu od *brute force* napada. Nadalje, *root* prijava je omogućena prema zadanim postavkama, ali uvijek je bolje upotrijebiti uobičajeni korisnički račun za iniciranje veze, koristeći *sudo* naredbu. Izravne *root* prijave mogu rezultirati neželjenim posljedicama za korisnika, [47].

4) Sigurnost vatrozida

Vatrozid je jedan od najvažnijih elementa obrane od napada. U mnogim slučajevima vatrozid je prva linija obrane od napada na sustav. Vatrozid pomaže u obrani na tri glavna načina: obrađivanje neželjenog dolaznog prometa, obrađivanje neželjenog odlaznog prometa i bilježenje sumnjivog prometa ili prometa za koji se zna da ima zlonamjernu namjeru. Vatrozid služi kao obrambena mjera i kao sustav ranog upozorenja. Doktrina za postavljanje najsigurnijeg mogućeg vatrozida odražava koncept minimalizma. Vatrozid bi trebao biti minimalističkog dizajna i upravljati po iznimkama. Najjednostavnije i najsigurnije konfiguriran vatrozid je onaj koji sve odbija, od svugdje i prema svemu. Svaki pristup uređaju trebao bi biti iznimka, a ne pravilo.

U Linuxu moguće je zaštititi pojedinačnih računala s *Netfilter*-om kroz njegovo korisničko sučelje *iptables*. *Netfilter* omogućuje izgradnju sigurnih vatrozida bez ograničavanja mogućnosti aplikacija i usluga. On dopušta OS-u da vrši filtriranje i oblikovanje paketa na razini jezgre, a to znači da ima manje ograničenja nego programi u korisničkom prostoru. *Netfilter* radi pozivajući se na skup tablica. Ove tablice sadrže lance, koji sadrže grupe sličnih pravila. Pravila su osnovne konfiguracijske stavke *Netfilter*-a, zapisana u *iptables*, koja sadrže kriterije za podudaranje određenog prometa i izvođenje akcija sukladno tom prometu. Promet koji se obrađuje, uspoređuje se s ovim pravilima i ako trenutni paket koji se obrađuje zadovoljava kriterije odabira pravila, tada se provodi akcija određena tim pravilom. Te akcije, između ostalog, mogu biti ignoriranje paketa, prihvaćanje paketa, odbijanje paketa ili prosljeđivanje paketa drugim pravilima za precizniju obradu.

Svako pravilo iz *iptables* oslanja se na određivanje skupa mrežnih parametara kao kombinaciju izvorišne IP (engl. *Internet Protocol*) adrese, izvorišnog porta, odredišne IP adrese i odredišnog porta. Osim toga, komunikacija se izvodi na mreži temeljenoj na TCP/IP protokolnom stožaju, te

se često koriste tri protokola: ICMP (engl. *Internet Control Message Protocol*), TCP (engl. *Transmission Control Protocol*) i UDP (engl. *User Datagram Protocol*). S ovih pet parametara mogu se izgraditi korisna pravila filtriranja, [45].

Očvršćivanje OS-a može pomoći smanjiti rizik od uspješnog kibernetičkog napada, međutim da bi bilo doista učinkovito, mora se koristiti uz implementaciju dodatnih sigurnosnih poboljšanja.

7.2.2. Implementacija sigurnosnih poboljšanja

Linux sadrži strogi model korisničkih ovlasti i nudi veliki izbor ugrađenih sigurnosnih obrana jezgre za zaštitu od ranjivosti i napada (engl. *Linux Security Modules* – LSM). Za razliku od Windows OS-a gdje su svi korisnici administratori, Linux uvelike ograničava *root* pristup kroz diskrecijsku kontrolu pristupa (engl. *Discretionary Access Control* – DAC). U Linux-u superkorisnik posjeduje sve ovlasti, a obični korisnici dobivaju samo dovoljno dopuštenja za obavljanje uobičajenih zadataka. Budući da korisnici Linux-a imaju niska prava automatskog pristupa i zahtijevaju dodatne dozvole za otvaranje privitaka, pristup datotekama ili podešavanje opcija kernela, teže je širiti zlonamjerni softver na Linux sustavu. Stoga ova inherentna ograničenja služe kao ključna obrana od napada i ugrožavanja sustava, [40].

