**TEHNIČKO VELEUČILIŠTE U ZAGREBU**

**STRUČNI DIPLOMSKI STUDIJ**

**Specijalizacija informatika**

Domagoj Buljan

**Implementacija arhitekture nultog povjerenja**

DIPLOMSKI RAD

Zagreb, lipnja 2025.

**Sadržaj**

[1. Uvod 1](#_Toc201602089)

[2. Tradicionalni sigurnosni modeli i njihovi nedostaci 3](#_Toc201602090)

[2.2 Tradicionalni sigurnosni modeli 3](#_Toc201602091)

[2.3 Nedostaci tradicionalnih modela 4](#_Toc201602092)

[2.4 Primjeri neuspjeha tradicionalnih modela 5](#_Toc201602093)

[2.5 Povezivanje s arhitekturom nultog povjerenja 7](#_Toc201602094)

[3. Arhitektura nultog povjerenja 8](#_Toc201602095)

[3.1 Načela arhitekture nultog povjernja 8](#_Toc201602096)

[3.2 Komponente arhitekture nultog povjerenja 10](#_Toc201602097)

[3.3 Implementacija arhitekture nultog povjerenja 11](#_Toc201602098)

[3.4 Prednosti arhitekture nultog povjerenja 14](#_Toc201602099)

[4.Tehnologije za implementaciju arhitekture nultog povjerenja 17](#_Toc201602100)

[4.1 Alati za autentifikaciju i autorizaciju 17](#_Toc201602101)

[4.2 Tehnologije za šifriranje i zaštitu podataka 18](#_Toc201602102)

[4.3 Praćenje i detekcija prijetnji 20](#_Toc201602103)

[4.4 Segmentacija mreže 22](#_Toc201602104)

[4.5 Potrebni resursi za implementaciju arhitekture nultog povjerenja 23](#_Toc201602105)

[5. Praktični dio: Implementacija i testiranje 27](#_Toc201602106)

[5.1 Implementacija modela u simuliranom okruženju 27](#_Toc201602107)

[5.2 Analiza i testiranje 40](#_Toc201602108)

[6. Zaključak 48](#_Toc201602109)

[7. Literatura 50](#_Toc201602110)

**Popis oznaka i kratica**

|  |  |
| --- | --- |
| ACL | Access Control List |
| AD | Active Directory |
| ADFS | Active Directory Federation Services |
| AES | Advanced Encryption Standard |
| AI | Artificial Intelligence |
| API | Application Programming Interface |
| APP | Application |
| BYOD | Bring Your Own Device |
| CAPTCHA | Completely Automated Public Turing test to tell Computers and Humans Apart |
| CASB | Cloud Access Security Broker |
| CISA | Cybersecurity and Infrastructure Security Agency |
| CPU | Central Processing Unit |
| DB | Database |
| DNS | Domain Name System |
| ELK | Elasticsearch, Logstash, Kibana |
| EMM | Enterprise Mobility Management |
| GPU | Graphics Processing Unit |
| HSM | Hardware Security Module |
| HTTP | HyperText Transfer Protocol |
| HTTPS | HTTP Secure |
| IAM | Identity and Access Management |
| IDS | Intrusion Detection System |
| IPS | Intrusion Prevention System |
| IT | Information Technology |
| KMS | Key Management System |
| LDAP | Lightweight Directory Access Protocol |
| MDM | Mobile Device Management |
| MFA | Multi Factor Authentication |
| ML | Machine Learning |
| NAT | Network Address Translation |
| NIC | Network Interface Card |
| NIST | National Institute of Standards and Technology |
| NTP | Network Time Protocol |
| OIDC | OpenID Connect |
| OS | Operating System |
| PA | Policy Administrator |
| PAM | Privileged Access Management |
| PE | Policy Engine |
| PEP | Policy Enforcement Point |
| PKI | Public Key Infrastructure |
| QR | Quick Response (Code) |
| RSA | Rivest–Shamir–Adleman |
| SAML | Security Assertion Markup Language |
| SDN | Software Defined Networking |
| SIEM | Security Information and Event Management |
| SOAR | Security Orchestration, Automation and Response |
| SQL | Structured Query Language |
| SSL | Secure Sockets Layer |
| SSO | Single Sign On |
| TCP | Transmission Control Protocol |
| TLS | Transport Layer Security |
| TOTP | Time-Based One Time Password |
| UBA | User Behavior Analytics |
| UDP | User Datagram Protocol |
| UEBA | User and Entity Behavior Analytics |
| URI | Uniform Resource Identifier |
| URL | Uniform Resource Locator |
| USB | Universal Serial Bus |
| VLAN | Virtual LAN |
| VPN | Virtual Private Network |
| WAN | Wide Area Network |
| ZTA | Zero Trust Architecture |
| ZTNA | Zero Trust Network Access |

**Popis slika**

[Slika 1: Proboj Capital One (preuzeto iz [5], slika 1) 6](#_Toc201600785)

[Slika 2: Shema arhitekture Beyond Corp (preuzeto iz [9], slika 1) 13](#_Toc201600786)

[Slika 3: Arhitekutra implementirane arhitekture nultog povjerenja 27](#_Toc201600787)

[Slika 4: Konfiguracija pfSense vatrozida s definiranim mrežnim sučeljima 28](file:///C:\Users\GIGABYTE\Desktop\DIPL-------\DomagojBuljanDiplomskiRad%20(1).docx#_Toc201600788)

[Slika 5: Skup pravila vatrozida 31](#_Toc201600789)

[Slika 6: Konfiguracija Keycloaka za Flask aplikaciju 34](#_Toc201600790)

[Slika 7: Prikaz toka agregiranih zapisa 37](#_Toc201600791)

[Slika 8: Grafana nadzorna ploča 39](file:///C:\Users\GIGABYTE\Desktop\DIPL-------\DomagojBuljanDiplomskiRad%20(1).docx#_Toc201600792)

[Slika 9: Prikaz zapisa u Kibani 42](#_Toc201600793)

[Slika 10: Prikaz opterećenja CPU-a poslužitelja aplikacije tijekom simuliranog napada 44](#_Toc201600794)

[Slika 11: primjer poruke o pogrešci povezane s nevažećim redirect URI parametrom 45](#_Toc201600795)

# Uvod

U suvremenom digitalnom okruženju, informacijski sustavi sve su izloženiji složenim i učestalim sigurnosnim prijetnjama koje dolaze kako izvan tako i unutar organizacijskih perimetara. Tradicionalni sigurnosni modeli, temeljeni na pretpostavci da sve što se nalazi unutar mrežnog perimetra može biti smatrano pouzdanim, pokazali su se nedostatnima u kontekstu sve kompleksnijih i dinamičnijih IT infrastruktura. S obzirom na rastući broj udaljenih korisnika, sve širu primjenu usluga u oblaku, povećan broj povezanih uređaja te prisutnost hibridnih i distribuiranih sustava, jasno je da granice mreže više nisu čvrsto definirane, a samim time ni pouzdanje zasnovano na mrežnoj lokaciji nije više primjeren temelj sigurnosti.

Upravo iz potrebe za redefiniranjem pristupa kibernetičkoj sigurnosti proizašla je koncepcija arhitekture nultog povjerenja. Riječ je o sigurnosnom modelu koji odbacuje ideju implicitnog povjerenja prema korisnicima i uređajima na temelju njihove pozicije u mreži te inzistira na načelu stalne provjere svakog pristupa bez obzira na to dolazi li izvan ili unutar organizacije. Arhitekture nultog povjerenja polazi od pretpostavke da nijedan entitet, bio on korisnik, uređaj ili aplikacija, ne smije dobiti pristup resursima dok se ne potvrdi njegov identitet i sigurnosni status, a i tada mu se dodjeljuje samo onaj minimum privilegija koji je potreban za izvršavanje točno određene funkcije. Takav pristup podrazumijeva visoku razinu segmentacije mreže, granuliranu kontrolu pristupa, stalno praćenje aktivnosti te korištenje centraliziranih sustava za upravljanje identitetom i pristupom.

Temeljem tih načela, u sklopu ovog diplomskog rada provedena je praktična implementacija arhitekture nultog povjerenja u kontroliranom, virtualiziranom laboratorijskom okruženju. Okruženje je izgrađeno na temelju više virtualnih strojeva međusobno povezanih pomoću pfSense rješenja koje je korišteno kao vatrozid i alat za segmentaciju mreže. Sustav je arhitektonski podijeljen na nekoliko logičkih i funkcionalnih mreža, mrežu za klijente, mrežu za sustavno upravljanje, mrežu za aplikacijske servise te mrežu za nadzor i prikupljanje podataka. Takav postupak omogućuje primjenu mikrosegmentacije, gdje se pojedine komponente sustava međusobno izoliraju na mrežnoj razini, a komunikacija među njima strogo se kontrolira i nadzire.

Za potrebe implementacije nadzornih i upravljačkih mehanizama korišten je niz alata otvorenog koda. Grafana je korištena za vizualizaciju sustavnih metrika, dok je za prikupljanje i analizu zapisa implementirana ELK platforma[[1]](#footnote-1), odnosno kombinacija Filebeata, Elasticsearcha i Kibane. Upravljanje korisnicima i autentifikacija provedeni su korištenjem Keycloak sustava, koji omogućuje centralizirano upravljanje identitetima, a u pozadini sustava je LDAP [[2]](#footnote-2)(*Lightweight Directory Access Protocol*) servis kao spremište korisničkih podataka. Poslužiteljski sustavi temelje se na operacijskom sustavu Ubuntu Server, dok je klijentska interakcija realizirana putem Ubuntu Desktop instanci.

Cilj ovog rada bio je demonstrirati kako se koncepti arhitekture nultog povjerenja mogu provesti u stvarno, iako testno, okruženje, koje simulira organizacijsku infrastrukturu s višestrukim mrežnim zonama i kontrolom pristupa temeljenom na identitetu. Kroz detaljnu implementaciju i konfiguraciju prikazan je postupak stvaranja sigurnosnog okvira koji se temelji na nepovjerenju, ali i na povjerenju koje se dinamički stječe kroz jasno definirane sigurnosne protokole i mehanizme autentifikacije. Osim toga, analiza dobivenih rezultata pruža uvid u prednosti, ali i izazove koji prate primjenu ovakvog modela u praksi. Iako implementacija pristupa zahtijeva značajnu promjenu paradigme u odnosu na tradicionalne sigurnosne arhitekture, ona donosi veću fleksibilnost i otpornost na prijetnje u okruženjima u kojima su prijenosi podataka, korisnici i resursi disperzirani i dinamični. Upravo zato ovaj pristup sve više dobiva na značaju, osobito u kontekstu digitalne transformacije i sve većih zahtjeva za robusnom i prilagodljivom sigurnosnom infrastrukturom. Ovaj rad doprinosi razumijevanju i praktičnom približavanju tih koncepata kroz izgradnju cjelovitog, funkcionalnog i sigurnosno osviještenog sustava.

# Tradicionalni sigurnosni modeli i njihovi nedostaci

Tradicionalni sigurnosni modeli dugo vremena su bili glavni oslonac u informacijskim sustavima. Najčešće korišteni pristupi perimetralne sigurnosti, zaštite temeljene na identitetu i mrežne segmentacije, razvijeni su u vremenu kada je jasno postojala granica između unutarnje mreže (koja je smatrana sigurnom) i vanjskog svijeta (prijetnje). Međutim, razvoj novih tehnologija, rast broja uređaja i migracija u oblak doveli su do povećanja broja ranjivosti ovih tradicionalnih pristupa, zbog čega su oni danas nedostatni [1].

## 2.2 Tradicionalni sigurnosni modeli

### 2.2.1 Perimetralna sigurnost

Model perimetralne sigurnosti temelji se na analogiji srednjovjekovnog dvorca okruženog opkopom („castle-and-moat“). Cilj ovog pristupa bio je kreirati jaku barijeru (vatrozid, virtualna privatna mreža - VPN, sustav za sprječavanje upada - IPS, sustav za detekciju upada - IDS) između organizacije i vanjskog okruženja [2]. Ovaj je model funkcionirao dobro dok su mreže bile jednostavne, a vanjske prijetnje jasno definirane.  
Slučaj korištenja ovog pristupa bio bi da tvrtka koristi vatrozid kako bi filtrirala promet izvana prema unutra, dopuštajući pristup isključivo određenim servisima (poput web poslužitelja). No, ako jedan zaposlenik nenamjerno klikne na phishing poveznicu i napadač dobije pristup njegovom računalu, vatrozid više neće pružiti zaštitu jer se napadač već nalazi unutar perimetra.

### 2.2.2 Zaštita temeljena na identitetu i uređaju

Ovaj model provjerava korisnikov identitet (korisničko ime, lozinka, certifikati) i uređaj koji pristupa sustavu. Jednom kad su autentifikacija i autorizacija provedene, korisniku je dodijeljen određeni nivo povjerenja [3]. Ukoliko tvrtka koristi sustav autentifikacije na bazi korisničkog imena i lozinke. Korisnik nakon prijave ima pristup svim resursima koji su mu dozvoljeni prema njegovoj ulozi. U slučaju da napadač ukrade korisnikovu lozinku putem phishinga ili neki drugi ilegalan način, autentifikacija postaje beskorisna jer napadač sada ima potpuno istu razinu pristupa kao legitimni korisnik.

### 2.2.3 Mrežna segmentacija

Segmentacija mreže podrazumijeva podjelu mreže na manje dijelove odnosno segmente, čime se smanjuje rizik kretanja napadača kroz mrežu. Cilj ovog pristupa je ograničiti pristup određenim resursima ili dijelovima mreže.  
U velikim tvrtkama, ako je sve pravilno implementirano, mreža je segmentirana prema odjelima kao što su računovodstvo, prodaja, informatika itd. Ukoliko je računalo u odjelu prodaje kompromitirano, napadač u teoriji ne može pristupiti sustavima računovodstva jer je taj segment zasebno osiguran. Međutim, u praksi napadači često koriste ranjivosti poput neispravnih konfiguracija segmentacije ili slabo osigurane pristupne točke između segmenata.

## 2.3 Nedostaci tradicionalnih modela

### 2.3.1 Implicitno povjerenje

Tradicionalni modeli pretpostavljaju da su svi korisnici i uređaji unutar mreže pouzdani čim jednom prođu inicijalnu autentifikaciju ili pristupe mreži [1]. To predstavlja kritičnu sigurnosnu manu. U slučaju napada na tvrtku Target 2013., napadači su kompromitirali jednog dobavljača i time dobili pristup mreži. Budući da je unutar mreže postojao visok stupanj implicitnog povjerenja, napadači su se mogli slobodno kretati kroz sustave, što je dovelo do krađe milijuna osobnih podataka [4].

### 2.3.2 Nedostatak kontinuirane verifikacije

Tradicionalni sigurnosni pristupi temelje se na jednokratnoj autentifikaciji, bez kontinuirane verifikacije identiteta i namjere korisnika tijekom sesije [3].  
Ukoliko korisnik nakon inicijalne autentifikacije promijeni ponašanje (npr. pokušava pristupiti neuobičajenim resursima ili podacima u neobično vrijeme), tradicionalni sustavi to uglavnom neće detektirati, omogućavajući napadačima nesmetan pristup nakon što kompromitiraju autentifikaciju.

### 2.3.3 Ograničena vidljivost i kontrola

Tradicionalni modeli rijetko imaju sposobnost detaljnog praćenja aktivnosti na mreži, osobito ako je napadač već unutar mreže [2]. Tvrtka koja nema sustav za nadzor mrežnih aktivnosti neće primijetiti neuobičajen promet ili ponašanje korisnika. Često prođe dugo vremena (mjeseci) prije nego što organizacija shvati da je kompromitirana, omogućujući napadačima neometanu eksfiltraciju podataka.

## 2.4 Primjeri neuspjeha tradicionalnih modela

### 2.4.1 Napad WannaCry (2017.)

Napad WannaCry jedan je od najpoznatijih globalnih napada ucjenjivačkim softverom (*ransomware*), koji je pogodio više od 230.000 računala u preko 150 zemalja diljem svijeta tijekom svibnja 2017. godine. Napad se koristio iskorištavanjem sigurnosne ranjivosti poznate kao EternalBlue, koja je omogućavala brzo širenje kroz mreže koje su koristile tradicionalne sigurnosne modele (perimetralnu zaštitu). Mnoge tvrtke i institucije nisu primijenile učinkovitu segmentaciju mreže, što je olakšalo lateralno širenje napada unutar mreža. WannaCry je blokirao pristup podacima žrtve kriptirajući ih, a zatim tražio plaćanje otkupnine u bitcoinima kako bi korisnicima ponovno omogućio pristup njihovim datotekama. Napad je razotkrio nedostatke tradicionalnih pristupa sigurnosti, posebno važnost pravilne mrežne segmentacije i upravljanja ranjivostima [3].

