Sustav za pronalazak sigurnosnih propusta Analyzeme.dev

Mustafi, Senad

Sveučilište Jurja Dobrile u Puli

Fakultet Informatike u Puli

Senad Mustafi

Sustav za pronalazak sigurnosnih propusta Analyzeme.dev

Završni rad

Pula, rujan 2024.

Sveučilište Jurja Dobrile u Puli

Fakultet Informatike u Puli

Senad Mustafi

Sustav za pronalazak sigurnosnih propusta Analyzeme.dev

Završni rad

JMBAG: 0303100029

Studijski smjer: Informatika

Znanstveno područje: Društvene znanosti

Znanstveno polje: Informacijske i komunikacijske znanosti

Znanstvena grana: Informacijski sustavi i informatologija

Kolegiji: Web aplikacije

Mentor: doc. dr. sc. Nikola Tanković

Pula, rujan 2024.

SADRŽAJ

SAŽETAK

Analyzeme.dev je platforma koja omogućava entuzijastima iz područja penetracijskog ispitivanja i kibernetičke sigurnosti da uštede vrijeme koristeći integrirane alate za testiranje. Platforma nudi sedam alata, uključujući alate za ispitivanje mrežnih sustava, skeniranje ranjivosti web aplikacija, identifikaciju poddomena, tehnologija internet stranica, DNS analizu i skeniranje otvorenih portova. Platforma je napisana u JavaScript jeziku. Klijentski sloj razvijen je u Vue.js okviru, dok je poslužiteljski sloj razvijen korištenjem Express.js okvira. Cijeli sustav je hostiran u oblaku, omogućujući veću fleksibilnost i sigurnost. U ovom radu detaljno je opisano kako je platforma izgrađena, logika iza svakog od alata, te način na koji su oni integrirani i automatizirani. Također, objašnjen je proces implementacije u oblaku i prednosti koje to pruža za korisnike u smislu dostupnosti i performansi.

Ključne riječi: Penetracijsko testiranje, express.js, vue.js, alat, api

ABSTRACT

Analyzeme.dev is a platform that enables penetration testing and cybersecurity enthusiasts to save time by using integrated testing tools. The platform offers seven tools, including tools for testing network systems, scanning web application vulnerabilities, identifying subdomains, website technology identification, DNS analysis, and open port scanning. The platform is written in JavaScript. The client layer (Front end) is developed using the Vue.js framework, while the server layer (Back end) is developed using Express.js framework. The entire system is hosted in the cloud, providing greater flexibility, and security. This paper provides a detailed description of how the platform was built, the logic behind each tool, and how they are integrated and automated. It also explains the cloud implementation process and the benefits it offers to users in terms of availability and performance.

Keywords: Penetration testing, express.js, vue.js, tool, api

1. Uvod

U današnjem digitalnom svetu, privatnost i zaštita informacija postaju sve značajniji i važniji. Sve vrste organizacija su, u manjoj ili većoj meri, digitalizovale deo svog poslovanja. Zbog toga je neophodna zaštita poverljivih informacija i informatičke infrastrukture. Penetracijsko testiranje jedna je od najvažnijih stavki u identifikaciji ovih pretnji. Kako Georgia Weidman objašnjava u svojoj knjizi "Penetration Testing: A Hands-On Introduction to Hacking", „Penetracijsko testiranje je ključno za otkrivanje i ublažavanje sigurnosnih ranjivosti pre nego što ih zlonamerni akteri mogu iskoristiti“ (Weidman, 2014). Ovo testiranje omogućava otkrivanje potencijalnih ranjivosti pre nego što ih napadači zloupotrebe.