Sigurnosno poboljšani Linux (engl. *Security-Enhanced Linux* – SELinux) implementacija je obveznog mehanizma kontrole pristupa (engl. *Mandatory Access Control* – MAC) u jezgri Linuxa, koja provjerava dopuštene operacije nakon što se provjere standardne diskrecijske kontrole pristupa. SELinux može postaviti pravila o datotekama i procesima u Linux sustavu, te o njihovim radnjama, na temelju definiranih pravila. Datoteke, uključujući direktorije i uređaje, nazivaju se objektima. Procesi, kao što je korisnik koji izvodi naredbu ili aplikacija, nazivaju se subjektima. DAC kontrolira način na koji subjekti komuniciraju s objektima i način na koji subjekti međusobno djeluju. Na OS-ima koji koriste DAC, korisnici kontroliraju dozvole objekata koje posjeduju. Oslanjanje samo na DAC mehanizme fundamentalno je neadekvatno za jaku sigurnost sustava. Odluke o pristupu DAC-u temelje se samo na identitetu korisnika i vlasništvu, zanemarujući druge informacije relevantne za sigurnost kao što je uloga korisnika, funkcija i pouzdanost programa te osjetljivost i integritet podataka. Svaki korisnik obično ima potpunu diskreciju nad svojim datotekama, što otežava provođenje sigurnosne politike za cijeli sustav, [48].

Prednosti korištenja SELinux-a, [49]:

- Procesi su odvojeni jedan od drugog tako što se izvode u vlastitim domenama, a SELinux pravila definiraju kako procesi međusobno djeluju s datotekama, kao i kako procesi međusobno djeluju. Pristup je dopušten samo ako postoji pravilo koje to izričito dopušta.
- Kontrola pristupa koja odluke o pristupu SELinux-u temelje se na svim dostupnim informacijama, kao što su SELinux korisnik, uloga, tip i razina.
- Pravila SELinuxa su administrativno definirana, provode se na cijelom sustavu i ne postavljaju se prema nahođenju korisnika.
- Smanjena ranjivost na napade eskalacije ovlasti, budući da se procesi izvode u domenama, te su odvojeni jedni od drugih, i budući da pravila politike SELinuxa definiraju kako procesi pristupaju datotekama i drugim procesima, ako je proces ugrožen, napadač ima pristup samo normalnim funkcijama tog procesa, i datotekama za koje je proces konfiguriran da ima pristup. Na primjer, ako je Apache HTTP poslužitelj ugrožen, napadač ne može upotrijebiti taj proces za čitanje datoteka u korisničkim kućnim direktorijima, osim ako nije dodano ili konfigurirano specifično pravilo SELinux politike da dopusti takav pristup.
- SELinux se može koristiti za provođenje povjerljivosti i integriteta podataka, kao i za zaštitu procesa od nepouzdanih unosa.

SELinux je dizajniran da poboljša postojeća sigurnosna rješenja, a ne da ih zamijeni. Čak i kada se koristi SELinux, važno je nastaviti slijediti dobre sigurnosne prakse, kao što je održavanje softvera ažurnim, korištenje lozinki koje je teško probiti, vatrozida i drugo.

Kako bi spriječili eksploatiranje ranjivosti potrebno je ograničiti ili ukloniti programe koji omogućuju prijenos datoteka, kao što su FTP (engl. *File Transfer Protocol*), SCP (engl. *Secure Copy Protocol*) ili curl ili ih ograničiti na određene korisnike ili IP adrese. To može spriječiti prijenos zlonamjernih datoteka na ciljni uređaj. Dodatno potrebno je ukloniti ili ograničiti pristup kompajlerima, kao što je GCC te im nikada ne davati administratorska prava kao ni tumačima ili uređivačima, uključujući *vi*, *more*, *less*, *nmap*, *perl*, *ruby*, *python*, *gdb*. Najbolja je praksa ne davati administratorska prava nijednom programu koji omogućuje pokretanje ljuske. Također poželjno je ograničiti direktorije u koje se može pisati ili koji se mogu izvršiti.