### 2.4.2 Socijalni inženjering

*Phishing* [[3]](#footnote-3) i drugi oblici socijalnog inženjeringa ostaju među najčešćim i najučinkovitijim tehnikama koje koriste kibernetički kriminalci. Napadi često ciljaju na ljudski faktor, iskorištavajući povjerenje ili nedovoljnu svijest zaposlenika o kibernetičkim prijetnjama. Primjeri uključuju hakiranje elektorničke pošte velikih korporacija i financijskih institucija, gdje se napadači lažno predstavljaju kao legitimne osobe ili službe te na taj način navode zaposlenike da otkriju povjerljive podatke ili preuzmu zlonamjerne privitke. Takvi slučajevi jasno pokazuju da same tehničke sigurnosne mjere nisu dovoljne, već je potrebna kontinuirana edukacija i osvještavanje zaposlenika kako bi se smanjila vjerojatnost uspješnosti napada putem socijalnog inženjeringa.

### 2.4.3 Proboj Capital One

Proboj Capital One iz 2019. godine ističe probleme tradicionalnih sigurnosnih modela primijenjenih na infrastrukture u oblaku (Slika 1). Napadač je iskoristio pogrešnu konfiguraciju i propuste u zaštiti sustava u oblaku te pristupio osjetljivim podacima preko 100 milijuna korisnika Capital One banke. Napad je bio moguć upravo zbog implicitnog povjerenja i ograničene vidljivosti koje tradicionalni sigurnosni modeli omogućavaju. Naime, tradicionalni pristupi često ne pružaju dovoljno granularnu kontrolu pristupa ili pregled nad aktivnostima u oblaku, čime se povećava ranjivost sustava na sofisticirane napade. Proboj Capital One ukazao je na potrebu implementacije sigurnosnih modela koji eksplicitno provjeravaju svaki zahtjev za pristupom te kontinuirano prate aktivnosti unutar okruženja u oblaku, čime se umanjuje mogućnost sličnih propusta [5].

Slika na kojoj se prikazuje tekst, snimka zaslona, dijagram, crta

Sadržaj generiran uz AI možda nije točan.

Slika 1: Proboj Capital One (preuzeto iz [5], slika 1)

## 2.5 Povezivanje s arhitekturom nultog povjerenja

Zbog jasno izraženih nedostataka tradicionalnih sigurnosnih modela koji se oslanjaju na perimetarsku zaštitu, implicitno povjerenje unutar mreže i nedovoljna vidljivost aktivnosti korisnika i sustava, razvijen je koncept arhitekture nultog povjerenja. Ključno načelo ovog pristupa sažeto je u izrazu „nikad ne vjeruj, uvijek provjeravaj“. Za razliku od klasičnih modela koji automatski vjeruju korisnicima i uređajima unutar mreže, ovaj model polazi od pretpostavke da se nijedan zahtjev za pristupom ne smije smatrati pouzdanim bez prethodne stroge provjere, bez obzira na to odakle zahtjev dolazi, bio on unutar ili van mreže.

Arhitektura nultog povjerenja eliminira implicitno povjerenje te inzistira na kontinuiranoj provjeri identiteta korisnika, uređaja, i konteksta pristupa. To znači da se svaki pokušaj pristupa resursima mora provjeriti u stvarnom vremenu, uključujući faktore prava pristupa, ponašanje korisnika, i dr.

Jedan od temelja ovog pristupa je mikrosegmentacija mreže, kojom se mreža dijeli na manje, izolirane dijelove kako bi se spriječilo kretanje napadača unutar sustava. Time se značajno ograničava prostor napada čak i ako dođe do kompromitacije jednog segmenta. Arhitektura nultog povjerenja također osigurava detaljno praćenje i bilježenje svih aktivnosti, omogućujući sigurnosnim timovima da brže prepoznaju sumnjivo ponašanje i brzo reagiraju na potencijalne prijetnje. Na taj način, arhitektura nultog povjerenja ne pruža samo preventivne mehanizme, već i detekcijske i reaktivne sposobnosti, stvarajući dinamičan, kontekstualno svjestan sigurnosni okvir koji bolje odgovara suvremenim prijetnjama, posebno u okruženjima s distribuiranim resursima [6].

# 3. Arhitektura nultog povjerenja

Tradicionalni pristup kibernetičkoj sigurnosti polazio je od pretpostavke da se resursi unutar granica mreže mogu smatrati pouzdanima, uz snažnu zaštitu mreže. No, pojavom rada na daljinu, BYOD [[4]](#footnote-4)(*bring your own device*) police i računalstva u oblaku, takav model pokazuje ozbiljne nedostatke, jednom kada napadač probije perimetar, dobiva preširok pristup unutarnjim resursima. Ovaj izazov potaknuo je razvoj koncepta arhitekture nultog povjerenja koja eliminira implicitno povjerenje u mreži. Termin *Zero Trust* uveo je analitičar John Kindervag 2010. godine kao novi model zaštite informacija [7]. Ključna premisa arhitekture nultog povjerenja je da se nikad ne vjeruje, uvijek provjerava svakog korisnika i uređaj, bez obzira nalazi li se unutar ili izvan mreže [8]. Za svaki pristup resursima zahtijevaju se stroga provjera identiteta i konteksta, jer se unaprijed pretpostavlja narušavanje sigurnosti sistema. Ovakav paradigmatski zaokret, pomak od statičnih mrežnih perimetara pomičući fokus prema korisnicima, uređajima i podacima, definiran je i standardiziran kroz službene smjernice organizacija poput NIST-a[[5]](#footnote-5) [6]. Drugim riječima, model nultog povjerenja ne pruža nikakvo implicitno povjerenje korisnicima ili uređajima temeljem njihove mrežne lokacije, već svaki zahtjev za pristupom smatra potencijalno zloćudnim dok se ne dokaže suprotno [6].

## 3.1 Načela arhitekture nultog povjernja

Arhitektura nultog povjerenja se bazira na nekoliko temeljnih načela i pravila ponašanja sustava. Ona se mogu sažeti u „nikad ne vjeruj, uvijek provjeravaj“, što znači da niti jedan entitet ne dobiva povjerenje dok se ne autentificira i autorizira [8]. Iz takvog stava proizlazi i pretpostavka da se sustav može u svakom trenutku može biti ugrožen, zbog čega je potrebno kontinuirano pratiti i provjeravati sve aktivnosti. Glavna načela pristupa nultog povjerenja obrađena su idućim paragrafima.

Provjera identiteta i uređaja, svaki korisnik i uređaj moraju proći strogu autentifikaciju (višefaktorska autentifikacija - MFA) i autorizaciju prije pristupa bilo kojem resursu. Identitet se pritom provjerava eksplicitno na temelju više dostupnih podataka (uloga korisnika, lokacija, vrijeme pristupa) umjesto da se pristup odobrava samo zato što je zahtjev došao iz interne mreže.

Najmanji potrebni pristup, korisnicima i sustavima dodjeljuju se isključivo minimalne ovlasti potrebne za obavljanje posla, principom najmanjih privilegija. Svaka sesija dobiva ograničena prava pristupa jasno definirana politikama, čime se smanjuje potencijalno ugrožavanje sustava koje zlonamjeran korisnik ili program može izazvati ako dobije pristup.

Arhitektura nultog povjerenja pretpostavlja kontinuirano nadgledanje aktivnosti korisnika i uređaja u stvarnom vremenu zbog mogućih sumnjivih ponašanja i brzog reagiranja na potencijalne prijetnje. Autentifikacija i autorizacija nisu jednokratan događaj – sustav stalno provjerava je li korisnik i njegov uređaj i dalje ispunjavaju uvjete sigurnosnih politika tijekom cijele sesije.

Mreža i resursi podijeljeni su na manja područja povjerenja kako bi se ograničio opseg kretanja potecijalnog napadača. Pristup podacima i uslugama određuje se na osnovi konteksta, uloga i lokacije, čime se sprječava nekontrolirano lateralno kretanje unutar mreže čak i ako je neki dio ugrožen.

Šifriranje i integritet podataka, podaci u prijenosu kroz mrežu, kao i u „mirovanju“, trebaju biti kriptografski zaštićeni. Snažno šifriranje komunikacije smanjuje mogućnost presretanja ili ugroze podataka čak i u slučaju da mrežni promet prolazi kroz nepouzdane djelove.

Ove principe zajednički provodi arhitektura nultog povjerenja, skup pravila koji definiraju tko, kada i pod kojim uvjetima smije pristupiti određenom resursu. Važno je naglasiti da ovakva arhitektura nije jedinstven proizvod, već strategija i skup politika u kojem se ovi principi implementiraju kroz kombinaciju tehnologija (npr. sustavi za upravljanje identitetima, mrežni kontroleri, sigurnosni agenti na uređajima) i procesa unutar organizacije[10].

## 3.2 Komponente arhitekture nultog povjerenja

Za ostvarenje navedenih načela potrebna je odgovarajuća arhitektura. NIST definira nekoliko logičkih komponenti koje čine jezgru arhitekture nultog povjerenja [6]. Komponente su u međusobnoj interakciji i zajednički provode politike sigurnosti nad zahtjevima za pristup.

*Policy Engine* (PE) je središnji modul (logički pogon politika) koji donosi konačne odluke hoće li korisniku odnosno uređaju biti odobren ili odbijen pristup određenom resursu. PE primjenjuje pravila sigurnosne politike organizacije i procjenjuje povjerenje na temelju više izvora informacija (identitet korisnika, stanje uređaja, kontekst zahtjeva, vanjski obavještajni podaci o prijetnjama itd.) kako bi izračunao razinu povjerenja i odlučio ispunjava li zahtjev uvjete za pristup.

*Policy Administrator* (PA), komponenta koja izvršava odluke donesene od strane *Policy Enginea*. PA uspostavlja ili prekida komunikacijski kanal između subjekta (korisnika/aplikacije) i traženog resursa na temelju odluke PE-a. Ako je pristup odobren, PA izdaje potrebne vjerodajnice i konfigurira elemente kako bi se veza uspostavila. Ukolko je zahtjev odbijen (ili ranije odobren pristup treba prekinuti), PA nalaže prekid veze. U praksi, PE i PA mogu biti implementirani zajedno kao jedinstveni servis za odlučivanje i provođenje politike.

*Policy Enforcement Point* (PEP) obično su distribuirani komponentni čvorovi koji stoje između klijenta i resursa te fizički provode sigurnosne odluke. PEP je zadužen za omogućavanje ili blokiranje pojedinih konekcija na temelju uputa od PA, nadzor aktivnih sesija te eventualno njihov prekid ukoliko dođe do promjene stanja (npr. povlačenja odobrenja). PEP se može sastojati od klijentske strane (agent na uređaju korisnika ili aplikacije) i poslužiteljske strane (ispred resursa) kroz koje prolazi sav promet prema resursu, kako bi se svaka komunikacija autorizirala i nadzirala [6].

Iza ovih ključnih djelova, okruženje nultog povjerenja oslanja se i na niz pomoćnih sustava koji pružaju informacije i provode dodatne kontrole. Tu spadaju sustavi za upravljanje identitetima (npr. LDAP direktoriji za korisničke račune i autentifikaciju), zatim upravljanje uređajima i povjerenjem uređaja, alati za procjenu stanja uređaja (jesu li ažurirani, imaju li dopušten softver i sl.) koji kontinuirano dostavljaju podatke *Policy Engineu*.

U daljnoj implementaciji arhitekture nultog povjerenja integriraju se sustavi praćenja događaja i prijetnji, kao što su SIEM [[6]](#footnote-6)(*Security Information and Event Management*) alati koji prikupljaju zapise o aktivnostima, mrežnom prometu i sigurnosnim incidentima, što sve može poslužiti za podešavanje odluka PE-a u realnom vremenu. Repozitoriji politika i pravila pristupa također su dio arhitekture, oni sadrže definicije dopuštenih kombinacija uloga i resursa (što tko smije raditi) i predstavljaju osnovu na temelju koje *Policy Engine* donosi odluke. Unutar arhitekture su infrastruktura javnih ključeva (PKI) i sustavi za izdavanje certifikata za autentifikaciju, kao i razni kontroleri mreže ili oblaka koji provode segmentaciju i usmjeravanje prometa prema pravilima politike nultog povjerenja [6]. Sve ove komponente zajednički tvore arhitekturu u kojoj se kontrolni dio (PE, PA, PEP i povezani servisi) logički odvaja od podatkovnog djela (stvarnog prometa do aplikacija), omogućujući centralizirano i kontinuirano provođenje sigurnosnih politika u cijelom sustavu.

## 3.3 Implementacija arhitekture nultog povjerenja

Uvođenje arhitekture nultog povjerenja u organizaciju složen je pothvat koji zahtijeva pažljivo planiranje, postupnu integraciju tehnologija i prilagodbu procesa. Budući da arhitektura nije pojedinačni proizvod, već skup principa i zahtjeva, svaka organizacija mora izraditi strategiju implementacije koja odgovara njenim specifičnim rizicima i infrastrukturi. Tipično je potrebno fazno uvesti više međusobno usklađenih sigurnosnih rješenja i politika unutar postojeće okoline. Prvi korak je procjena trenutne sigurnosne infrastrukture i identifikacija kritičnih resursa, korisničkih skupina i potencijalnih ranjivosti. Na temelju te analize, izrađuje se plan kako zatvoriti identificirane rupe u povjerenju te kako postojeće kontrole uskladiti s principima arhitekture nultog povjerenja. U ranoj fazi ključno je osigurati potporu menadžmenta i educirati IT osoblje o novom modelu, budući da implementacija može zahtijevati značajne organizacijske promjene u upravljanju pristupom. Nakon planiranja, kreće se s provedbom niza ključnih mjera za postizanje okruženja nultog povjerenja.

Prvi korak implementacije je jačanje autentifikacije i identiteta, uvesti snažne mehanizme autentifikacije poput višefaktorske provjere identiteta (MFA) za sve korisnike, uključujući interne zaposlenike, udaljene korisnike i partnere. Paralelno je potrebno uspostaviti učinkovito upravljanje identitetima i pristupima (*Identity and Access Management* – IAM) kako bi se centraliziralo kontroliranje tko ima pristup kojim resursima, uz primjenu tehnologija jednosturke prijave i strožih pravila zaporki.

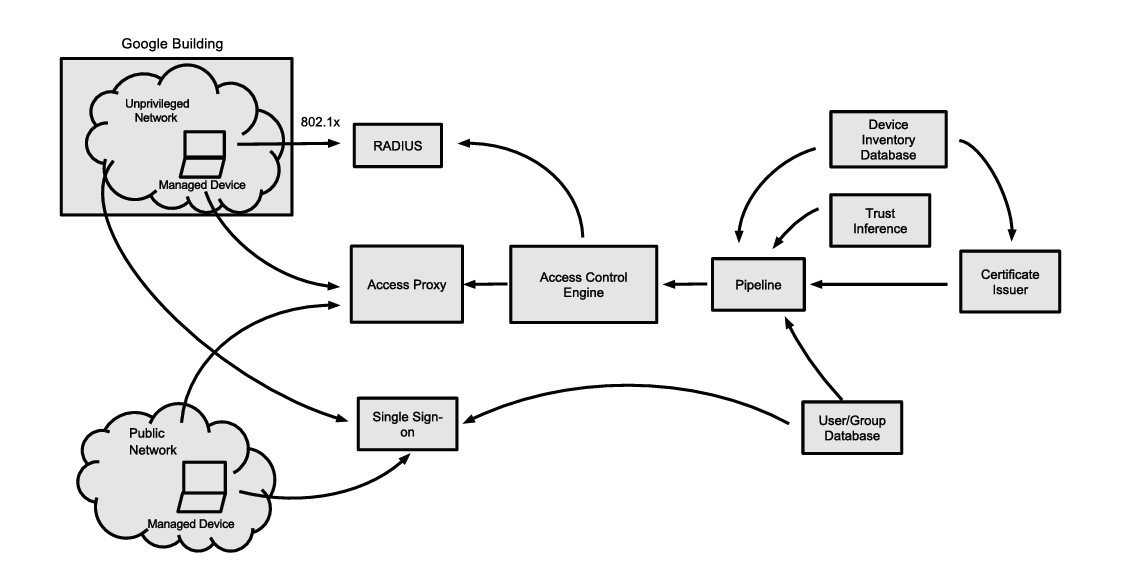
Nadalje, provjera stanja i sigurnosti uređaja, gdje treba osigurati da samo uređaji koji zadovoljavaju sigurnosne zahtjeve mogu pristupiti mreži, primjerice, uređaj mora biti ažuriran te odgovarajuću antivirusnu zaštitu . Ovo se postiže integracijom rješenja za upravljanje mobilnim uređajima i krajnjim točkama (MDM/EMM[[7]](#footnote-7)) te agenata koji izvještavaju o stanju uređaja prilikom svakog pokušaja pristupa. Nepoznati i potencijalno ugrožavajući uređaji trebaju biti izolirani ili podvrgnuti dodatnoj provjeri prije nego im se dopusti ograničen pristup.

Mikrosegmentacija mreže i resursa pregrađuje mrežu na manje segmente prema razini povjerljivosti podataka i važnosti usluga. Implementirati pravilo da svaki dijelić mrežnog prometa prolazi kroz kontrolnu točku (PEP) gdje se provjerava politika. Na primjer, pristup poslužiteljima baza podataka ograničiti samo na legitimne aplikacijske poslužitelje umjesto da cijela interna mreža ima pristup. Segmentacija može biti ostvarena VLAN-ovima, softverski definiranom mrežom (SDN) ili usmjernim kontrolama, uz striktno filtriranje komunikacije između segmenata prema principu nužnosti.