Analyzeme.dev je online platforma koja je osmišljena s ciljem da automatizira i olakša postupak i proces penetracijskog ispitivanja. Korištenjem *Analyzeme.dev* korisnici mogu uštedjeti dobar dio svog vremena zahvaljujući automatizaciji koju platforma pruža. Platforma *Analyzeme.dev* nudi sedam penetracijskih alata na jednome mjestu, pružajući sveobuhvatan pristup sigurnosti mrežnih aplikacija i infrastrukture. Ključne funkcionalnosti *Analyzeme.dev* uključuju: *Web Status and Vulnerabilities Scanner* koji se odnosi na automatsko skeniranje mrežnih stranica radi identifikacije potencijalnih ranjivosti i statusa servera. Zatim, *WordPress Admin Usernames Scanner* koji se koristi za pronalaženje korisničkih imena administrativnih naloga na *WordPress* stranicama kako bi se otkrile potencijalne mete za napade. Nadalje, sljedeća važna funkcija ove platforme odnosi se na alat *Web Technology Identifier* koji služi za identifikaciju tehnologija koje su korištene za izradu i održavanje mrežnih stranica. Prema Stuttard i Pinto, „Razumevanje tehnologija koje se koriste na web stranici, kao i analiza njenih različitih komponenti, omogućava testiranju bezbednosti da se efikasnije identifikuju potencijalne ranjivosti i napadačke tačke“ (Stuttard & Pinto, 2011). Ovaj pristup je ključan za sveobuhvatan pregled sigurnosnih rizika i implementaciju odgovarajućih zaštitnih mera.

Alat kojim se skenira direktorijum kako bi se otkrile neželjene liste podataka kojim se mogu otkriti osjetljive informacije naziva se *Directory Listing Scanner. Subdomain Finder* alat je za pronalaženje poddomene povezanih sa glavnom domenom, otkrivajući dodatne

točke napada. Preostaju još dva važna alata za analiziranje mreža. Jedan od njih je DNS Lookup, čija se funkcija odnosi na detaljnu analizu DNS zapisa za identifikaciju potencijalnih problema i slabosti. Skeniranje otvorenih mrežnih ulaza na serverima radi identifikacije usluga koje mogu biti ranjive na napade, služi nam sljedeći alat naziva Port Scan. DNS analiza i skeniranje portova su ključni alati za otkrivanje mrežnih ranjivosti i procenu sigurnosnih rizika na mrežnoj infrastrukturi (Muniz, 2018).

Platforma *Analyzeme.dev* je osmišljena da entuzijastima u području kibernetičke sigurnosti omogući efikasno i brzo sprovođenje penetracijskog ispitivanja. Ova platforma, osim same efikasnosti, pruža i brzinu pentestiranja.

2. Tehnološki aspekti izrade *Analyzeme.dev*

Za izradu internet alata *Analyzeme.dev* korištene su suvremene tehnologije koje omogućavaju, efikasan i skalabilan sistem. Dakle, tijekom izrade prethodno navedene platforme valja naglasiti da je kao glavni programski jezik korišten JavaScript što je doprinijelo jednostavnijoj komunikaciji između klijentske i serverske strane te lakšu razmjenu podataka i potencijalnom smanjenju složenosti razvoja. Za praćenje promjena koda i upravljanje verzijama projekta, korišten je *GitHub*.To je popularna platforma za razvoj softvera koja omogućuje programerima da surađuju, dijele i pregledavaju izmjene u kodu. Sve promjene ovog projekta "*Analyzeme.dev*" nalaze se pohranjene na GitHub repozitoriju. Poveznica poslužiteljskog sloja do izvornog koda je:

https://github.com/senadmustafi/AnalyzeMe

Poveznica klijentskog sloja do izvornog koda je:

https://github.com/senadmustafi/analyzeme-fe

2.1. Poslužiteljski sloj

Za poslužiteljski sloj aplikacije korišćen je Express[1], mrežni okvir za Node.js[2], koji omogućava brzu i jednostavnu izradu API-ja i upravljanje serverima. Express pruža fleksibilnu i laganu strukturu koja je idealna za razvoj aplikacija kod kojih je potrebna visoka performansa. M. E. Brade u knjizi "Express.js Guide: The Comprehensive Book on Express.js" ističe da je "Express.js dizajniran da pruži jednostavan i efikasan način za izgradnju API-ja i upravljanje serverima, omogućavajući programerima da brzo razvijaju visoko performantne aplikacije" (Mardana, 2014)

2.2 Klijentski sloj

Za klijentski sloj korišćen je Vue.js[3], progresivni JavaScript okvir koji je poznat po svojoj jednostavnosti i mogućnosti za gradnju interaktivnih korisničkih sučelja. Vue.jomogućava

[1] Express.js je popularan web okvir za Node.js koji pojednostavljuje izgradnju i upravljanje web aplikacijama. URL: https://expressjs.com/ (17.08.2024)

[2] Node.js je višenamjensko okruženje za izvođenje JavaScript koda na serverskoj strani. URL: https://nodejs.org/ (15.08.2024)

[3] Vue.js je progresivni JavaScript okvir za izgradnju korisničkih sučelja i jedno-straničnih aplikacija.

jednostavnu izradu interaktivnih korisničkih sučelja kroz komponente koje olakšavaju razvoj i održavanje web aplikacija (Macrae, 2018).