8. Zaključak

U današnje vrijeme broj povezanih uređaja stalno raste i upravljanje njima postaje sve veći izazov. Tehnološki razvoj nigdje nije bio tako dinamičan i sveobuhvatan kao što je u području komunikacijske i informacijske tehnologije. Kontinuirana digitalna transformacija u telekomunikacijskoj industriji dovodi do razvoja i uvođenja novih usluga i proizvoda pri čemu sigurnosni aspekt, u pravilu, ima vrlo mali utjecaj na široko prihvaćanje novih tehnologija. U vremenu ubrzanog razvoja novih tehnologija i digitalizacije društva područje kibernetičke sigurnosti ima sve veći značaj i bilježi snažan globalni rast zbog permanentnog oslanjanja društva u cjelini, na umrežavanje i korištenje informacijskih sustava.

Sigurnost OS-a ključna je odrednica sigurnosti uređaja spojenih na mrežu, ali nipošto nije sigurna zaštita od zlonamjernih kibernetičkih napada. Učinkovita sigurnost ovisi o dubinskoj obrani, a drugi čimbenici, uključujući implementaciju najboljih sigurnosnih praksi i pametno ponašanje na mreži, igraju središnju ulogu u digitalnom sigurnosnom položaju. Povrh toga, odabir sigurnog OS-a od najveće je važnosti, budući da je OS najkritičniji dio softvera. Linux je izvrstan izbor jer ima potencijala biti vrlo siguran zbog svoje prirode otvorenog koda, modela strogih korisničkih ovlasti, raznolikosti i relativno male baze korisnika.

Međutim, Linux nije neprobojan kada je u pitanju kibernetička sigurnost. Sustav mora biti pravilno i sigurno konfiguriran. Također, ključno je shvaćati da je sigurnost povezana s kompromisima, kako između sigurnosti i upotrebljivosti, tako i između sigurnosti i jednostavnosti korištenja. Administratori bi trebali konfigurirati svoje sustave tako da budu onoliko sigurni koliko je praktično u njihovom okruženju. Što se tiče praktičnosti, za Linux je potreban dulji proces učenja, ali nudi značajne sigurnosne prednosti u odnosu na Windows ili MacOS.

U diplomskom radu je provedeno istraživanje ranjivosti Linux OS-a. Nakon prikaza arhitekture OS-a i njegovih komponenti, provedena je analiza ranjivosti jezgre Linux OS-a u kojoj je objašnjeno koje ranjivosti se najčešće iskorištavaju za provedbu kibernetičkih napada. Za provedbu simulacija kibernetičkih napada korišten je Kali Linux OS, koji se inače koristi za sigurnosna testiranja i analizu sigurnosnih značajki. Simulacijom su prikazana dva načina na koje je moguće eksploatirati ranjivosti Linux sustava te tako zadobiti kontrolu nad sustavom. Eskalacijom ovlasti postignuto je povećanje ovlasti sa standardnog korisničkog računa do *root*-a,

potpunog pristupa sustavu. Napadi eskalacije ovlasti na Linux-u često su rezultat pogrešnih konfiguracija, iskorištavanja ranjivosti te ciljanih napada.

Dodatno, prikazano je kako se mogu dohvatiti osjetljivi podaci, poput autorizacijskih, čime je prikazana ozbiljnost ovakvih napada. Na temelju analize provedenog istraživanja, predloženi su načini unaprjeđenja sigurnosti Linux OS-a koji su podijeljeni u dvije kategorije, a to su očvršćivanje OS-a i implementacija sigurnosnih poboljšanja. Primjenom ovih metoda moguće je ostvariti željenu razinu sigurnosti OS-a.

POPIS LITERATURE

[1] *Unix Vs Linux: What is Difference Between UNIX and Linux*. Preuzeto sa:

https://www.softwaretestinghelp.com/unix-vs-linux/#What_is_Linux [Pristupljeno: kolovoz 2022.]

[2] Hertzog R, O'Gorman J, Aharoni M, O'Gorman J. *Kali Linux Revealed: Mastering the Penetration Testing Distribution*. New York: OffSec Press; 2021.

[3] *GNU in a Nutshell*. Preuzeto sa: https://www.gnu.org/gnu/about-gnu.html [Pristupljeno: kolovoz 2022.]

[4] *Što je Linux?* Preuzeto sa: https://dir.hr/sto-je-linux/ [Pristupljeno: kolovoz 2022.]