Uspostavljanje sustava za praćenje aktivnosti korisnika i sustava (npr. prikupljanje zapisa, mrežni nadzor, UEBA – analitika ponašanja korisnika i entiteta) provodi se stalnim nadzorom i detekcijom anomalija. Korištenjem SIEM platformi i alata za detekciju prijetnji u stvarnom vremenu, svako odstupanje od uobičajenog ponašanja može se brzo uočiti i pokrenuti odgovarajući odgovor (npr. privremeno blokiranje sumnjivog korisničkog računa ili uređaja dok se ne izvrši provjera). Automatizacija je ovdje važna, mnoge implementacije uključuju SOAR (*Security Orchestration, Automation, and Response*) alate koji automatski reagiraju na incidente prema definiranim pravilima, čime se ubrzava odgovor i smanjuje prostor napadaču.

Proces izrade arhitekture nultog povjerenja nije gotov njegovom implementacijom, politike pristupa trebaju se stalno prilagođavati na temelju prikupljenih podataka i iskustava te zahtijeva neprekidno usavršavanje politika pristupa. Organizacija treba redovito preispitivati dodijeljene privilegije (provoditi recertifikaciju pristupa korisnika), testirati sigurnosne mehanizme (provjere probijanjem) i unaprjeđivati pravila kako se prijetnje razvijaju. Također, treba planirati integraciju novih tehnologija kako okruženje sazrijeva.[10]

Konkretan primjer realizacije arhitekture nultog povjerenja u praksi pružila je kompanija Google sa svojom inicijativom BeyondCorp (Slika 2). Još 2014. godine Google je objavio detalje svoje internetske arhitekture koja je potpuno napustila tradicionalni VPN pristup internoj mreži.



Slika 2: Shema arhitekture Beyond Corp (preuzeto iz [9], slika 1)

Umjesto privilegirane korporativne mreže, Google je svakom zaposleniku omogućio rad s bilo koje lokacije tako da svaki pristup internim resursima ide preko sigurnosnih provjera identiteta korisnika i stanja uređaja, uz šifriranu komunikaciju do aplikacija. Time je postignuto da lokacija s koje netko radi postaje nebitna, svi zahtjevi (bilo iz Google ureda ili s kućne mreže) tretiraju se jednako, i prolaze kroz isti sustav provjera prije dobivanja pristupa [9]. Rezultat je da Googleovi zaposlenici mogu sigurno raditi bez upotrebe VPN-a, a sigurnost se oslanja na kontrolu pristupa temeljenu na kontekstu i povjerenju uređaja/korisnika umjesto na mrežnoj zoni. Ovaj primjer pokazuje da je implementacija modela nultog povjerenja ostvariva i u vrlo složenim okruženjima, ali zahtijeva višegodišnje ulaganje i promjenu načina razmišljanja o sigurnosti. Danas i mnoge druge velike organizacije uvode slične koncepte, a dostupni su i komercijalni alati pod nazivom „mrežni pristup bez povjerenja“ (ZTNA) koji pomažu u provođenju ovih principa u heterogenim mrežnim okolinama. Tijekom implementacije arhitekture nultog povjerenja organizacija treba uravnotežiti sigurnost s uporabljivošću. Previše restriktivan sustav mogao bi ometati poslovne procese, dok premalo restriktivan neće postići željene ciljeve. Stoga se model često uvodi iterativno, uz model koji predviđa faze razvoja (početna, srednja, napredna faza) u kojima se postupno povećava razina provjera i automatizacije. Primjer takvog pristupa je model zrelosti koji je izradio američki CISA (*Cybersecurity and Infrastructure Security Agency*), a koji omogućuje organizacijama da ocijene gdje se nalaze na putu prema arhitekturi nultog povjerenja, ciljanom stanju i planiraju sljedećih koraka implementacije.

## 3.4 Prednosti arhitekture nultog povjerenja

Arhitektura nultog povjerenja donosi niz konkretnih sigurnosnih i poslovnih prednosti za organizaciju. Uklanjanjem implicitnog povjerenja unutar mreže značajno se smanjuje površina napada. Čak i ako napadač kompromitira jednu točku (korisnički račun ili uređaj), striktna segmentacija i provjere na svakom koraku onemogućuju ga da se slobodno kreće lateralno kroz sustav i pristupa osjetljivim podacima. Na taj način arhitektura nultog povjerenja ograničava domet štete potencijalnog napada i otežava uspjeh napadača.

Svaki zahtjev za pristupom resursima mora zadovoljiti definirane politike i proći autentifikaciju/autorizaciju, čime se osigurava da samo ovlašteni korisnici pod odgovarajućim uvjetima mogu pristupiti povjerljivim informacijama. Time se sprječava neovlašten pristup podacima čak i od strane internih korisnika koji inače imaju mrežnu povezanost – npr. zaposlenik iz jednog odjela ne može pristupiti podacima drugog odjela ako za to nema odobrenje. Sve skupa, model osigurava *„just-in-time“* i *„just-enough“* pristup, čuvajući podatke od zlouporabe.

Kontinuirano praćenje aktivnosti u okruženju nultog povjerenja daje sigurnosnim timovima mnogo bolji uvid u to tko i što se događa na mreži u svakom trenutku. Centralizirano prikupljanje zapisa o pristupima, pokušajima neovlaštenih ulaza i anomalijama ponašanja omogućuje bržu detekciju napada u ranoj fazi. Također, kroz analitiku tih podataka organizacija može prepoznati slabe točke i poboljšati sigurnosne politike. Ukratko, arhitektura nultog povjerenja pruža kontrolu pristupa uz bogat skup telemetrijskih podataka za nadzor sigurnosti.

Striktnim kontrolama pristupa i evidencijama svakog zahtjeva, model olakšava zadovoljavanje sigurnosnih zahtjeva iz propisa kao što su GDPR [[8]](#footnote-8)i slični standardi o zaštiti podataka. Budući da implementira načela poput najmanjih privilegija i detaljnog audita aktivnosti, arhitektura nultog povjerenja pomaže organizacijama da demonstriraju visok stupanj kontrole nad pristupom osjetljivim informacijama te tako budu u skladu s regulatornim okvirom [10].

Ispravno implementirana arhitektura nultog povjerenja omogućuje sigurniji rad na daljinu i iz oblaka bez ugrožavanja zaštite. Korisnici mogu pristupati resursima s bilo koje lokacije (ured, dom, javna mreža) na ujednačen način, što je postalo ključno u doba povećanog remote rada. Primjer Google BeyondCorp pokazao je da zaposlenici mogu raditi bez VPN-a uz zadržavanje ili poboljšanje korisničkog iskustva, dok sigurnost i dalje ostaje na visokoj razini [9]. Takva fleksibilnost u konačnici doprinosi produktivnosti – sigurnosne kontrole rade u pozadini i ne ometaju korisnika više nego što je nužno, a IT odjeli mogu lakše podržati hibridne i okoline u oblaku.

Uz ove prednosti, vrijedi spomenuti da arhitektura nultog povjerenja doprinosi i kulturi sigurnosti unutar organizacija: povećava se svijest korisnika da svaki njihov pristup može biti provjeren i evidentiran, što samo po sebi djeluje preventivno na nepažljivo ponašanje. Sve navedeno rezultira značajno poboljšanom ukupnom sigurnosnoti organizacije.

# 4.Tehnologije za implementaciju arhitekture nultog povjerenja

Model nultog povjerenja predstavlja sigurnosni pristup koji uklanja implicitno povjerenje u mreži, nitko i ništa se ne smatra pouzdanim bez provjere, bilo unutar ili izvan organizacijskog perimetra [11]. Ovaj model slijedi načelo „nikad ne vjeruj, uvijek provjeravaj“, što znači da se svaki zahtjev za pristupom resursima mora autentificirati, autorizirati i kontinuirano validirati. U praksi, implementacija arhitekture oslanja se na čitav niz tehnologija i alata koji zajednički omogućuju takav strog pristup sigurnosti. U nastavku su obrađene ključne kategorije tehnologija potrebnih za ostvarenje modela: od identitetskih rješenja, preko šifriranja podataka i nadzora anomalija, do mrežne segmentacije i specifičnih alata te infrastrukturnih resursa.

## 4.1 Alati za autentifikaciju i autorizaciju

Jedan od temelja modela je snažna autentifikacija i detaljna autorizacija za svakog korisnika i uređaj prije odobravanja pristupa resursima. U tu svrhu koriste se sljedeći alati i mehanizmi.

Višefaktorska autentifikacija (MFA) koja od korisnika zahtijeva kombinaciju više faktora (lozinka, mobilni token, biometrija itd.) kako bi dokazali svoj identitet. MFA dramatično smanjuje rizik neovlaštenog pristupa jer kompromitirana lozinka sama po sebi više nije dovoljna za ulazak u sustav. U okruženju nultog povjerenja MFA se primjenjuje široko, čak i za interne resurse, kako bi se osigurala stalna provjera identiteta korisnika [12]. Prethodna istraživanja upozoravaju da nedostatna implementacija MFA ostavlja sustave izloženima napadima putem ukradenih vjerodajnica [2]. Stoga je uvođenje MFA mehanizama (uključujući i mogućnosti bezlozinkske autentifikacije) jedan od prvih koraka u izgradnji arhitekture nultog povjerenja.

Jedinstvena prijava (*Single Sign-On* - SSO) korisnicima omogućuje pristup više aplikacija i usluga s jednom kombinacijom vjerodajnica uz jednu prijavu. Iako se na prvi pogled SSO odnosi na praktičnost za korisnike, važan je sigurnosni alat. Centraliziranom autentifikacijom postiže se bolja kontrola nad pristupom te administratori na jednom mjestu upravljaju pristupnim pravilima i ovlastima za sve povezane aplikacije. U kontekstu arhitekture nultog povjerenja, SSO u kombinaciji s MFA-om osigurava da svaki session bude propisno autentificiran, dok istovremeno smanjuje opterećenje korisnika (jedna prijava umjesto mnogih) i potiče korisnike na usvajanje jačih sigurnosnih praksi (jer ne moraju pamtiti mnogo lozinki). Time se smanjuje vjerojatnost nesigurnih ponašanja poput ponovne upotrebe lozinki, a sustav i dalje provodi strogu provjeru svakog pristupa.

Upravljanjem privilegiranim pristupom (*Privileged Access Managemen* - PAM) provodi se načelo najmanjih privilegija koje nalaže da korisnici i računala imaju samo onaj minimum ovlasti koji im je potreban za rad. PAM sustavi pomažu u ostvarenju tog cilja ograničavanjem pristupa administrativnim i drugim osjetljivim računima. To uključuje mehanizme kao što su privremene (*just-in-time*) privilegije, trezor lozinki za privilegirane račune i strogi nadzor akcija koje provode administratori. Korištenjem PAM alata može se spriječiti da kompromitacija jednog visokoprivilegiranog računa ugrozi čitavu organizaciju. Istraživanja pokazuju da pretjerane korisničke ovlasti ili stalni administrativni pristup predstavljaju značajan sigurnosni rizik, budući da jedan ugroženi račun može dovesti do širokog neovlaštenog pristupa sustavu [12]. PAM uvodi kontrolne točke (npr. dodatne MFA provjere za privilegirane radnje, odobravanje pristupa po potrebi i sl.) kako bi se takvi rizici umanjili te kako bi se osiguralo da se čak i unutar mreže svaki pristup kritičnim resursima iznova autorizira.

## 4.2 Tehnologije za šifriranje i zaštitu podataka

Robusna zaštita podataka, kako onih u prijenosu kroz mrežu tako i onih pohranjenih na sustavima, ključna je za strategiju nultog povjerenja. Zbog toga što arhitektura nultog povjerenja podrazumijeva nepovjerenje u mrežu (uključujući i internu mrežu organizacije), šifriranje (enkripcija) podataka postaje neophodno na svim razinama.

Simetrični algoritmi (poput AES) koriste se za brzu i učinkovitu zaštitu većih količina podataka, često za šifriranje podataka u mirovanju (na disku ili u bazi podataka) i za šifriranje komunikacijskog kanala nakon uspostave sesije. Asimetrični algoritmi (poput RSA) primjenjuju se pri razmjeni ključeva i digitalnom potpisivanju, odnosno svugdje gdje je potrebno osigurati siguran dogovor tajnih ključeva ili provjeriti identitet pošiljatelja. U okruženju nultog povjerenja kombinacija ovih metoda osigurava da samo ovlašteni subjekti mogu dešifrirati i koristiti podatke. Uspostavom kripitirane (TLS/SSL) komunikacije, javni i privatni ključevi koriste se za razmjenu simetričnog ključa sesije, kojim se potom šifrira cjelokupan promet između klijenta i poslužitelja.

Svi podaci koji putuju preko mreže lokalne mreže, interneta ili nekim drugim načinom trebaju biti šifrirani za što je potrebna enkripcija podataka u prijenosu. Tradicionalno se unutar perimetra organizacije dio prometa smatra „pouzdanim“ i često je bio nešifriran. Arhitektura nultog povjerenja napušta takav pristup, svaki mrežni spoj smatra se potencijalno nesigurnim te se zahtijeva autentifikacija svih veza i kriptiranje cjelokupnog prometa [6]. To podrazumijeva korištenje protokola kao što su HTTPS/TLS za komunikaciju preko interneta, VPN tunela ili novijih pristupa poput tuneliranja[[9]](#footnote-9) koji enkriptiraju promet između klijenta i poslužiteljskih resursa. Šifriranje u prijenosu onemogućuje presretanje osjetljivih podataka i otežava napadačima izvođenje MitM (*Man-in-the-Middle*) napada ili prisluškivanje mreže, čak i ako se nalaze unutar iste mrežne infrastrukture.

Zaštita pohranjenih podataka jednako je važna. To znači šifriranje sadržaja baza podataka, sustava datoteka, sigurnosnih kopija, pa i samih diskova. U oblaku, gdje podaci organizacije mogu biti pohranjeni na vanjskoj infrastrukturi, enkripcija u mirovanju osigurava da pružatelj usluge ili neovlaštene osobe ne mogu jednostavno pročitati povjerljive informacije čak ni ako dođu do medija za pohranu. Implementacija uključuje uporabu jakih enkripcijskih algoritama te odgovarajuće upravljanje ključevima koje treba čuvati sigurno, odvojeno od podataka, te po mogućnosti koristiti module za sigurno pohranjivanje ključeva.

Migracija podataka u oblak donosi posebne izazove sigurnosti, koje arhitektura nultog povjerenja nastoji adresirati. Osim kriptiranja u prijenosu i u mirovanju, ovdje je važna i kontrola pristupa podacima u oblaku. To uključuje korištenje posrednika sigurnosti pristupa u oblaku (*Cloud Access Security Broker* - CASB) za nametanje sigurnosnih politika prilikom pristupa uslugamau u oblaku, te oznaka povjerljivosti i klasifikacija podataka kako bi se odredilo koje podatke smije vidjeti određeni korisnik ili sustav. Pristup nultog povjerenja potiče i koncept „*bring your own key*“, gdje organizacije upravljaju vlastitim kriptografskim ključevima za šifriranje podataka u oblaku, umjesto da se u potpunosti oslanjaju na zadane ključeve pružatelja. Tako se održava veća razina kontrole i nepovjerenja prema trećim stranama. Cjelovita zaštita podataka u modelu nultog povjerenja zahtijeva integraciju kriptografije u sve faze životnog ciklusa podataka, od kreiranja, preko prijenosa, do pohrane i korištenja, uz stalnu provjeru prava pristupa i integriteta tih podataka.

## 4.3 Praćenje i detekcija prijetnji

U arhitektur nultog povjerenja, stalno praćenje aktivnosti i brza detekcija odstupanja, su od kritične važnosti za preduhitriti sigurnosne incidente. Budući da se ni autentificirani i autorizirani korisnici/uređaji ne smatraju u potpunosti pouzdanima, sustav mora stalno pratiti njihovo ponašanje i događaje u mreži ne bi li uočio znakove potencijalne kompromitacije ili zlouporabe.

SIEM sustavi prikupljaju zapise (logove) i događaje iz mrežne opreme, poslužitelja, aplikacija, sigurnosnih uređaja te ih centralizirano pohranjuju i analiziraju u stvarnom vremenu. SIEM služi kao centralni „mozak“ za praćenje sigurnosti koji korelira potencijalno nepovezane događaje kako bi otkrio sumnjive obrasce, upozorava na incidentne situacije i pruža uvid u sve aktivnosti korisnika i uređaja [13]. Primjerice, korisnik nakon uspješne autentifikacije odjednom pokuša pristupiti velikom broju osjetljivih datoteka što nije njegovo uobičajeno ponašanje, SIEM može takav niz događaja označiti sumnjivim. Moderne SIEM platforme imaju i ugrađene mehanizme za detekciju incidenata pomoću pravila i strojne obrade podataka, a mogu se integrirati s ostalim sigurnosnim alatima (vatrozid, IDS/IPS) kako bi automatski pokrenuli odgovore na detektirane prijetnje. Za arhitekturu nultog povjerenja je važno da SIEM pokriva cijeli sustav, od lokalne infrastrukture do servisa u oblaku i da je postavljen na principe stalne verifikacije (npr. generira upozorenja ako se detektira pristup bez prethodne odgovarajuće autentifikacije ili neuspjeli pokušaji pristupa zabranjenim segmentima mreže).