2.3 Baza podataka

Za skladištenje podataka korišćena je MongoDB[4], NoSQL baza podataka koja omogućava fleksibilno i skalabilno upravljanje podacima. MongoDB je baza koja je smeštena u oblaku, što osigurava visoku dostupnost i lakoću upravljanja podacima, posebno za aplikacije koje zahtevaju dinamičko skaliranje i upravljanje velikim količinama podataka. Prema Kristini Chodorow u knjizi "MongoDB: The Definitive Guide," MongoDB je dizajnirana da pruži visoku skalabilnost i fleksibilnost za rad sa velikim količinama podataka, omogućavajući efikasno upravljanje i pristup podacima u realnom vremenu (Chodorow, 2013).

URL https://vuejs.org/ (21.08.2024)

[4] MongoDB je program klasificiran kao NoSQL proizvod baze podataka. URL: https://www.mongodb.com/ (21.08.2024)

2.4 Biblioteke

Biblioteke u programiranju odnose se na zbirke unaprijed napisanih funkcija, klasa, ili modula koje omogućavaju programerima da koriste već gotove kodove za obavljanje određenih zadataka. One omogućuju lakše i brže pisanje koda, odnosno ne piše se kod od početka već programeri unose biblioteke i koriste njihove funkcionalnosti te time štede vrijeme i smanjuju mogućnost pogrešaka čime se ubrzava proces izrade kompjuterskih programa.

Biblioteka Bcrypt koristi se za sigurno šifriranje korisničkih lozinki. Ova biblioteka omogućava preobrazbu lozinki na siguran način, čime se osigurava zaštita korisničkih podataka. Bcrypt koristi prilagodljivu funkciju derivacije ključa kako bi osigurao da čak i slabije lozinke postanu otpornije na napade poput *brute-force* napada, što je ključno za zaštitu osjetljivih podataka (Stallings, 2017).

Biblioteka korištena za skeniranje otvorenih računalnih ulaza na serverima je *NodePortScanner*. Ovaj alat omogućava identifikaciju usluga koje su dostupne na određenim računalnim ulazima, što je ključno za procjenu sigurnosti servera. *Wappalyzer* je alat korišten za identifikaciju tehnologija koje su korištene za izradu i održavanje mrežnih stranica. *Wappalyzer* omogućava analizu mrežnih tehnologija, što pomaže u boljem razumijevanju infrastrukture i potencijalnih ranjivosti ciljanih aplikacija.

PicoCSS[5] je minimalistički CSS okvir dizajniran za jednostavnost i lakoću upotrebe. Omogućava brzo stiliziranje mrežnih stranica bez potrebe za pisanjem opsežnog CSS koda. Korištenje minimalističkih okvira kao što je PicoCSS može značajno ubrzati razvoj i stiliziranje web stranica, pružajući elegantna rješenja za uobičajene dizajnerske izazove (Verou, 2015).

[5] Minimalistički i lagani početni kit koji prioritet daje semantičkoj sintaksi, čineći svaki HTML element responzivnim i elegantnim prema zadanim postavkama. URL: https://picocss.com (25.08.2024)

2.5 Funkcionalnosti *Analyzeme.dev*

Platforma *Analyzeme.dev* je napravljena da entuzijastima pruži alate na jednome mjestu za efikasno i brzo pentestiranje. Automatizacija ključnih procesa pentestiranja omogućava brže i točnije rezultate.