[5] *Architecture of Linux*. Preuzeto sa: https://www.javatpoint.com/architecture-of-linux [Pristupljeno: kolovoz 2022.]

[6] Bovet DP, Cesati M. *Understanding the Linux kernel*. 3. izd. O'Reilly Media; 2006. Preuzeto sa: https://www.cs.utexas.edu/~rossbach/cs380p/papers/ulk3.pdf [Pristupljeno: kolovoz 2022.]

[7] Ward B. *How linux works: what every superuser should know*. 2. izd. San Francisco: No Starch Press, Inc; 2015.

[8] *Types of shell in Unix*. Preuzeto sa: https://dokumen.tips/documents/types-of-shells-in-linux.html [Pristupljeno: kolovoz 2022.]

[9] Rodriguez-Rivera GA, Ennen J. *Introduction to Systems Programming: a Hands-on Approach.* Preuzeto sa: https://www.cs.purdue.edu/homes/grr/SystemsProgrammingBook/ [Pristupljeno: kolovoz 2022.]

[10] Entrup G, Herrmann F, Matter S, Entrup E, Jakob M, Eberhardt J, Casselt M. *Architecture of the Linux kernel*; 2018. Preuzeto sa: https://www.sra.uni-hannover.de/Lehre/WS17/S_AKSI/preview/document.pdf [Pristupljeno: kolovoz 2022.]

[11] *Properties of Linux*. Preuzeto sa: https://www.linuxtopia.org/online_books/introduction_to_linux/sect_01_04.html [Pristupljeno: kolovoz 2022.]

[12] *Quick Answer: How Many Users Use Linux*. Preuzeto sa: https://www.quickanswer.blog/quick-answer-how-many-users-use-linux/ [Pristupljeno: kolovoz 2022.]

[13] *Linux Distribution*. Preuzeto sa: https://www.suse.com/suse-defines/definition/linux-distribution/ [Pristupljeno: kolovoz 2022.]

[14] *Introduction to Linux*. Preuzeto sa: https://researchhubs.com/post/computing/linux-cmd/introduction-to-linux.html [Pristupljeno: kolovoz 2022.]

[15] *Non-Commercial Linux Use on the Rise*. Preuzeto sa: https://www.serverwatch.com/guides/non-commercial-linux-use-on-the-rise/ [Pristupljeno: kolovoz 2022.]

[16] Peraković D, Cvitić I. *Sigurnost i zaštita informacijsko komunikacijskog sustava*. [Skripta] Fakultet prometnih znanosti Sveučilišta u Zagrebu; 2021.

[17] Kelley K. *Vulnerability in Security: A Complete Overview*. Preuzeto sa: https://www.simplilearn.com/vulnerability-in-security-article [Pristupljeno: kolovoz 2022.]

[18] Paladin D. *Upravljanje ranjivostima u vremenima ciljanih prijetnji*. Preuzeto sa: https://www.borea.hr/images/dokumentacija/VulnerabilityMgmt_TargetedThreats_Web.pdf [Pristupljeno: kolovoz 2022.]

[19] Mann DE, Christey SM. *Towards a Common Enumeration of Vulnerabilities*. Bedford: The MITRE Corporation; 1999. Preuzeto sa: https://www.cve.org/Resources/General/Towards-a-Common-Enumeration-of-Vulnerabilities.pdf [Pristupljeno: kolovoz 2022.]

[20] Sumpter J. *CVSS Scores: A Practical Guide for Application*; 2021. Preuzeto sa: https://www.zerofox.com/blog/cvss-scores-practical-guide-application/ [Pristupljeno: kolovoz 2022.]

[21] *Common Vulnerability Scoring System (CVSS)*. Preuzeto sa: https://www.cve.org/About/RelatedEfforts#CVSS [Pristupljeno: kolovoz 2022.]

[22] *National Vulnerability Database*. Preuzeto sa: https://nvd.nist.gov/ [Pristupljeno: kolovoz 2022.]

[23] *About CWE*. Preuzeto sa: https://cwe.mitre.org/about/index.html [Pristupljeno: kolovoz 2022.]

[24] *Scoring CWEs*. Preuzeto sa: https://cwe.mitre.org/cwss/cwss_v1.0.1.html [Pristupljeno: kolovoz 2022.]