Analitika ponašanja korisnika *(User Behavior Analytics* - UBA) je klasični pristup detekciji prijetnji oslanja se na poznate uzorke napada ili unaprijed definirana pravila, što nije uvijek dovoljno za složene ili napade novije vrste. UBA donosi strojno učenje i analitiku u praćenje sigurnosti. Sustavi analitike ponašanja prvo prikupljaju velik broj podataka o aktivnostima korisnika i uređaja tijekom duljeg razdoblja kako bi izgradili bazni profil „normalnog“ ponašanja za svakog korisnika ili entitet. Nakon toga, pomoću algoritama strojnog učenja UBA sustav kontinuirano prati odstupanja od tog uobičajenog obrasca. Ukoliko korisnik obično pristupa sustavima u radno vrijeme iz Zagreba, a odjednom se prijavljuje noću s IP adrese u drugoj državi, ili počne preuzimati neuobičajeno veliku količinu podataka, UBA će to prepoznati kao anomaliju. Takve anomalije mogu ukazivati na ugrožavanje korisničkog računa ili unutarnju prijetnju. U arhitekturi nultog povjerenja, gdje se pretpostavlja da čak i ovlašteni korisnik može postati prijetnja (namjerno ili zbog ugroze), UBA značajno pomaže u stalnoj provjeri. UBA može dinamički zahtijevati dodatnu provjeru (npr. ponovnu MFA autentifikaciju) ako detektira neuobičajeno ponašanje korisnika ili jednostavno obavijestiti sigurnosno osoblje o sumnjivim aktivnostima [14]. Integracijom UBA rezultata sa SIEM-om i sustavima za reakciju (SOAR), organizacije mogu pravovremeno reagirati na moguće napade prije nego što prouzroče štetu.

Osim UBA, šire korištenje umjetne inteligencije i strojnog učenja u sigurnosnom nadzoru daje dodatnu snagu modelu nultog povjerenja. Umjetna inteligencija (AI) može u realnom vremenu analizirati goleme količine podataka iz mrežnog prometa i sustava te identificirati obrasce koji ukazuju na napad ili zlonamjerno djelovanje, čak i kada ti obrasci nisu unaprijed poznati. AI modeli mogu nadzirati kriptirani mrežni promet i na temelju metapodataka (učestalost paketa, veličine itd.) detektirati komunikaciju zlonamjernog softvera unutar inače šifriranog kanala [6]. Strojno učenje (ML) se koristi za treniranje tih modela na temelju legitimnog prometa i poznatih prijetnji, kako bi sustav naučio razlikovati normalno od zloćudnog. U praksi, moderni sigurnosni alati integrirani u arhitekturu nultog povjerenja koriste ML modele za otkrivanje anomalija i potencijalnih sigurnosnih povreda te često imaju komponente ugrađene inteligencije. Takvi sustavi mogu automatski izolirati uređaj ili korisnički račun kada uoče sumnjivo ponašanje, čime se ograničava šteta prije nego što analitičar uopće stigne reagirati. Važno je napomenuti da AI/ML sustavi trebaju kontinuirano osvježavanje i nadzor eksperta kako bi ostali učinkoviti i kako bi se izbjegli lažno pozitivni ili lažno negativni rezultati. Uvođenjem AI i strojnog učenja, arhitektura dobiva dinamičku komponentu, prilagođava se novim prijetnjama brže nego što bi to bilo moguće ručnim praćenjem, što je ključno u današnjem okruženju evoluirajućih kibernetičkih napada.

## 4.4 Segmentacija mreže

Tradicionalne mreže često su podijeljene na interne zone povjerenja i vanjski nepouzdani segment, s jakim vatrozidom na granici. Model nultog povjerenja odbacuje ideju širokog pouzdanog perimetra i umjesto toga uvodi segmentaciju mreže kako bi se ograničio domet napada i kretanje potencijalnih napadača. Dva ključna koncepta koja se koriste za segmentaciju u arhitekturama nultog povjerenja su *Software-Defined Perimeter* (SDP) i mikrosegmentacija.

Softverski definirani perimetar (SDP) je arhitektura u kojoj se pristup resursima dozvoljava tek nakon što se entitet kriptografski autentificira i autorizira kroz kontrolni sustav, čineći resurse nevidljivima dok se ne ispune uvjeti pristupa. Za razliku od tradicionalnih mreža gdje su određeni servisi vidljivi na mreži (npr. otvoreni portovi koji čekaju povezivanje), SDP pristup stvara svojevrsni „nevidljivi omotač“ oko resursa. Komponente SDP-a tipično uključuju kontrolor (koji provjerava identitet, kontekst i politiku za zahtjev pristupa) i agente (koji uspostavljaju siguran tunel ka resursu ako je pristup odobren). Tek kada kontrolor potvrdi da korisnik/uređaj smije pristupiti određenom resursu, tom se korisniku dinamički izgrađuje privatan povjerljivi put do tog resursa, npr. otvara se privremeni port i uspostavlja kriptirana veza. Svi ostali neautorizirani pokušaji ostaju nevidljivi i odbijeni. Ovakav pristup efektivno skriva internu infrastrukturu od promatranja i smanjuje napadačku površinu: skeneri ne mogu pronaći otvorene portove ili usluge ako prethodno nisu prošli strogu provjeru identiteta. SDP koncept prvotno je promoviran od strane *Cloud Security Alliance*, a praktičnu primjenu ima u mnogim modernim mrežnim pristupima bez povjerenja (ZTNA) rješenjima. U njima se klasični VPN zamjenjuje SDP principom. Rezultat je da se svakom korisniku omogućuje pristup isključivo onim pojedinačnim aplikacijama ili uslugama za koje je autoriziran, bez mogućnosti da „lutaju“ po ostatku mreže. Time se uvelike smanjuje mogućnost laterlanog pokreta napadača čak i ako kompromitiraju jednog korisnika ili uređaj.

Mikrosegmentacija ide korak dalje od klasične segmentacije mreže dijeljenjem mreže na vrlo male segmente, potencijalno sve do razine pojedinačne aplikacije, virtualne mašine ili kontejnera. Svaki segment ima definirane sigurnosne politike koje određuju koji subjekti ili sustavi smiju komunicirati s tim segmentom. Na primjer, baza podataka može biti jedan segment kojem mogu pristupiti samo aplikacijski poslužitelji segmentirani u drugoj zoni, i to isključivo preko potrebnih mrežnih protokola. Sav ostali promet prema bazi bio bi blokiran. Ovakav pristup efektno ograničava kretanje, čak i ako napadač kompromitira jedan dio sustava, ne može lako prodrijeti dalje jer je svaki sljedeći korak druga segment sa zasebnim provjerama. Tehnološka implementacija mikrosegmentacije često se oslanja na virtualizacijske i alate u oblaku (npr. VMWare NSX[[10]](#footnote-10)) koji omogućuju definiranje mrežnih politika na razini virtualnih NIC-ova [[11]](#footnote-11)ili kroz softverski definirane mreže (SDN). Pravila se mogu temeljiti na identitetu aplikacija, oznakama, uslugama i kontekstu, a ne samo na IP adresama i portovima kao u tradicionalnim ACL-ovima[[12]](#footnote-12). Arhitektura nultog povjerenja koristi mikrosegmentaciju kako bi svaka komponenta bila okružena vlastitim „mikroperimetrom“. Time se postiže visoka izolacija resursa. U slučaju da je korisnička stanica zaražena, pravilno mikrosegmentirana mreža spriječit će da malware dođe do kritičnih poslužitelja jer neće ispuniti kriterije politike za te segmente. Važno je napomenuti da pravilna konfiguracija segmentacijskih pravila zahtijeva detaljno poznavanje sustava i komunikacijskih tokova – greške u konfiguraciji mogle bi ostaviti neke dijelove mreže izloženima ili nenamjerno onemogućiti legitimnu komunikaciju [12]. Stoga se preporučuje postupno uvoditi mikrosegmentaciju, uz praćenje i podešavanje pravila, i koristiti alate za automatsko predlaganje politika na temelju uočenog prometa. Kada je ispravno proveden, koncept mikrosegmentacije značajno podiže razinu sigurnosti mreže bez dodatnog utjecaja na performanse sustava [13].

## 4.5 Potrebni resursi za implementaciju arhitekture nultog povjerenja

Uvođenje modela nultog povjerenja u organizaciju zahtijeva pažljivo planiranje resursa, i softverskih i hardverskih, koji će omogućiti provedbu opisanih tehnologija. Za uspješnu implementaciju potrebno je osigurati sljedeće komponente.

Prvi i najvažniji element je središnji sustav za upravljanje identitetima korisnika i uređaja. To može biti usluga u oblaku(poput Okta, Azure AD) ili lokalni sustav (npr. *Microsoft Active Directory* u kombinaciji s ADFS - *Active Directory Federation Services*) koji podržava moderne protokole (SAML - *Security Assertion Markup Language*, OAuth/OIDC) i MFA integraciju. Također, potreban je i PAM (*Privileged Access Management*) alat za kontrolu privilegiranih računa ako organizacija raspolaže osjetljivim administrativnim pristupima. Nadalje, treba uvesti infrastrukturu za MFA, bilo kroz eksternog providera ili lokalnom rješenje, uključujući aplikacije za generiranje kodova, push notifikacije ili fizičke token uređaje. Za mrežnu segmentaciju i pristup, organizacija će trebati softver za SDP/ZTNA, koji može biti u obliku servisa u oblaku (npr. Zscaler ZTNA) ili softvera koji se instalira u vlastitom podatkovnom centru i na klijentske uređaje. Također, SIEM platforma je ključna, nju je moguće ostvariti kroz alate poput Splunk ili SIEM rješenja u oblaku, ovisno o veličini i potrebi organizacije. SIEM često dolazi ruku pod ruku s platformom koja automatizira odgovore na incidente. Implementacija takve platforme preporučljiva je kako bi se pravovremeno reagiralo na detektirane prijetnje. Za UBA/UEBA funkcionalnost, organizacija može koristiti samostalna rješenja (npr. Securonix, Vectra AI) ili iskoristiti module unutar postojećeg SIEM/SOAR rješenja koji obavljaju analitiku ponašanja. Sustavi za upravljanje ključevima (KMS) ili certifikacijska infrastruktura (PKI) još su jedan potrebni element, oni osiguravaju izdavanje i upravljanje certifikatima/ključevima za šifriranje komunikacija i podataka, što je temelj za „*encrypt everywhere*“ pristup arhitekture nultog povjerenja. U kontekstu zaštite krajnjih točaka, poželjne su moderne krajnje točke kao klijenti koji mogu provoditi politike na uređajima koje će provjeravati integritet uređaja, verziju OS-a, prisutnost sigurnosnih zakrpa, antivirusne statuse itd., te bi to stanje javljali centralnom sustavu koji odlučuje smije li uređaj pristupiti resursima. Većina navedenih softverskih komponenti može biti integrirana, primjerice, jedan poslužitelj može isporučiti i identitetski sustav i mrežni pristupnik (*gateway*) i SIEM, ili se mogu orkestrirati zajedno putem otvorenih API-ja ako su od različitih dobavljača.

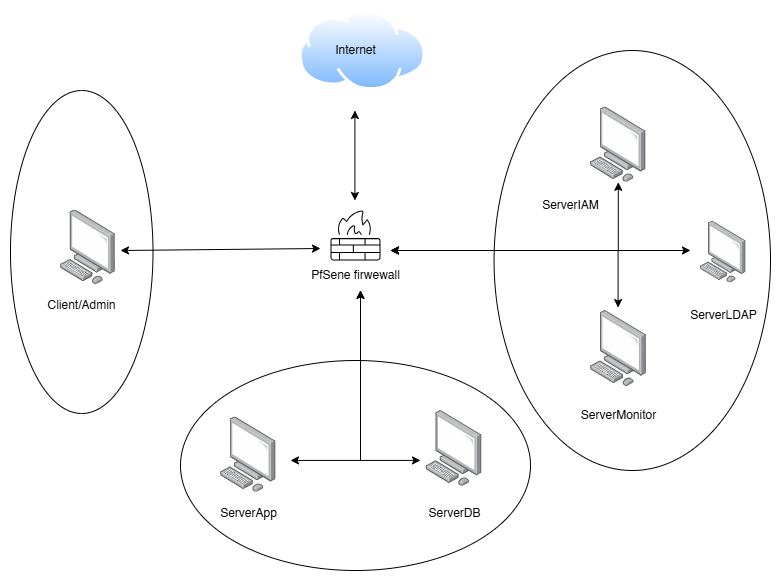
Iako arhitektura nultog povjerenja naglasak stavlja na softver i identitet, određeni hardverski elementi su potrebni za podršku i provođenje tih sigurnosnih politika. Prvo, organizacija mora imati mrežnu opremu koja podržava virtualizaciju mrežnih funkcija i segmentaciju, mrežne preklopnike (*switch*) i usmjernik (*router*) koji mogu provoditi VLAN segmentaciju, vatrozide ili SDN kontrolere koji omogućuju mikrosegmentaciju u podatkovnom centru. Mnogi od tih segmenata zapravo će biti implementirani softverski (npr. unutar hipervizora ili oblaka), ali za lokalnu mrežu često će biti potrebna nadogradnja postojeće mrežne opreme ili barem nadogradnja firmvera kako bi se mogle primijeniti politike pristupa. Dalje, kod provjere identiteta korisnika, često se koriste hardverski tokeni ili uređaji za MFA, poput pametnih kartica, USB tokena ili čak biometričkih skenera, što spada u potrebne resurse za distribuciju korisnicima. Za PKI infrastrukturu, organizacija može trebati HSM (*Hardware Security Module*) uređaje koji sigurno pohranjuju kriptografske ključeve i vrše operacije šifriranja/potpisivanja, osiguravajući visoku razinu zaštite privatnih ključeva. Još jedan aspekt su poslužitelji i pohrana za sve nove sustave – primjerice, ako se SIEM uvodi lokalno, trebat će dovoljno snažni poslužitelji i memorija za obradu i pohranu velike količine zapisa, ako se koristi analitika ponašanja u realnom vremenu, treba osigurati i računalne resurse (GPU) za pokretanje ML modela na premisi, ukoliko se ne koristi usluga u oblaku. Također, za implementaciju SDP-a na premisi mogu biti potrebni specifični serveri koje je potrebno postaviti ispred zaštićenih resursa. Ukratko, infrastrukturna spremnost vrlo je važna: organizacija mora imati mrežnu i računalnu infrastrukturu sposobnu podržati povećano šifriranje prometa (što može opteretiti starije procesore na vatrozidima), centralizirano logiranje, te mnoštvo dodatnih provjera pri svakom pristupu. Planiranje kapaciteta i moguće nadogradnje hardvera stoga su dio implementacije arhitekture nultog povjerenja.

Iako nisu „tehnologija“ u užem smislu, vrijedno je napomenuti da uspješna implementacija zahtijeva i ulaganje u ljudske resurse (obuku osoblja, možda nova zapošljavanja stručnjaka za područja poput podatkovne analitike ili upravljanja identitetima) te procese. Primjerice, uvođenje rigoroznog nadzora i autentifikacije možda će zahtijevati prilagodbu postojećih poslovnih procesa i tijekova rada. Ključni „resurs“ je i podrška višeg menadžmenta, jer arhitektura nultog povjerenja nije jednokratan proizvod nego strategija koja obuhvaća čitavu organizaciju. U smislu tehnologije, međutim, gore nabrojani programi i hardver tvore okosnicu ekosustava nultog povjerenja. Mnoge organizacije već posjeduju neke od tih komponenti kao dio klasične sigurnosne infrastrukture, ali arhitektura nultog povjerenja traži njihovu tijesnu integraciju i konfiguraciju prema načelima modela nultog povjerenja. Uz odgovarajuće resurse i plan, prijelaz na arhitekturu nultog povjerenja rezultira značajno ojačanom sigurnosnom pozicijom organizacije, smanjenom površinom napada te većom otpornošću na kibernetičke prijetnje u modernom, distribuiranom IT okruženju.