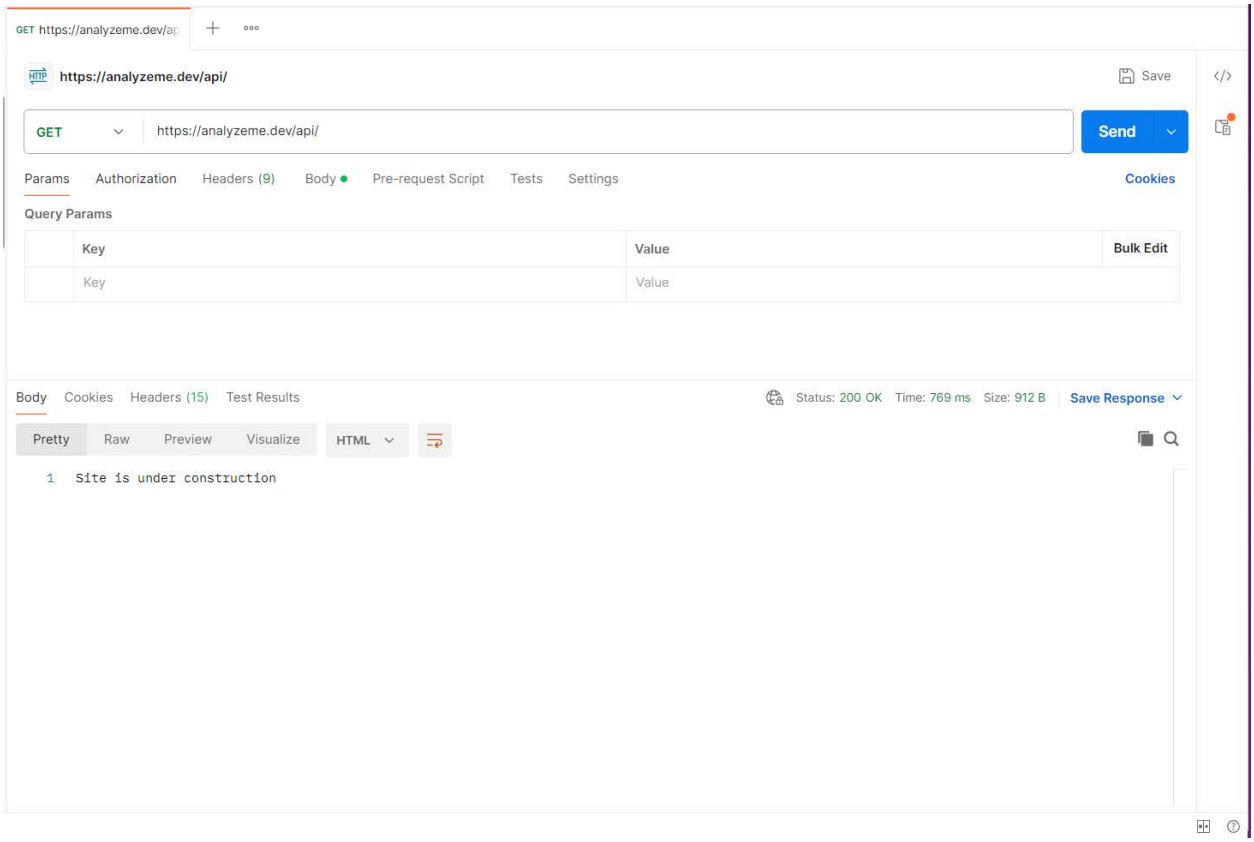
Jedna od glavnih funkcija, a samim time i prednosti ove platforme jest pravovremeno otkrivanje pa posljedično tome i otklanjanje propusta vezanih za mrežnu sigurnost i kompjuterskih programa. Iako ovakva vrsta alata već postoji, platforma *Anlyzeme.dev* nudi dobar dio alata na jednome mjestu. Ono što automatizacija, u ovome kontekstu, podrazumijeva jest prepoznavanje ranjivosti, skeniranje mreža, analizu aplikacija i provjeru sigurnosnih postavki i sve to dostupno na jednome mjestu. Platforma je dizajnirana tako da je pogodna i za početnike i za iskusne stručnjake, jer kombinira jednostavno sučelje s naprednim funkcionalnostima. Uz *Analyzeme.dev*, penetracijsko testiranje postaje brže, lakše i točnije, omogućujući korisnicima da identificiraju i riješe sigurnosne probleme prije nego što postanu ozbiljan rizik.

3. Temeljni proces izrade alata *Analyzeme.dev*

Izrada platforme započeta je temeljnim radom na poslužiteljskom sloju, koristeći *Express* biblioteku i *Node.js* mrežni okvir. Ovaj okvir je odabran zbog svoje fleksibilnosti i visokih performansi, što je vrlo važno za platformu koja će obrađivati velike količine zahtjeva u kratkom vremenu.