[25] Sarkar S. *CWE vs CVE*. Preuzeto sa: https://medium.com/@tosukriti5/cwe-vs-cve-1f0cb1cfcd19 [Pristupljeno: kolovoz 2022.]

[26] Niu S, Mo J, Zhang Z, Lv Z. *Overview of Linux Vulnerabilities*. International Conference on on Soft Computing in Information Communication Technology; 2014. Preuzeto sa: https://www.researchgate.net/publication/301387934_Overview_of_Linux_Vulnerabilities [Pristupljeno: kolovoz 2022.]

[27] *Linux: Vulnerability Statistics*. Preuzeto sa: https://www.cvedetails.com/vendor/33/Linux.html [Pristupljeno: kolovoz 2022.]

[28] Shameli- Sendi A. *Understanding Linux kernel vulnerabilities*. Preuzeto sa: https://www.researchgate.net/publication/350624677_Understanding_Linux_kernel_vulnerabilities [Pristupljeno: kolovoz 2022.]

[29] Canepa G. *How to Secure Network Services Using TCP Wrappers in Linux*; 2016. Preuzeto sa: https://www.tecmint.com/secure-linux-tcp-wrappers-hosts-allow-deny-restrict-access/ [Pristupljeno: kolovoz 2022.]

[30] *CVSS Score Distribution For Top 50 Vendors By Total Number Of "Distinct" Vulnerabilities*. Preuzeto sa: https://www.cvedetails.com/top-50-vendor-cvssscore-distribution.php [Pristupljeno: kolovoz 2022.]

[31] *Common Vulnerability Scoring System version 3.1: Specification Document*. Preuzeto sa: https://www.first.org/cvss/specification-document [Pristupljeno: kolovoz 2022.]

[32] *Metasploitable 2 Exploitability Guide*. Preuzeto sa: https://docs.rapid7.com/metasploit/metasploitable-2-exploitability-guide/ [Pristupljeno: kolovoz 2022.]

[33] *Greenbone OpenVAS*. Preuzeto sa: https://www.openvas.org/ [Pristupljeno: kolovoz 2022.]

[34] I*ntroduction to Metasploit*. Preuzeto sa: https://www.offensive-security.com/metasploit-unleashed/introduction/ [Pristupljeno: kolovoz 2022.]

[35] *Distccd- Linux man page*. Preuzeto sa: https://linux.die.net/man/1/distccd [Pristupljeno: kolovoz 2022.]

[36] Đuras K, Dugandžić N, *Metasploit framework i izrada modula za MSF*. Preuzeto sa: https://security.foi.hr/wiki/index.php/Metasploit_framework_i_izrada_modula_za_MSF.html#Metasploit_Meterpreter [Pristupljeno: kolovoz 2022.]

[37] Hernandez J. *What is Apache? In-Depth Overview of Apache Web Server*. Preuzeto sa: https://www.sumologic.com/blog/apache-web-server-introduction/ [Pristupljeno: kolovoz 2022.]

[38] Oh EM. *VulnerabilityDetails*. Preuzeto sa: https://github.com/dirtycow/dirtycow.github.io/wiki/VulnerabilityDetails [Pristupljeno: kolovoz 2022.]

[39] Moon S. *Cracking linux password with john the ripper- tutorial*. Preuzeto sa: https://www.binarytides.com/cracking-linux-password-with-john-the-ripper-tutorial/ [Pristupljeno: kolovoz 2022.]

[40] Day B. *How Secure Is Linux?* 2021. Preuzeto sa: https://linuxsecurity.com/features/how-secure-is-linux [Pristupljeno: kolovoz 2022.]

[41] *Understanding Privilege Escalation and 5 Common Attack Techniques*. Preuzeto sa: https://www.cynet.com/network-attacks/privilege-escalation/ [Pristupljeno: kolovoz 2022.]

[42] *Kibernetička sigurnost*. Preuzeto sa: https://rdd.gov.hr/kiberneticka-sigurnost-1436/1436 [Pristupljeno: kolovoz 2022.]

[43] Wita R, Teng-Amnuay Y. *Vulnerability profile for Linux*. 19th International Conference on Advanced Information Networking and Applications; 2005. Preuzeto sa: https://ieeexplore.ieee.org/document/1423610 [Pristupljeno: kolovoz 2022.]