# 5. Praktični dio: Implementacija i testiranje

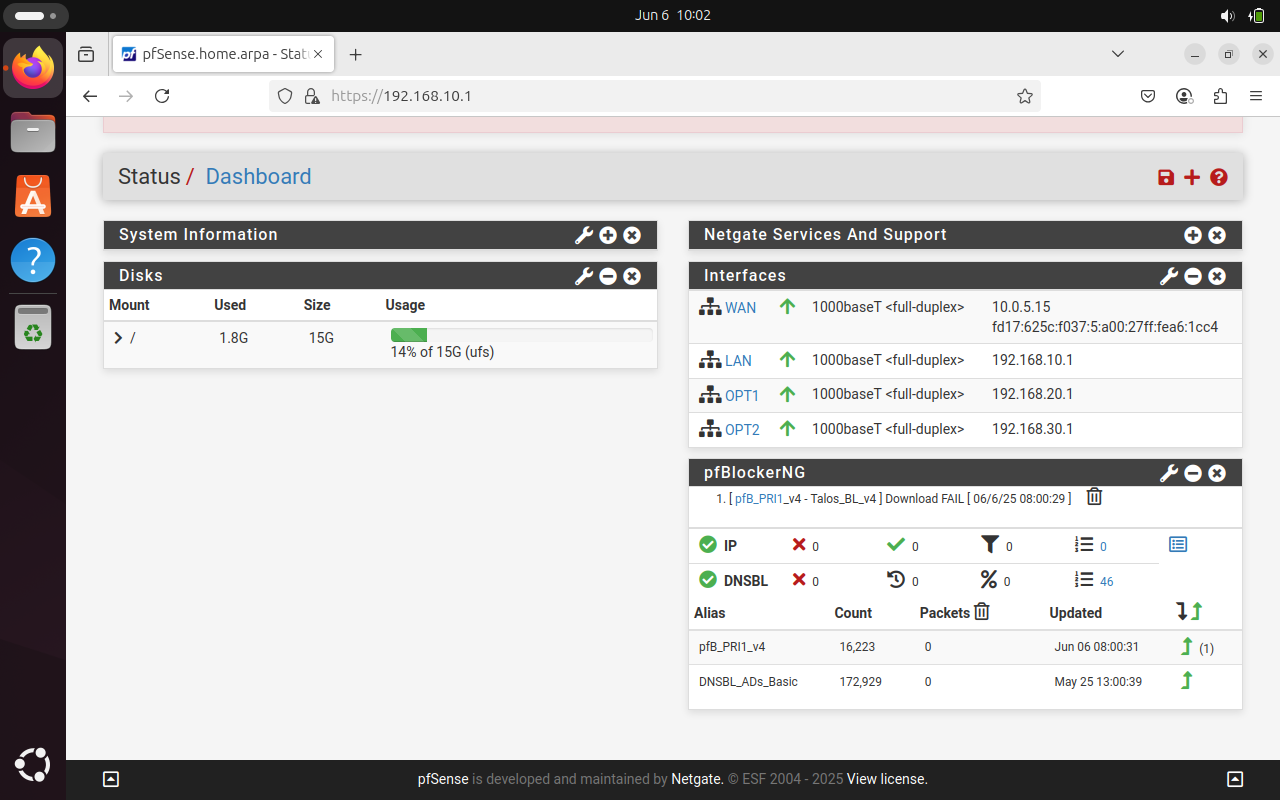
## 5.1 Implementacija modela u simuliranom okruženju

U ovom poglavlju opisuje se postupna implementacija sigurnosnog modela u izoliranom simuliranom okruženju (Slika 3). Arhitektura nultog povjerenja pristup temelji se na načelu da se niti jednom entitetu ili mrežnom segmentu ne vjeruje implicitno, već se svaki pristup strogo provjerava neovisno o njegovu porijeklu. Kako bi se to postiglo, potrebno je primijeniti više sigurnosnih mehanizama, od mikrosegmentacije mreže, preko snažne autentifikacije i autorizacije korisnika, do kontinuiranog nadzora i analize prometa i aktivnosti. U nastavku je kroz narativni pregled detaljno prikazana implementacija ovih elemenata u praktičnom okruženju s više virtualnih poslužitelja i mrežnih segmenata.



Slika 3: Arhitekutra implementirane arhitekture nultog povjerenja

### 5.1.1 Mrežna segmentacija i osiguranje komunikacije

Mrežna arhitektura postavljena je tako da odražava ključni princip segmentacije mreže radi suzbijanja lateralnog kretanja napadača. Okruženje se sastoji od tri odvojena mrežna segmenata: aplikacijski, klijent i upravljački dio . Svaki segment predstavlja zasebnu sigurnosnu zonu s točno definiranim ulogama, aplikacijski segment sadrži dva servera (bazu podataka i server sa aplikacijom). Upravaljčki segment sadrži server sa sustavom za upravljanje identitetima (Keycloak) i server za nadgledanje koji obuhvaća komponente za prikupljanje zapisa i nadzor sustava. Klijentski dio predstavlja mrežu iz koje dolaze korisnici ili administratori. Svi ovi segmenti međusobno su izolirani na razini drugog sloja mrežnog modela te se njihov promet može susresti samo preko centralnog vatrozida (Slika 4).

Slika 4: Konfiguracija pfSense vatrozida s definiranim mrežnim sučeljima

U implementaciji je korišten pfSense kao središnji mrežni vatrozid i usmjernik koji povezuje navedene segmente. PfSense je vatrozid otvorenog koda koji podržava višestruke mrežne adaptere i omogućuje napredno definiranje pravila prometa između segmenata. U virtualnom okruženju pfSense je dobio nekoliko virtualnih mrežnih sučelja, jedno sučelje je označeno kao WAN i povezano na NAT mrežu (radi omogućavanja izlaska na internet za potrebe instalacije nadogradnji i sličnih operacija), dok su ostala sučelja povezana na interne mreže VirtualBox-a koje odgovaraju ostalnim mrežnim segmentima. Svaki od tih internih segmenata implementiran je kao odvojena VirtualBox interna mreža, što znači da je komunikacija moguća samo među virtualnim strojima unutar istog segmenta, dok je izolirana od računala poslužitelja i drugih segmenata. PfSense na svakom od tih sučelja ima statičku IP adresu koja služi kao mrežni pristupnik za virtualne strojeve. Takva arhitektura odražava mikrosegmentaciju, mreža je podijeljena na manje, izolirane segmente zbog fine kontrole pristupa i ograničavanja lateralnog pokreta potencijalnog napadača.

Svaki segment ima jasnu funkciju i minimalan skup dozvola za komunikaciju s drugim segmentima, što je temelj sigurnosti u modelu nultog povjerenja. Konfiguracija pfSense vatrozida odražava načelo „*deny-by-default*“. PfSense po svojoj naravi radi kao „*stateful*“ vatrozid te inicijalno blokira sav promet između segmenata ukoliko se ne definiraju izričito pravila koja ga dopuštaju. Odmah nakon inicijalne instalacije, pfSense samo na prvom sučelju automatski omogućuje sav izlazni promet prema internetu, kako bi osnovna povezanost bila uspostavljena za tipične korisnike kućnih mreža. Međutim, za svako dodatno sučelje koje se doda ne postoji nikakvo predefinirano pravilo, promet je po zadanoj vrijednosti blokiran dok se ručno ne dopusti ono što je potrebno.

U skladu s arhitekturom nultog povjerenja, upravo takav strog početni položaj je poželjan jer osigurava da nijedna mrežna komunikacija između segmenata nije moguća bez eksplicitnog odobrenja. Nakon kreiranja sučelja u pfSense-u za svaki segment, pristupa se definiranju preciznih pravila vatrozida koja dopuštaju samo nužan promet između određenih zona. Pri definiranju pravila vatrozida, vodilo se računa o načelu najmanjih privilegija u mrežnom kontekstu. Primjerice, aplikacijski poslužitelj u aplikacijskom segmentu treba komunicirati s bazom podataka isključivo na portu 5432 za PostgreSQL, stoga je izrađeno pravilo koje dozvoljava promet s aplikacijskog poslužitelja prema bazi podataka samo na tom odredišnom portu. Svi ostali portovi ili neautorizirane komunikacije između aplikacijskog poslužitelja i baze podatka su blokirane. Slično tome, aplikacijski poslužitelj treba uslugu autentifikacije od Keycloak servera u IAM (*Identity and Access Management*) segmentu, što se odvija putem standardnog HTTPS protokola. Definirano je pravilo koje dozvoljava promet iz aplikacijskog segmenta prema upravljačkom segmentu na server sa Keycloakom na port 8080, omogućujući tako da aplikacija dosegne Keycloak API i poslužitelj za potrebe provjere tokena i autentifikacije korisnika. Istovremeno, pristup iz aplikacijskog segmenta prema bilo kojoj drugoj usluzi u upravljačkoj mreži nije dozvoljen. Za korisnički segment postavljena su pravila koja dopuštaju korisnicima pristup web aplikaciji u APP segmentu na port 5000 kao i pristup Keycloak sustavu u IAM segmentu na port 8080, budući da korisnički web preglednik mora dohvatiti i Keycloak stranicu za prijavu prilikom autentifikacijskog procesa. Time je omogućeno da krajnji korisnik iz svoje mreže može inicirati autentifikaciju i koristiti aplikaciju, ali pritom i dalje nema direktnog pristupa bazi podataka niti upravljačkom segmentu, čime se minimizira doseg potencijalno kompromitiranog korisničkog računala. U pfSense-u su također definirana pravila za omogućavanje nužnih infrastrukturnih servisa unutar segmentiranog okruženja. Na primjer, DNS rezolucija je centralizirana – pfSense poslužitelj djeluje kao DNS rekurzivni poslužitelj za klijente. Prema tome, svim internim segmentima dozvoljen je DNS promet (UDP/TCP port 53) isključivo prema pfSense-ovoj IP adresi na odgovarajućem sučelju. Tako se osigurava da virtualni poslužitelji mogu razriješiti imena (i pfSense po potrebi prosljeđuje upite prema vanjskim DNS serverima preko WAN sučelja), ali ne mogu slobodno slati DNS upite van svojih segmenata ili na internet osim kroz pfSense. Sličan princip vrijedi i za ostale osnovne servise: npr. u svrhu sinkronizacije vremena, pfSense može djelovati i kao NTP poslužitelj ili se mogu dopustiti NTP upiti iz određenih segmenata prema internetu, no takva su dopuštenja pod strogom kontrolom.

Slika 5 prikazuje primjer skupa pravila vatrozida unutar pfSense sučelja koja implementiraju segmentaciju, vidljivo je da za ovaj segment, kao i za svaki drugi, postoji skup dopuštenih komunikacija prema drugim segmentima ili internetu, dok je ostali promet blokiran. Ovakvom granuliranom politikom, svaki mrežni paket mora zadovoljiti određeni uvjet kako bi prošao kroz vatrozid, čime se postiže visoka razina kontroliranog pristupa u skladu sa paradigmom nultog povjerenja. Treba istaknuti da pfSense u ovakvoj ulozi služi kao segmentacijska brana koja provodi načela nultog povjerenja na mrežnoj razini. Svaki zahtjev između različitih zona mora proći kroz pfSense gdje se provjerava pravilo koje ga eksplicitno dopušta. Budući da ne postoji „prečac“ niti implicitno povjerenje među segmentima, značajno se smanjuje mogućnost da napadač, u slučaju kompromitacije jedne komponente, jednostavno zaobiđe sigurnosne kontrole i proširi napad dalje po sustavu.

Slika na kojoj se prikazuje tekst, snimka zaslona, softver, broj

Sadržaj generiran uz AI možda nije točan.

Slika 5: Skup pravila vatrozida

Ovo je praktična realizacija načela „nikad ne vjeruj, uvijek provjeravaj“ na mrežnoj razini. Uz to, pfSense omogućuje i *Network Address Translation* (NAT) za izlazni promet prema vanjskom svijetu. U ovoj implementaciji NAT je omogućen na WAN sučelju pfSense-a te se svi dozvoljeni odlasci prema Internetu (npr. za potrebe sustava za nadzor koji možda treba preuzeti određene pakete, ili za ažuriranje softvera na poslužiteljima) prevode na pfSense-ovu WAN IP adresu. Time se dodatno skriva interna struktura mreže. Važno je napomenuti da pristup iz vanjske mreže (Internet) prema unutarnjim segmentima nije općenito dozvoljen, osim možda kroz specifične VPN ili mehanizme proslijeđivanja portova ako bi se demonstrirao pristup izvana, no u duhu arhitekture nultog povjerenja, i takav bi pristup trebao biti strogo autentificiran i ograničen. U simuliranom okruženju fokus je prvenstveno bio na internim komunikacijama među segmentima, uz pretpostavku da su vanjski pristupi ograničeni.

Također, u skladu s arhitekturom nultog povjerenja implementirana je komunikacija putem SSL protokola, s ciljem osiguravanja enkriptiranog prijenosa podataka između lokalnih servisa i korisničkih sustava. Kako bi se omogućila sigurna komunikacija u testnom okruženju, izrađeni su lokalni SSL certifikati na Ubuntu operacijskim sustavima. Umjesto korištenja certifikata izdanih od strane službenog certifikacijskog tijela, primjenjena je metoda generiranja samopotpisanih certifikata korištenjem alata OpenSSL[[13]](#footnote-13). Tijekom tog procesa definirani su osnovni elementi certifikata, uključujući zajedničko ime, naziv organizacije te razdoblje valjanosti certifikata. Dobiveni certifikat i pripadajući privatni ključ implementirani su u lokalne servise kako bi se omogućila komunikacija putem HTTPS protokola. S obzirom na to da preglednici poput Firefoxa prema zadanim postavkama ne prepoznaju samopotpisane certifikate kao pouzdane, izvršen je ručni unos certifikata u lokalni certifikacijski repozitorij preglednika. Na taj je način izbjegnuto prikazivanje sigurnosnih upozorenja te je omogućeno testiranje aplikacija i servisa unutar sigurnog i lokalnog konteksta.

### 5.1.2 Integracija sustava za autentifikaciju

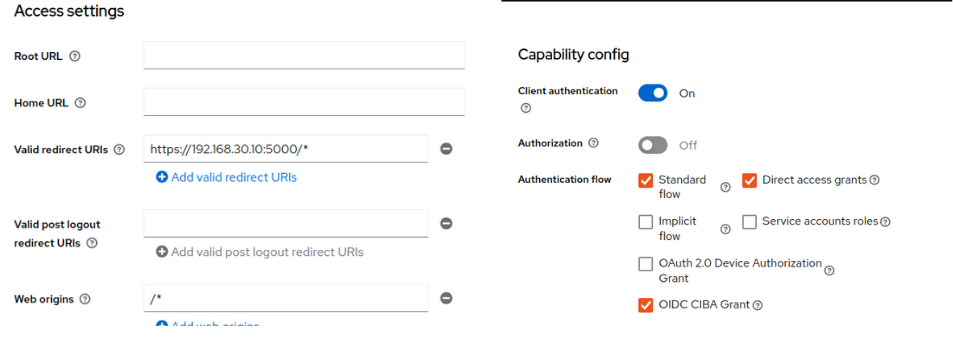
Kao centralni dio sustava za provjeru identiteta i autorizaciju odabran je Keycloak, sustav otvorenog koda za upravljanje identitetima i pristupima (IAM). Keycloak je moćan sustav koji nudi brojne moderne autentifikacijske značajke, što ga čini pogodnim za izgradnju okruženja usklađenog načelima nultog povjerenja.

U ovom modelu identitet korisnika i uređaja postaje nova granica, svaki zahtjev treba biti autentificiran i autoriziran, neovisno o tome odakle dolazi. Keycloak je implementiran u izdvojenom IAM segmentu, čime se postiže dodatna izolacija usluge za provjeru identiteta. Ovakav pristup osigurava da samo ovlašteni servisi i klijenti (poput aplikacije u aplikacijskom segmentu) mogu komunicirati s IAM sustavom, i to kroz strogo definirana pravila, dok ostatak mreže nema pristup tom segmentu. U praktičnom smislu, postavljen je zaseban virtualni stroj na kojem je instaliran Keycloak server. Unutar Keycloak-a se konfigurira realm (pojam u Keycloak terminologiji koji predstavlja izoliranu domenu upravljanja identitetima, možemo zamisliti realm kao sigurnosnu domenu specifičnu za našu aplikaciju). Unutar realma je klijent za Flask aplikaciju, entitet koji predstavlja samu aplikaciju kao klijenta koji će koristiti Keycloak za autentifikaciju korisnika. Pri kreiranju klijenta, dodijeljen mu je jedinstveni identifikator i konfigurirana je odredišna adresa na koju će Keycloak preusmjeriti korisnika nakon uspješne prijave.