Prvi korak bio je implementacija sustava za registraciju i prijavu novih korisnika. Ovaj sustav omogućava korisnicima kreiranje računa (registraciju) i prijavu u sustav (logiranje). Pouzdanost je ključna funkcija koja osigurava da samo ovlašteni korisnici imaju pristup aplikaciji. U ovom koraku, korištena je *bcrypt* biblioteka za sigurnu preobrazbu lozinki, čime se osigurava zaštita korisničkih podataka u bazi.

Tijekom razvoja poslužiteljskog sloja, Postman[6] je korišten kao alat za testiranje API-ja. Postman je omogućio simulaciju različitih zahtjeva prema serveru, kao što su registracija, prijava i rukovanje greškama, što je značajno ubrzalo proces razvoja i osiguralo da su sve funkcije ispravno implementirane. Postman omogućava jednostavno testiranje RESTful API-ja simuliranjem stvarnih zahtjeva, što ubrzava proces otklanjanja grešaka i poboljšava performanse aplikacije (Westerveld, 2021).



Slika 1. Sučelje alata za testiranje “Postman“

[6] Postman je popularna platforma za razvoj API-ja koja omogućava korisnicima da kreiraju, testiraju, dokumentiraju i dijele API-je. URL: https://www.postman.com (17.08.2024)

```
1    async authenticateUser(email, password){
2        let db = await connection();
3        let user = await db.collection("users").findOne({email: email});
4
5        if(user && user.password && (await bcrypt.compare(password, user.password))){
6            delete user.password;
7            let token = jwt.sign(user, process.env.JWT_SEC, {algorithm: "HS512", expiresIn: "1 week"});
8            return{token, email:user.email};
9
10       }else{
11           throw new Error("Cannot auth");
12       };
13   },
```

Programski kod 1. Funkcija za registraciju korisnika

```
1  app.post('/users', async (req, res) => {
2    let user = req.body;
3
4    let id;
5    try {
6      id = await auth.registerUser(user);
7    }
8    catch (e) {
9      res.status(500).json({ error: e.message })
10   }
11
12   res.json({ id });
13
14 })
```

Programski kod 2. Krajnja točka (Endopoint) za registraciju korisnika

3.1 Razvoj specifičnih alata unutar aplikacije

Nakon postavljanja osnovnih funkcija računalne sigurnost, prešlo se na razvoj specifičnih alata koji čine suštinu *Analyzeme.dev* aplikacije. Prvi alat razvijen u ovoj fazi bio je *WordPress Admin Usernames Scanner*. Ovaj alat je dizajniran za otkrivanje korisničkih imena administrativnih korisnika na *WordPress*[7] mrežnim stranicama, što može biti ključna informacija za procjenu sigurnosnih prijetnji. Alat funkcionira tako da šalje zahtjev na */wp-json/wp/v2/users* endpoint na *WordPress* stranicama i prikazuje listu korisnika koji su povezani s tim stranicama, omogućujući korisnicima identifikaciju potencijalnih meta za napade. *WordPress* REST API pruža informacije o korisnicima putem javno dostupnih endpointa, što može predstavljati sigurnosni rizik ako nije pravilno konfiguriran (Król, 2019).

```
1  //Scan wordPress admins
2  app.post('/scan-wp-users', async (req, res) => {
3    try{
4    let domain = req.body;
5    const ourdata = await axios.get(domain.domain + "/wp-json/wp/v2/users" );
6    const filtererddata = ourdata.data;
7    let lista = []
8    filtererddata.forEach(author => {
9      lista.push(author.slug);
10 })
11
12 res.json(lista);
13   }
```

[7] WordPress je sustav otovrenog koda za upravljanje sadržajem (CMS) koji se koristi za izgradnju i upravljanje web stranicama. URL: https://wordpress.com (25.08.2024)

```
14  catch(e){
15    return res.json("Data is not available on this wp site")
16  }
17
18 })
```

Programski kod 3. Krajnja točka (Endopoint) za skeniranje wordpress administratora

Nakon razvoja ovog alata, sljedeći korak bio je izrada *Subdomain Finder* alata. Cilj ovog alata je pronalazak poddomena povezanih s glavnom domenom, što je važno za otkrivanje dodatnih točaka napada. *Subdomain Finder* radi tako što koristi GET zahtjev prema https://crt.sh, platformi koja prikuplja podatke o certifikatima, kako bi dobio informacije o svim poddomenama vezanim uz određenu domenu. Na taj način, korisnici mogu brzo i jednostavno doći do popisa poddomena koje bi mogle predstavljati sigurnosne rizike.