[44] *udev*. Preuzeto sa: https://wiki.archlinux.org/title/udev [Pristupljeno: kolovoz 2022.]

[45] Turnbull J. *Hardening Linux*. 2005. Preuzeto sa: https://docs.alexomar.com/biblioteca/Linux%20Hardening.pdf [Pristupljeno: kolovoz 2022.]

[46] Frisch A. *Essential System Administration*, 3. izd. O'Reilly Media; 2002. Preuzeto sa: https://doc.lagout.org/operating%20system%20/linux/Essential%20System%20Administration.pdf [Pristupljeno: kolovoz 2022.]

[47] Nepal A. *Linux Server & Hardening Security*. Preuzeto sa: https://www.researchgate.net/publication/265162827_Linux_Server_Hardening_Security [Pristupljeno: kolovoz 2022.]

[48] Smalley S, Vance C, Salamon W. *Implementing SELinux as a Linux Security Module*. NSA; 2001. Preuzeto sa: https://www.nsa.gov/portals/75/documents/resources/everyone/digital-media-center/publications/research-papers/implementing-selinux-as-linux-security-module-report.pdf [Pristupljeno: kolovoz 2022.]

[49] Jahoda M, Krátký R, Ančincová B. *Security-Enhanced Linux. User Guide: Chapter 2. Introduction*. Preuzeto sa: https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/6/html/security-enhanced_linux/chap-security-enhanced_linux-introduction#ftn.idm139898887491376 [Pristupljeno: kolovoz 2022.]

[50] *Privilege Escalation Attack Vectors*. Preuzeto sa: https://www.cynet.com/network-attacks/privilege-escalation/#heading-4 [Pristupljeno: kolovoz 2022.]

POPIS KRATICA

OS operating system

DoS Denial of Service

GPL General Public License

CPU central processing unit

IPC inter-process communication

RAM random access memory

VFS virtual file system

MMU memory management unit

MB megabyte

FDDI Fiber Distributed Data Interface

HIPPI High Performance Parallel Interface

KVM Kernel-based Virtual Machine

ISO The International Organization for Standardization

CVE Common Vulnerability Enumeration

IDS intrusion detection system

OWASP Open Web Application Security

CVSS Common Vulnerability Scoring System

NVD National Vulnerability Database

NIST National Institute of Standards and Technology

CWE Common Weakness Enumeration

CWSS Common Weakness Scoring System

USB Universal Serial Bus

TCP Transmission Control Protocol

ARM Advanced RISC Machines

SSH Secure Shell

NVT Network Vulnerability Tests

PDF Portable Document Format

XML The Extensible Markup Language

HTML HyperText Markup Language

XSS Cross-Site Scripting

GCC GNU Compiler Collection

PID process identifier

BIOS Basic Input/ Output System

PAM Pluggable Authentication Modules

IP Internet Protocol

ICMP Internet Control Message Protocol

UDP User Datagram Protocol

LSM Linux Security Modules

DAC Discretionary Access Control

SELinux Security-Enhanced Linux

MAC Mandatory Access Control

FTP File Transfer Protocol

SCP Secure Copy Protocol

POPIS SLIKA

POPIS TABLICA

Sveučilište u Zagrebu
Fakultet prometnih znanosti
Vukelićeva 4, 10000 Zagreb

IZJAVA O AKADEMSKOJ ČESTITOSTI I SUGLASNOSTI

Izjavljujem i svojim potpisom potvrđujem da je diplomski rad isključivo rezultat mojega vlastitog rada koji se temelji na mojim istraživanjima i oslanja se na objavljenu literaturu, a što pokazuju upotrijebljene bilješke i bibliografija. Izjavljujem da nijedan dio rada nije napisan na nedopušten način, odnosno da je prepisan iz necitiranog rada te da nijedan dio rada ne krši bilo čija autorska prava. Izjavljujem, također, da 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 završnog/diplomskog rada pod naslovom Analiza ranjivosti jezgre Linux operativnog sustava kroz simulaciju kibernetičkih napada, u Nacionalni repozitorij završnih i diplomskih radova ZIR.

Student/ica:

U Zagrebu, 08.09.2022

(ime i prezime, *potpis*)