Korisnik pristupa web aplikaciji, biva preusmjeren na Keycloak login (koji je smješten u IAM segmentu), uspješno se prijavljuje te ga Keycloak vraća u aplikaciju s potrebnim tokenima. Flask aplikacija zatim pomoću dobivenog tokena prepoznaje korisnika i omogućuje pristup zaštićenim funkcionalnostima. U slučaju da korisnik nije autoriziran za određenu akciju ili pokuša pristupiti resursu za koji nema prava, aplikacija (ili Keycloak posredno) to onemogućuje, držeći se načela da se nikome ne vjeruje samo na temelju prethodne autentifikacije ili položaja u mreži. Nakon uspostave osnovne autentifikacije slijedi višefaktorska autentifikacije (2FA) za korisničke račune, u svrhu dodatnog učvršćivanja povjerenja u identitet. Keycloak podržava više metoda 2FA, uključujući jednokratne lozinke temeljene na vremenu (TOTP) generirane u aplikacijama poput Google Authenticator ili FreeOTP, kao i moderne standarde poput WebAuthn za autentifikaciju putem biometrije ili fizičkih sigurnosnih ključeva

Konkretno, korisnik bi nakon unosa lozinke bio upitan da skenira QR kod u svojoj mobilnoj autentifikatorskoj aplikaciji te generira inicijalni kôd, kojim potvrđuje povezanost aplikacije i svog korisničkog računa. Nakon toga, prilikom svake prijave korisnik mora osim lozinke unijeti i aktualni jednokratni kod sa svog mobilnog uređaja. Ova dodatna razina sigurnosti značajno otežava zlouporabu kompromitiranih lozinki, u skladu s principom „uvijek provjeravaj“. Osim 2FA, Keycloak pruža i druge pogodnosti kao što su centralizirano upravljanje korisničkim ulogama i pravima, jedinstvena prijava (SSO) za više aplikacija, kao i mogućnost integracije s vanjskim identitetskim. Sve su te značajke u skladu s filozofijom nultog povjerenjajer omogućuju finu kontrolu nad time tko može pristupiti kojim resursima i pod kojim uvjetima. Važno je napomenuti da integracija aplikacije s Keycloak IAM sustavom nije samo tehnički pogodna, nego i konceptualno usklađena s načelima nultog povjerenja. Identitet korisnika postao je glavni kriterij dozvole pristupa, sama činjenica da se korisnik nalazi unutar internog mrežnog segmenta ne daje mu nikakve posebne povlastice dok se ne autentificira i ne dokaže svoj identitet putem Keycloak-a. Tek nakon uspješne autentifikacije (i eventualno autorizacije u smislu provjere uloga/permisija) aplikacija će mu dopustiti odgovarajuće akcije. Svaki daljnji zahtjev aplikacija može uvjetovati posjedovanjem ispravnog tokena, čime se uvodi kontinuirana provjera tijekom trajanja sesije. Keycloak izdaje vremenski ograničene tokene, što znači da ako bi došlo do kompromitacije sjednice ili odugovlačenja komunikacije, pristup može biti onemogućen po isteku tokena ili ranije ako se korisnika aktivno odjavi ili mu se oduzmu privilegije. Tako se ostvaruje načelo da se sigurnost ne oslanja na jedinstvenu točku provjere, već je proces koji traje.

Slika 6 prikazuje administrativno sučelje Keycloak sustava s konfiguracijom klijenta za Flask aplikaciju, uključujući postavke dozvoljenih preusmjeravanja (redirect URI) i opcije vezane uz autentifikaciju (OIDC parametre). Na slici se može vidjeti primjer kako je definiran klijent sa URL-ovima na koje Keycloak smije preusmjeravati nakon prijave i odjave. Uz navedeno, u ostalim dijelovima defeniramo što želimo raditi prisiljavanje korisnika na određene radnje (uspostava dodatne autentifikacije), izrada novih korisnika definiranje novih sučelja prilikom prijave i dr.



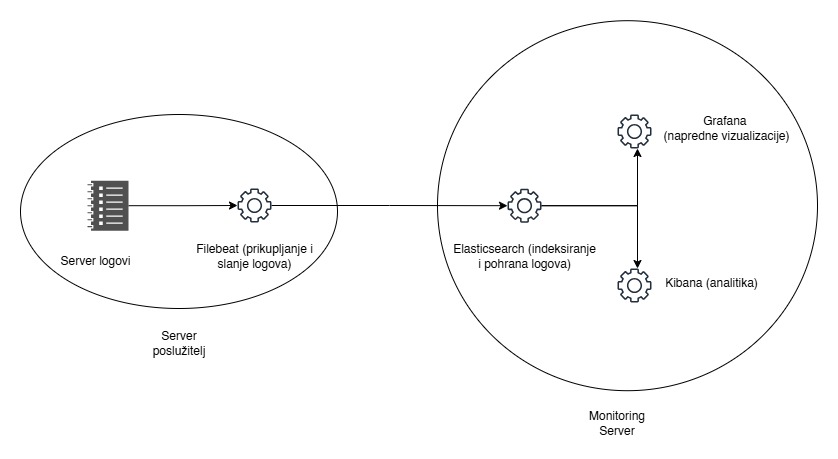
Slika 6: Konfiguracija Keycloaka za Flask aplikaciju

U okviru provedbe sustava za autentifikaciju temeljenog na LDAP direktoriju, implementirano je rješenje koristeći slapd servis kao središnji direktorijski poslužitelj. Korisnike dodaje administrator sustava, te im određuje prava koja im pripadaju. Tako obični korisnik ,primjerice, neće imati pravo izvršavanja „sudo“ naredbi. U ovoj konfiguraciji, klijentski sustavi u potpunosti ovise o dostupnosti LDAP poslužitelja za autentifikaciju korisnika. To znači da korisnici ne mogu pristupiti sustavu, odnosno izvršiti prijavu, ukoliko slapd servis na udaljenom poslužitelju nije aktivan ili dostupan. Iako ovakav pristup omogućuje centralizirano upravljanje pristupom i korisničkim računima, on istovremeno unosi i određeni stupanj ovisnosti o mrežnoj infrastrukturi i dostupnosti autentifikacijskog servisa. Kako bi se osigurala mogućnost administracije i u slučajevima prekida veze s LDAP poslužiteljem, autor je dodatno konfigurirao lokalni korisnički račun s administrativnim ovlastima. Time je omogućeno održavanje i osnovno upravljanje sustavom čak i u uvjetima parcijalne nedostupnosti mrežnih servisa, čime je postignuta viša razina stabilnosti i operativnosti klijentskog računala. Time je dovršen sloj provjere identiteta u našem modelu.

### 5.1.3 Prikupljanje i analiza zapisa

Jedno od ključnih načela arhitekture nultog povjerenja je stalni nadzor i provjera aktivnosti unutar sustava. To uključuje sustavno prikupljanje zapisa sa svih relevantnih izvora i njihovu centraliziranu analizu, kako bi se u svakom trenutku moglo detektirati odstupanja od uobičajenog ponašanja, pokušaje neautoriziranih pristupa ili druge sigurnosne incidente. U simuliranom okruženju implementiran je centralizirani sustav prikupljanja zapisa korištenjem Filebeat za prikupljanje i transport zapisa, Elasticsearch za pohranu i indeksiranje, te Kibana za vizualizaciju i pretraživanje zapisa. Filebeat je odabran kao lagani agent koji se može pokrenuti na svakom poslužitelju (ili na centralnom poslužitelju) radi prikupljanja zapisa sustava. U našem slučaju, Filebeat je instaliran na središnjem upravljačkom poslužitelju na serveru za nadgledanje, te konfiguriran da prikuplja zapise iz više izvora. Prije svega, bilo je potrebno prikupljati zapise pfSense, osobito vatrozida, jer oni pružaju uvid u to koji je promet bio blokiran ili dozvoljen između segmenata, što je direktna provjera funkcionira li segmentacija u skladu s očekivanjima i ima li pokušaja neovlaštenog proboja između zona. PfSense, kao i većina mrežnih uređaja, podržava slanje zapisa na udaljeni *syslog* poslužitelj. Ta mogućnost je iskorištena tako da smo na pfSense-u uključili opciju „*Remote Logging*“. PfSense je konfiguriran da sve relevantne zapise (uključujući događaje vatrozida) šalje na IP adresu centralnog poslužitelja u skladu s uobičajenim *syslog* standardom.

Filebeat na monitoring poslužitelju ima uključen *syslog input* modul koji presreće dolazne poruke na tom portu. Tako svaki puta kada pfSense zabilježi, primjerice, blokirani paket između dvije mreže ili dopušteni izlaz korisnika iz mreže na internet, detalji tog događaja (vremenska oznaka, izvorna i odredišna adresa, port, akcija blokirano/dozvoljeno, i sl.) automatski se prosljeđuju u centralni poslužitelj. Uz pfSense zapise, Filebeat se koristio i za prikupljanje zapisa s ostalih sustava. Na poslužitelju aplikacije (aplikacijski segment) prikupljani su zapisi Flask aplikacije, primjerice zapisi o HTTP zahtjevima klijenata, eventualnim greškama na poslužitelju, čime se može pratiti obrasce korištenja aplikacije i eventualni pokušaji pristupa zaštićenim resursima bez autentifikacije. Također, na bazi podataka Filebeat je pratio PostgreSQL zapise, što je korisno za uočavanje neželjenih upita ili grešaka u upitima koji bi mogli ukazivati na pokušaje SQL injekcija ili sličnih anomalija. Keycloak poslužitelj generira vlastite zapise (događaje prijave, neuspjelih pokušaja prijave, administrativnih aktivnosti itd.), koje je također bilo moguće prikupljati Filebeat agentom. S obzirom da je Keycloak baziran na Java tehnologijama, njegovi zapisi su tekstualni zapisi koje Filebeat može čitati i slati dalje. Pritom je posebno zanimljivo uključivanje revizije i zapise događanja iz Keycloak-a, kako bismo u Kibani mogli vidjeti je li bilo neuspjelih pokušaja prijave (što može signalizirati napad ili kompromitiran račun) ili drugih sigurnosno značajnih događaja. Filebeat je konfiguriran da šalje prikupljene zapise direktno u Elasticsearch instancu (koja se vrti na poslužitelju za nadgledanje). Elasticsearch indeksira sve pristigle podatke i omogućuje njihovo brzo pretraživanje te agregaciju. Prikaz toka zapisa sustava prikazan je na Slika 7. Na temelju prikupljenih zapisa, postoji nekoliko preglednih vizualizacija u Kibani. Na primjer, Kibana nadzorna ploča koji u realnom vremenu prikazuje broj blokiranih mrežnih paketa po segmentima. Druga nadzorna ploča prikazuje uspješne i neuspješne prijave korisnika u sustav preko Keycloak-a. Kombinirajući te informacije, moguće je u svakom trenutku steći uvid u sigurnosno stanje sustava. Primjerice, ako se u zapisima sustava primijeti neuobičajeno velik broj neuspjelih prijava za nekog korisnika ili nagli skok u blokiranom prometu prema bazi, to se može protumačiti kao znak pokušaja kompromitacije te se može reagirati (npr. privremeno zaključati korisnički račun ili pooštriti pravila vatrozida za taj promet).

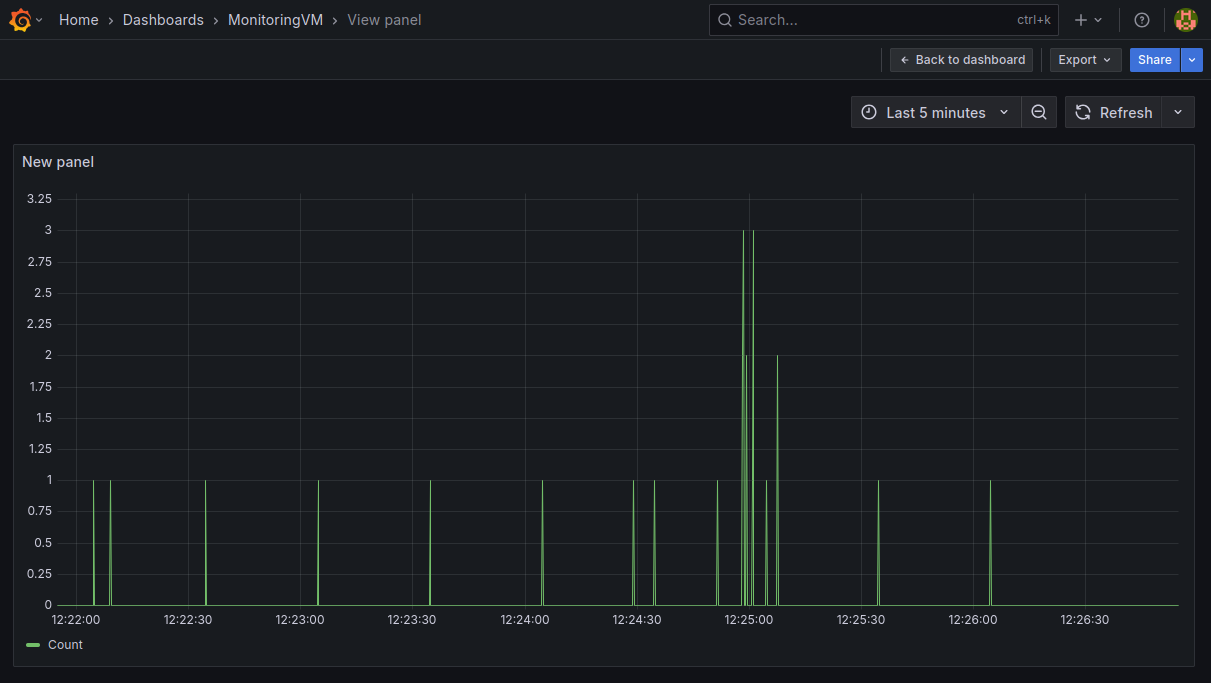


Slika 7: Prikaz toka agregiranih zapisa

Time se postiže puna vidljivost nad sustavom unatoč njegovoj segmentaciji, iako su mreže razdvojene, administratori putem zapisa i alata za analitiku mogu „vidjeti“ cjelinu i uočiti ako nešto ili netko pokušava probiti granice segmenata ili zaobići sigurnosne mehanizme. Takav pristup značajno povećava spremnost za odgovor na incidente i forenzičku analizu jer postoje trajni zapisi o svakoj važnijoj akciji u sustavu.

### 5.1.4 Nadzor performansi i sustava

Osim prikupljanja zapisa, okruženje zahtijeva i aktivno praćenje stanja sustava, ne samo u smislu sigurnosnih događaja, već i općeg zdravlja i performansi komponenti. Razlog tome je dvostruk. Stabilnost i dostupnost sustava je važna (jer sigurnost ne smije doći nauštrb osnovne funkcionalnosti), također nagle promjene u performansama mogu indikirati sigurnosni problem (npr. malware koji troši resurse, ili napadač koji eksfiltrira podatke generirajući neuobičajeni mrežni promet). U ovoj implementaciji je zbog toga integriran alat Prometheus za prikupljanje metrika, uz popratne komponente Node Exporter na svakom poslužitelju i Grafana za vizualizaciju tih metrika. Na svaki virtualni poslužitelj instaliran je Node Exporter, mali agent koji šalje osnovne metričke podatke o sustavu (iskorištenost procesora, zauzeće memorije, opterećenje diska, mrežni promet po sučelju, broj aktivnih procesa i dr.). Node Exporter standardno sluša na portu 9100 i odgovara na upite s trenutnim vrijednostima tih metrika. U upravljačkom segmentu pokrenut je Prometheus server, konfiguriran da povremeno (svakih nekoliko sekundi) prikuplja metrike sa svakog Node Exportera. To znači da je pfSense-ov vatrozid morao imati otvorena pravila koja dopuštaju promet iz Monitor segmenta prema ostalim segmentima na port 9100 (kako bi Prometheus mogao dohvatiti metrike sa svakog hosta). U Prometheus konfiguraciji Definiran je skup meta, IP adrese i portovi svih relevantnih instanci Node Exportera. Prometheus skladišti prikupljene vremenske serije metrika u svoju bazu podataka vremenskih serija, omogućujući postavljanje upita i definiciju alarma na temelju tih podataka. Za interpretaciju i preglednost tih podataka korištena je Grafana, popularno rješenje otvorenog koda za analizu metrika. Grafana je podešena da se poveže na Prometheus kao izvor podataka. Izrađeno je nekoliko nadzornih ploča (dashboards) koje prikazuju ključne parametre svake komponente. Na primjer, jedna Grafana ploča prikazuje status aplikacijskog poslužitelja: CPU i memorijsko opterećenje, broj zahtjeva, te mrežni promet na serveru sa aplikacijom. Druga ploča fokusira se na bazu podataka: opterećenje procesora na DB stroju, broj otvorenih konekcija na PostgreSQL i korištenje memorije. Treća nadzorna ploča agregira podatke bitne s aspekta sigurnosti, ukupni odlazni mrežni promet izvan interne mreže preko pfSense-a (što može upućivati na potencijalni napad ako raste neobjašnjivo), ili broj procesora i memorije koje koristi Keycloak (jer eventualni skok u korištenju resursa Keycloak-a mogao bi značiti da netko pokušava masovno pristupiti sustavu). Grafana omogućuje i postavljanje alarma. Primjerice, ukoliko iskorištenost CPU-a na bazi podataka kontinuirano prelazi 80% ili ako memorija Keycloak poslužitelja curi (*memory leak*) pa doseže kritične razine, sustav bi poslao obavijest administratorima. U arhitekturi nultog povjerenja, ova vrsta nadzora je značajna jer brza detekcija anomalija može spriječiti da sigurnosni problem preraste u ozbiljan incident.



Slika 8: Grafana nadzorna ploča

Slika 8 prikazuje jednu od Grafana nadzornih ploča s kombiniranim prikazom metrika, na slici se jasno mogu vidjeti granice između normalnog ponašanja i potencijalno abnormalnih aktivnosti. Spoj prikupljanja zapisa i metrika daje cjelovit uvid u sustav, zapisi pružaju diskretan pregled događaja i akcija, dok metričke vrijednosti otkrivaju trendove i opterećenja. Oba pristupa su predviđena modelom nultog povjerenja kao nužnost za održavanje visoke razine sigurnosti. Nije dovoljno samo postaviti politike i kontrole (segmentacija, autentifikacija) nego je potrebno i provjeravati u stvarnom vremenu poštuju li se te politike i postoje li indikacije pokušaja njihovog zaobilaženja. Kao što literature o arhitekturi nultog povjerenja naglašavaju, „nikad ne vjeruj“, to se odnosi i na vlastitu infrastrukturu. Uvijek se pretpostavlja mogući proboj te se sustav projektira tako da se kontinuirano nadgleda i sumnja u svaku aktivnost dok se ne potvrdi da je legitimna. Implementirani sustav logiranja i nadzora upravo omogućuje takav stupanj nepovjerenja i provjere, svaka autentifikacija, svaki mrežni paket i svaka promjena stanja sustava negdje su zabilježeni i mogu se analizirati.

## 5.2 Analiza i testiranje

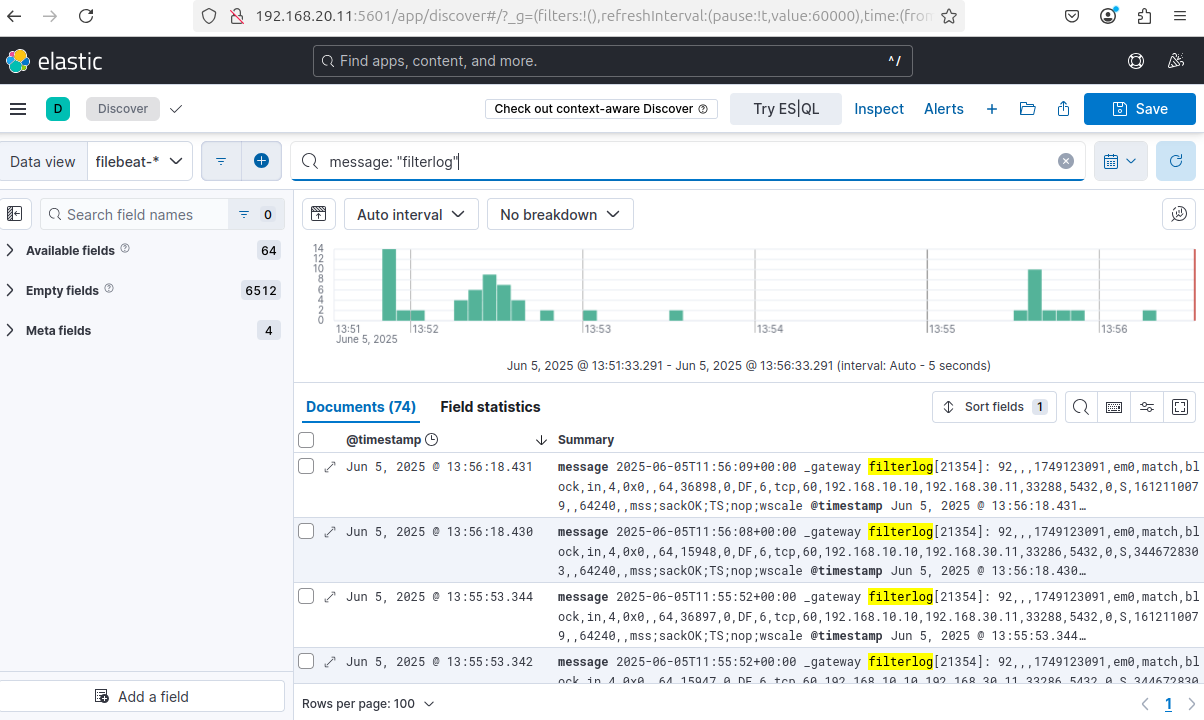
U ovom poglavlju opisuje se provedena analiza i testiranje sigurnosnih alata u simuliranom okruženju. Okruženje se sastoji od više međusobno povezanih komponenti koje zajedno utjelovljuju načela arhitekture nultog povjerenja: pfSense vatrozida za mikrosegmentaciju mreže, Flask aplikacije koja koristi Keycloak za autentifikaciju korisnika (OpenID Connect protokol), PostgreSQL baze podataka kao sigurnosno osjetljivog segmenta, sustava za prikupljanje i analizu zapisa (Filebeat i Elasticsearch) s Kibana sučeljem za vizualizaciju, te alata za nadzor performansi (Node Exporter i Prometheus) čiji se podaci mogu pregledavati u Grafani. U nastavku se izlažu testne metode, dobiveni rezultati te uočeni izazovi tijekom testiranja ovog okruženja.

### 5.2.1 Metodologija testiranja

Kako bi se provjerila učinkovitost implementiranih sigurnosnih mehanizama, osmišljeno je nekoliko testnih scenarija koji obuhvaćaju različite vrste potencijalnih napada i nepravilnosti. Prvo, testirana je mrežna mikrosegmentacija i otpornost vatrozida pfSense na neovlaštene pristupe. U tu svrhu simulirani su pokušaji pristupa zaštićenim resursima iz nepouzdanih segmenata mreže. Na primjer, pokušali smo ostvariti izravnu vezu prema PostgreSQL bazi podataka s računala koje ne pripada autoriziranom segmentu aplikacije. Korišteni su alati poput *nmap* (za skeniranje otvorenih portova) i *curl* (za ručno slanje HTTP zahtjeva) kako bi se simulirao napadač koji pokušava zaobići uobičajene puteve pristupa i pronaći eventualne propuste u mrežnim pravilima. Paralelno s time, provedena su testiranja autentifikacijskog sustava baziranog na Keycloak poslužitelju i Flask aplikaciji. Izvedeni su pokusi valjane i nevaljane prijave korisnika. U scenarijima valjane prijave korisnik unosi ispravne vjerodajnice te se preko Keycloaka uspješno autentificira i dobiva pristup zaštićenoj ruti Flask aplikacije. Nasuprot tome, u scenarijima nevaljane prijave namjerno su korištene pogrešne lozinke ili neispravni/istekli tokeni kako bi se provjerilo hoće li sustav odbiti pristup i pravilno evidentirati takve incidente. Također je simuliran niz uzastopnih neuspješnih prijava (npr. višestruko unošenje krivih zaporki) kako bismo vidjeli hoće li se aktivirati mehanizmi zaštite od grubih napada (poput privremene blokade korisničkog računa ili zahtjeva za dodatnom potvrdom, ako je podešeno). Uz testove sigurnosnih kontrola, praćene su i performanse sustava tijekom opterećenja. Generirano je kontrolirano opterećenje na ključnim komponentama – primjerice, slanjem velikog broja uzastopnih zahtjeva prema Flask aplikaciji simulirana je situacija DoS (*Denial of Service*) napada ili naglog priljeva korisnika. Cilj je bio uočiti hoće li doći do usporavanja ili rušenja servisa te kako će sustav za nadzor (Prometheus u kombinaciji s Grafanom) zabilježiti te promjene u opterećenju. Node Exporter je prikupljao metrike sustava (poput iskorištenosti CPU-a, memorije i mrežnog prometa) na svakom relevantnom čvoru, dok je Prometheus periodički prikupljao te podatke za analizu. Tako prikupljene metrike omogućile su praćenje stanja svakog dijela okruženja za vrijeme trajanja testova. Tijekom svih testnih scenarija, sustav za logiranje i nadzor bio je aktivno promatran. Filebeat agenti bili su konfigurirani da prikupljaju zapise iz više izvora: pfSense vatrozida (koji bilježi dopuštene i blokirane mrežne događaje), operativnog sustava i aplikacije (Flask/Keycloak), te baze podataka. Ti su zapisi u stvarnom vremenu slani u Elasticsearch, gdje su pohranjeni i spremni za analizu. Kroz Kibana sučelje pratili smo pojavljuju li se očekivani događaji (npr. neuspjela prijava ili blokirani mrežni zahtjev) te sadrže li zapisi dovoljno podataka za analizu. Istovremeno, putem Grafane su praćeni grafovi performansi kako bi se uočile korelacije između potencijalnih napada i opterećenja sustava.

### 5.2.2 Rezultati testiranja i analiza

Rezultati testova mrežne mikrosegmentacije pokazali su da pfSense vatrozid ispravno obavlja svoju ulogu u našem okruženju. Pokušaji neovlaštenog pristupa resursima izvan dopuštenog segmenta bili su blokirani prema definiranim pravilima. Primjerice, pri pokušaju izravnog povezivanja na PostgreSQL bazu podataka s neautorizirane adrese, pfSense je prekinuo taj promet i zabilježio ga kao blokiran događaj. Takav ishod potvrđen je pregledom zapisa, u Kibani se pojavio zapis iz pfSense sustava koji ukazuje na odbijen dolazni zahtjev prema određenom mrežnom portu baze podataka



Slika 9: Prikaz zapisa u Kibani

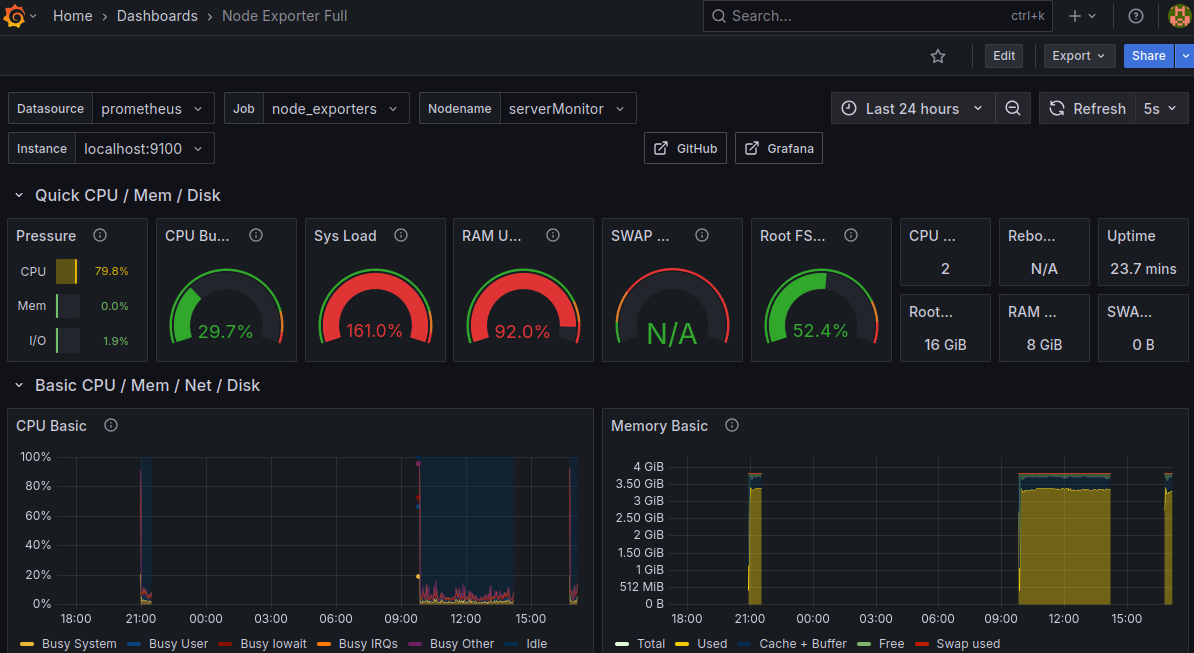
Slika 9 daje prikaz o pokušaju neovlaštenog pristupa segmentu baze podataka, ovi zapisi pokazuju da je mikrosegmentacija učinkovito spriječila lateralno kretanje napadača unutar mreže, svaki segment striktno je odvojen i komunikacija je dopuštena samo prema eksplicitno definiranim pravilima. Uz izravne pokušaje pristupa, port scanning test (nmap) dodatno je potvrdio robusnost mrežne segmentacije. Skeniranje portova iz nepouzdanog segmenta rezultiralo je tek ograničenim uvidom u okruženje, većina portova bila je nevidljiva ili označena kao filtrirana. PfSense je mnoštvo ovih skenirajućih pokušaja prepoznao i blokirao, što je također vidljivo u zapisima (više uzastopnih zapisa o blokiranom prometu s IP adrese napadača). Ovakva vidljivost pokušaja skeniranja važna je jer upućuje na to da sustav ne samo da uskraćuje neovlašteni pristup, nego i bilježi pokušaje prepoznavanja, čime administratori mogu biti svjesni da se njihova mreža našla na meti potencijalnog napada.

### 5.2.3 Autentifikacija i autorizacija korisnika

Testovi Keycloak autentifikacijskog sustava u suradnji s Flask aplikacijom pokazali su očekivano ponašanje u oba scenarija (uspješna i neuspješna prijava). Kod pokušaja prijave s valjanim vjerodajnicama, korisnik bi bio preusmjeren na Keycloak za unos korisničkog imena i lozinke, nakon čega bi, uz ispravan unos, Keycloak izdao OIDC token. Taj je token korišten u nastavku komunikacije te je Flask aplikacija dopuštala pristup traženim zaštićenim resursima. U zapisima se uspješna autentifikacija mogla pratiti kroz zapise Keycloaka (bilježi uspjeh prijave za danog korisnika i vrijeme prijave) kao i kroz zapise Flask poslužitelja koji bilježe pristup određenom resursu nakon autentifikacije. S druge strane, neuspješni pokušaji prijave ispravno su rezultirali odbijanjem pristupa. Keycloak je za krive vjerodajnice vraćao poruku o neuspjeloj prijavi, a Flask aplikacija nije dopuštala nastavak prema zaštićenim rutama bez valjanog tokena (odgovarala bi s HTTP statusom 401 (*Unauthorized*) za zahtjeve koji nemaju odgovarajuću autentifikaciju). Ključan dio ovih testova bila je provjera vidljivosti neuspjelih prijava. Uočeno je da svaki nevaljani pokušaj autentifikacije generira odgovarajući zapis, Keycloakov zapis evidentira neuspješnu prijavu (uključujući korisničko ime koje je pokušano, vrijeme i razlog neuspjeha), a ti su zapisi putem Filebeata proslijeđeni u Elasticsearch. U Kibani se tako mogu pretražiti svi incidenti neuspjelih prijava, što administratoru omogućuje detekciju potencijalnih napada na lozinke (npr. pokušaja *brute-force*). Tijekom testa višestrukih uzastopnih krivih lozinki, u zapisima su se jasno nizali zapisi o neuspjehu prijave za isti račun. Takvi ponavljajući događaji mogu poslužiti kao okidač za alarm ili daljnju istragu. Važno je napomenuti da je Keycloak nakon nekoliko uzastopnih neuspjeha aktivirao sigurnosne mjere (ovisno o konfiguraciji, to može biti privremena blokada korisnika ili dodatna provjera poput CAPTCHA izazova). Sustav je, dakle, ne samo bilježio neuspjehe, već je i aktivno reagirao ograničavanjem daljnjih pokušaja, čime se smanjuje rizik od automatiziranih napada pogađanja lozinki. Sve ove reakcije bile su vidljive kroz odgovarajuće zapise, što potvrđuje da je implementirana autentifikacijska komponenta u skladu s načelom „nikome ne vjeruj implicitno“.

### 5.2.4 Nadzor performansi i otpornost sustava

Tijekom testiranja opterećenja i potencijalnog rušenja servisa, sustav je pokazao određen stupanj otpornosti, uz istodobno pružanje bogatih podataka o performansama putem nadzornih alata. Kod simulacije DoS napada na Flask aplikaciju (slanje velikog broja zahtjeva u kratkom vremenu), aplikacija se nastavila izvršavati bez potpunog rušenja, ali su performanse očekivano degradirale – povećano vrijeme odziva i veće opterećenje CPU-a na poslužitelju aplikacije. Prometheus je zabilježio znatno povećanje iskorištenosti procesora na instanci Flask aplikacije, kao i porast memorijske potrošnje tijekom trajanja testa. Podatci su vizualizirani u Grafani, graf korištenja CPU-a jasno je pokazao skok na gotovo 80% opterećenja u periodima najintenzivnijeg napada, dok je memorija rasla do granica dostupnih resursa (Slika 10).

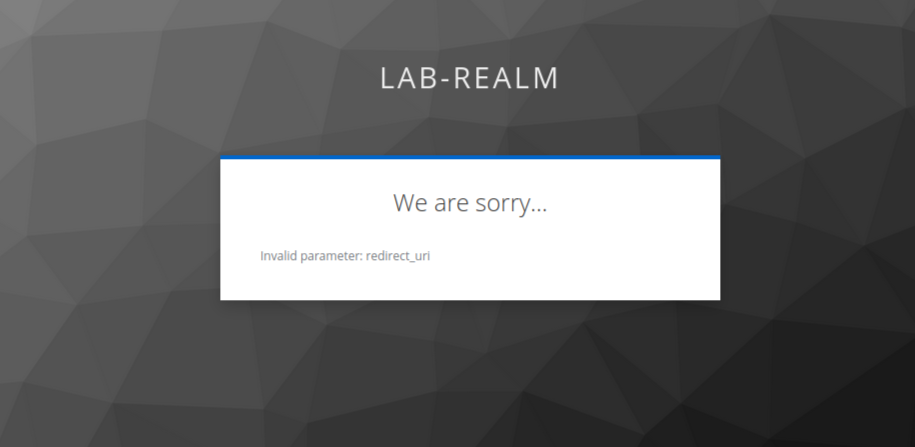


Slika 10: Prikaz opterećenja CPU-a poslužitelja aplikacije tijekom simuliranog napada

Pozitivno je što niti jedna komponenta nije neočekivano otkazala, sustav je ostao funkcionalan, čime je demonstrirana otpornost na barem kratkotrajne napade tog intenziteta. Nadzorni sustav pokazao se izuzetno korisnim u analizi ponašanja pod opterećenjem. Grafana je omogućila korelaciju vremena napada s konkretnim metričkim vrijednostima. Primjerice, moguće je precizno odrediti u kojem trenutku počinje rast opterećenja i koliko traje. Na temelju uočenih rezultata, mogu se definirati pragovi (npr. ako procesor koristi više od 90% dulje od 5 minuta ili ako broj neuspjelih prijava prijeđe određeni broj u kratkom vremenu) koji bi pokrenuli alarm i obavijestili administratore o sumnjivim aktivnostima ili potencijalnim problemima sa performansama. Ova integracija sigurnosnih alata i nadzornih mehanizama u cjelini potvrđuje važnu stavku pristupa nultog povjerenja: kontinuirano praćenje stanja sustava i brza detekcija anomalija jednako su bitni kao i preventivne mjere zaštite.

### 5.2.5 Izazovi i naučene lekcije

Tijekom implementacije i testiranja opisanog okruženja nultog povjerenja pojavili su se određeni izazovi i problemi koji su zahtijevali prilagodbe konfiguracije i poslužili kao vrijedne lekcije. Jedan od prvih izazova pojavio se kod integracije Flask aplikacije s Keycloak poslužiteljem za autentifikaciju, konkretno oko parametra preusmjeravanja URI u OIDC protokolu. U početnoj konfiguraciji, pokušaji autentifikacije bi rezultirali pogreškom jer se Keycloak nakon uspješne prijave korisnika nije mogao ispravno vratiti na Flask aplikaciju, prijavljivalo se da URI preusmjeravanja nije valjan. Ova je pogreška ukazivala na to da adresa na koju aplikacija očekuje povratak (nakon što korisnik unese vjerodajnice na Keycloak sučelju) nije bila ispravno registrirana u postavkama Keycloak klijenta. Rješenje je pronađeno ažuriranjem konfiguracije Keycloaka, dodana je točna URL putanja Flask aplikacije kao dozvoljeni redirect URI. Nakon ove izmjene, autentifikacijski tok počeo je teći neometano. Ovaj slučaj naglašava kako se i mala konfiguracijska pogreška može očitovati kao ozbiljan kvar sustava.



Slika 11: primjer poruke o pogrešci povezane s nevažećim redirect URI parametrom

Slika 11 je zabilježena tijekom testiranja prije ispravka konfiguracije. Takvi problemi istaknuli su važnost pažljive konfiguracije svih komponenti u distribuiranom okruženju, svaka karika (u ovom slučaju, Keycloak i aplikacija) mora biti usklađena kako bi sustav u cjelini funkcionirao. Daljnji izazovi bili su vezani uz fino podešavanje mrežnih pravila i logiranja. Prilikom inicijalne primjene mikrosegmentacije preko pfSense vatrozida, ustanovili smo da pretjerano restriktivna pravila mogu omesti legitimnu komunikaciju među komponentama. Primjerice, u ranoj fazi testiranja jedan je od servisa (Flask aplikacija) gubio vezu prema bazi podataka jer pravilo vatrozida nije isprva bilo konfigurirano da dozvoli tu legitimnu komunikaciju na odgovarajućem portu. Trebalo je pomno proanalizirati pfSense pravila i postupno ih prilagodit, balansirajući između principa najmanjih privilegija i funkcionalnih potreba sustava. Ova iteracija naučila nas je da je mikrosegmentacija djelotvorna samo ako se pravilno implementira; potrebno je detaljno razumjeti komunikacijske zahtjeve aplikacija kako bi se izbjeglo nenamjerno uskraćivanje dopuštenih aktivnosti. Istovremeno, bilo kakva preširoka pravila odmah su sužena čim bi se ustanovilo da nisu nužna, kako bi se smanjio potencijalni napadački prolaz. Tijekom ovog procesa, zapisi pfSense vatrozida pokazali su se vrlo korisnima: svako odbijanje legitimnog prometa evidentirano je, što je ubrzalo dijagnostiku (administrator može u Kibani odmah vidjeti je li određeni paket odbačen vatrozidom i zašto). Što se tiče zapisa i vidljivosti događaja, početna konfiguracija sustava za prikupljanje zapisa zahtijevala je doradu. U početku su neki zapisi pristizali u Elasticsearch neobrađeni ili nisu sadržavali dovoljno strukturirane podatke za laku analizu. Primjerice, pfSense generira opširan tekstualni zapis za svaki događaj; isprva su ti zapisi u Kibani bili prikazani kao nepregledan niz teksta, otežavajući filtriranje po IP adresi, portu, akciji ili sl. Uvođenjem Filebeat modula za pfSense (odnosno prilagođenog obrasca za obradu tih zapisa) postigli smo da se zapisi automatizirano raščlanjuju na polja (npr. source.ip). Time je pretraga i vizualizacija postala daleko učinkovitija, mogli smo, primjerice, jednostavno filtrirati sve blokirane događaje u određenom vremenskom rasponu ili iz određene mrežne zone. Slično tome, prilagođeni su i zapisi aplikacije i Keycloaka gdje se pazilo da ključni događaji (poput uspješnih i neuspješnih prijava, grešaka u radu servisa, upozorenja o resursima) bilježe s odgovarajućom razinom detalja. Morali smo pronaći ravnotežu u razini detaljnosti zapisa, premalo informacija otežava analizu, dok previše (zapisivanje svega) otežava filtriranje i može nepotrebno opteretiti sustav za prikupljanje podataka. Nakon podešavanja, sustav logiranja pružao je jasnu sliku sigurnosnih događaja i stanja sustava, što je bilo ključno za pouzdanu analizu. Naposljetku, identificirana su i područja za buduća poboljšanja koja u vrijeme inicijalnog testiranja nisu bila u potpunosti implementirana, ali su planirana radi podizanja sigurnosne razine. Prvo, uvođenje dvofaktorske autentifikacije (2FA) preko Keycloaka predstavljat će značajan dodatni sloj zaštite. Tijekom testiranja razmatran je scenarije u kojema bi napadač mogao kompromitirati lozinku korisnika; s uključenim 2FA, i takav kompromitirani račun zahtijevao bi dodatni jednokratni kod (npr. putem mobilne aplikacije) čime bi napad vjerojatno bio osujećen. U budućim testovima, nakon implementacije 2FA, valjalo bi provjeriti ponašanje sustava u slučajevima pokušaja prijave bez drugog faktora ili s pogrešnim drugim faktorom, kao i utjecaj na korisničko iskustvo i zapise (očekujemo, primjerice, da će neuspjeh drugog faktora također biti jasno evidentiran u Keycloak zapisima). Sažeto, provedenim testiranjem potvrđena je visoka razina sigurnosti i preglednosti našeg okruženja nultog povjerenja, uz istovremeno otkrivanje korisnih spoznaja za njegovo unaprjeđenje. PfSense vatrozid uspješno je izolirao sustav u više segmenata i onemogućio neovlašteno kretanje napada unutar mreže, što je potkrijepljeno zapisima događaja (Slika 9). Autentifikacijski sloj (Flask + Keycloak) osigurao je da samo legitimni korisnici s ispravnim vjerodajnicama i tokenima mogu pristupiti resursima, dok su svi pokušaji zaobilaženja tog mehanizma propali i pritom ostavili traga u zapisima. Nadzorni alati (Elasticsearch za zapise te Prometheus za metrike) pokazali su se neprocjenjivima u razumijevanju stanja sustava te svaki relevantan sigurnosni događaj ili anomalija u opterećenju sustava bio je vidljiv i analizabilan u realnom vremenu. Izazovi na koje smo naišli, pružili su nam dublji uvid u međusobnu ovisnost komponenti u arhitekturi nultog povjerenja. Svaka poteškoća uspješno je prevladana prilagodbom postavki i ponovnim testiranjem, čime je sustav postupno doveden na višu razinu zrelosti. Time je demonstrirano da je iterativni proces testiranja i podešavanja ključan za uspješnu implementaciju sigurnosnog modela. Kontinuirano poboljšavanje na temelju konkretnih saznanja osigurava da sigurnosni alati djeluju skladno i učinkovito u stvarnim scenarijima napada i opterećenja.

# 6. Zaključak

U praktičnom dijelu rada, principi nultog povjerenja primijenjeni su u laboratorijskom virtualnom okruženju. Evaluacija rezultata pokazuje brojne sigurnosne prednosti primjene modela. Ograničavanjem pristupa na osnovi najmanjih privilegija znatno se smanjuje površina napada, korisnici i uređaji mogu pristupiti samo onim resursima koji su im nužni za posao. Kontinuirana provjera identiteta i snažni mehanizmi autentikacije onemogućuju neprimjeren pristup čak i ako je jedan dio sustava kompromitiran. Primjena mrežne segmentacije dodatno izolira segmente tako da napadač, u slučaju probijanja jedne zone, ne može lako prijeći u drugu. U laboratorijskim testovima ta je kombinacija mjera rezultirala vidljivo smanjenim lateralnim kretanjem i bržom detekcijom sumnjivih aktivnosti. Osim toga, implementacija arhitekture nultog povjerenja donosi bolju vidljivost u rad mreže: kontinuiranim praćenjem i zapisivanjem događaja dobili smo bogat skup podataka za analizu i brzi odgovor na prijetnje.

Stručna literatura upozorava da potpuna implementacija zahtijeva detaljno mapiranje svih resursa i tokova unutar mreže, što je izrazito kompleksan i dugotrajan proces. Uz to, početno ulaganje u nadogradnju infrastrukture te edukaciju osoblja može biti vrlo skupo. Naš praktični rad potvrdio je ove nalaze. Iako model značajno pojačava sigurnosnu zaštitu, stvarna implementacija može biti zahtjevna i resursno intenzivna te zahtijeva visoku razinu stručnosti i opsežno planiranje. Ovaj diplomski rad doprinio je razumijevanju modela nultog povjerenja u praktičnom kontekstu demonstrirajući njegovu primjenu u kontroliranom eksperimentalnom okruženju.

Automatizacijom postupaka i podešavanjem pravila, smanjilo bi se vrijeme odgovora sigurnosnih sustava i opterećenje stručnjaka za sigurnost. Također bi se trebalo uvesti mjerenje ključnih performansi sustava, kao što je praćenje latencije provjere pristupa ili iskorištenosti procesorskih i mrežnih resursa, kako bi se procijenio utjecaj mehanizama na brzinu i stabilnost mreže. Primjena naprednih autorizacijskih metoda mogla bi još više poboljšati sigurnost i prilagoditi se kontekstu korisnika. Konačno, preporučuje se proširiti model na okruženja u oblaku i moderne hibridne scenarije. Integracija sustava sa servisima u oblaku ili korištenje mrežnih rješenja (ZTNA) omogućila bi siguran pristup korporativnim aplikacijama i podacima u oblaku, čime bi se dodatno učvrstila sigurnosna pozicija organizacije. U cjelini gledano, zaključak rada jest da arhitektura nultog povjerenja nudi značajan iskorak u odnosu na klasične sigurnosne modele, ali da je njezina potpuna implementacija izazovna. Napori i resursi uloženi u uspostavu opravdani su povećanjem razine zaštite i boljom spremnošću na suvremene prijetnje, ali zahtijevaju sustavan pristup i kontinuiranu evaluaciju. Naš doprinos ovoj temi sastoji se u praktičnoj demonstraciji principa nultog povjerenja i analizi izazova u eksperimentalnom okruženju, čime smo proširili znanje o tome kako se koncept nultog povjerenja može ostvariti u praksi. Daljnji rad na automatizaciji, praćenju performansi i integraciji s tehnologijama u oblaku dodatno bi unaprijedio učinkovitost i primjenjivost arhitekture.

# 7. Literatura

[1] Kang H, Liu G, Wang Q, Meng L, Liu J, „Theory and Application of Zero Trust Security: A Brief Survey“, PMC, 2023.

[2] Perception Point, „Zero Trust – What is it and what development & implementation trends are there for 2023“, s Interneta, <https://perception-point.io/>, 15. travnja, 2025.

[3] AZTech IT Solutions, „Zero Trust vs Traditional Security: What's the Difference?“, s Interneta, <https://www.aztechit.co.uk/blog/zero-trust-vs-traditional-perimeter-security>, 20. travnja, 2025.

[4] Majority staff report, „A Kill Chain Analysis of the 2013 Target Data Breach,“ 2014.

[5] Shaharyar Khan, Ilya Kabanov, Yunke Hua, Stuart Madnick, „Systematic Analysis of the Capital One Data Breach: Critical Lessons Learned“, 2022

[6] Rose S, Borchert O, Mitchell S, Connelly S, „Zero Trust Architecture,” NIST Special Publication 800-207, National Institute of Standards and Technology, 2020.

[7] J. Kindervag, „No More Chewy Centers: Introducing The Zero Trust Model of Information Security,” Forrester Research Inc., 2010.

[8] Microsoft, „What is Zero Trust?,”, s Interneta, https://www.microsoft.com/en-us/security/business/zero-trust, 15. svibnja, 2025.

[9] R. Ward, B. Beyer, „BeyondCorp: A New Approach to Enterprise Security”, vol. 39, no. 6, str. 6–11, 2014.

[10] CarPen Rebuild, „Zero Trust Security: Što je i zašto je važno?,“ s Interneta, <https://carpen-rebuild.hr/zero-trust-security-sto-je-i-zasto-je-vazno/> , 20. svibnja, 2025.

[11] J. Ardalić, Sigurnost IT sustava smještenih u višeoblačnom okruženju, Završni specijalistički rad, Univ. u Zagrebu, FER, 2023.

[12] V. Dakić, Z. Morić, A. Kapulica, D. Regvart, „Analysis of Azure Zero Trust Architecture Implementation for Mid-Size Organizations,“ Journal of Cybersecurity and Privacy, vol. 5, no. 1, 2024.

[13] S. Sarkar, G. Choudhary, S. K. Shandilya, A. Hussain, H. Kim, „Security of Zero Trust Networks in Cloud Computing: A Comparative Review“, Sustainability, vol. 14, no. 18, 2022.

[14] D. Tyler, T. Viana, „Trust No One? A Framework for Assisting Healthcare Organisations in Transitioning to a Zero-Trust Network Architecture“, Applied Sciences, vol. 11, no. 16, 2021.

[15] D. Klein, „Micro-Segmentation: Securing Complex Cloud Environments“, Network Security, vol. 2019, no. 8, 6–10, 2019.

[16] A. Hellström Ryckert, „Zero Trust in Private Home Networks : A Sociotechnical Perspective on Implementing Zero Trust to Enhance Home Network Security“, 2025.

1. ELK platforma je skup alata (Elasticsearch, Logstash i Kibana) za prikupljanje, obradu i vizualizaciju podataka u stvarnom vremenu. [↑](#footnote-ref-1)
2. LDAP servis je protokol za pristup i upravljanje informacijama o korisnicima i resursima unutar mreže. [↑](#footnote-ref-2)
3. Oblik prijevare u kojem napadač lažno predstavlja pouzdanu osobu ili instituciju kako bi naveo korisnika da otkrije osjetljive podatke poput lozinki, brojeva kartica ili osobnih podataka. [↑](#footnote-ref-3)
4. Pravilo koje omogućuje zaposlenicima da na poslu koriste vlastite uređaje, poput mobitela ili prijenosnih računala, uz određena sigurnosna pravila. [↑](#footnote-ref-4)
5. National Institute of Standards and Technology [↑](#footnote-ref-5)
6. Sustavi rješenja za prikupljanje, normalizaciju i automatiziranu analizu sigurnosnih događaja i zapisa. [↑](#footnote-ref-6)
7. Rješenja za upravljanje uređajima, aplikacijama, sadržajem i sigurnosnim politikama. [↑](#footnote-ref-7)
8. Zakon koji štiti privatnost i osobne podatke građana EU, te određuje kako organizacije smiju prikupljati, pohranjivati i koristiti te podatke. [↑](#footnote-ref-8)
9. Tuneliranje je sigurna metoda koja omogućava korisnicima pristup aplikacijama preko enkriptiranog tunela, osiguravajući provjeru identiteta i minimalan pristup. [↑](#footnote-ref-9)
10. Platforma za mrežnu virtualizaciju koja omogućava kreiranje, upravljanje i automatizaciju virtualnih mreža, nezavisno od fizičke mrežne infrastrukture. [↑](#footnote-ref-10)
11. Softverski definirane mrežne kartice koje omogućuju virtualnim mašinama povezivanje na mrežu, kao da imaju fizičku mrežnu karticu. [↑](#footnote-ref-11)
12. Lista pravila koja određuje tko i na koji način može pristupiti mrežnim resursima, uređajima ili podacima. [↑](#footnote-ref-12)
13. Otvorena softverska biblioteka koja omogućava sigurno komuniciranje preko mreže pomoću enkripcije, posebno za implementaciju protokola kao što su SSL i TLS. [↑](#footnote-ref-13)