Poddomeni često pružaju neočekivane ranjivosti, jer se obično ne nadgledaju pažljivo kao glavne domene, što ih čini privlačnim metama za napadače (Stutt, Pinto, 2011).

```
1  app.post('/subdomain', async (req, res) => {
2   try{
3     var  {mydomain}  = req.body
4    const ourdata = await axios.get(`https://crt.sh/?q=${mydomain}&output=json`);
5     const filtererddata = ourdata.data;
6   let lista = []
7   filtererddata.forEach(subdomain => {
8    if(lista.includes(subdomain.common_name)){
9    }else{
10   lista.push(subdomain.common_name);
11  }
12 })
13
14 res.json(lista);
15  }
```

```
16  catch(e){
17    return res.json("Subdomain not found")
18  }
19
20 })
```

Programski kod 4. Krajnja točka (Endopoint) za pronalaženje poddomene

Nakon što su ovi alati bili funkcionalni, razvijen je *Web Status and Vulnerabilities Scanner* alat, koji pruža sveobuhvatan pregled sigurnosnog stanja mrežne stranice. Ovaj alat koristi *Shodan*[8] API za prikupljanje podataka o tehnologijama i softverima koji se koriste na mrežnim stranici. Nakon prikupljanja informacija, alat provjerava verzije tih tehnologija u Nacionalnoj bazi ranjivosti (*National Vulnerability Database*) kako bi utvrdio postoje li poznati sigurnosni propusti. Na primjer, ako se otkrije da mrežna stranica koristi zastarjelu verziju *Apache* servera, alat će identificirati poznate ranjivosti za tu verziju i prikazati ih korisniku, čime omogućava brzo rješavanje potencijalnih sigurnosnih prijetnji.

```
2  app.post('/webstatus', [auth.verify], async (req, res) => {
3    let db = await connection();
4    var { ip } = req.body
5    console.log(ip)
6
7
```

[8] Shodan.io je pretraživač specijaliziran za pretraživanje i pronalaženje uređaja povezanih s internetom. URL: https://www.shodan.io (29.08.2024)

```
8  // Make request
9  try {
10   const shodan_get_data = await axios.get(`https://internetdb.shodan.io/${ip}`);
11   let shodan_data = (shodan_get_data.data)
12
13
14   let shodan_pass_data = {
15     your_email: req.jwt.email,
16     date: Date(),
17     country_name: shodan_data.country_name,
18     city: shodan_data.city,
19     ip: ip,
20     ports: shodan_data.ports,
21     os: shodan_data.os,
22     isp: shodan_data.isp,
23     vulns: shodan_data.vulns,
24
25   }
26
27   await db.collection('shodan_data').insertOne(shodan_pass_data)
28
29   return res.json({ shodan_pass_data })
30 }
31 catch (e) {
32   return res.json({ error: "Task failed" })
33 }
34
35
36
37 })
```

Programski kod 5. Krajnja točka (Endopoint) za pronalaženje sigurnosnih propusta na internet stranici

Nakon što su ovi alati bili funkcionalni, razvijen je *Web Technology Identifier*. On je razvijen s ciljem identifikacije tehnologija korištenih za izradu određene mrežne aplikacije.

Ovaj alat omogućava korisnicima da brzo i precizno otkriju koje tehnologije stoje iza određene mrežne stranice, što je ključno za procjenu sigurnosnih rizika i razumijevanje infrastrukture mrežnih aplikacija.

Za izradu ovog alata korištena je biblioteka *Wappalyzer*, koja je poznata po svojoj sposobnosti da prepozna širok spektar tehnologija, uključujući mrežne servere, okvire, biblioteke, sustave za upravljanje sadržajem (CMS), i još mnogo toga. Wappalyzer funkcionira na principu prepoznavanja obrazaca u kodu mrežnih stranica. Alat analizira HTML kod, *JavaScript*, zaglavlja, i komentare unutar koda, tražeći specifične indikatore koji otkrivaju korištene tehnologije.

Na primjer, određeni komentari unutar HTML-a ili specifične strukture *JavaScript* datoteka mogu otkriti prisutnost određenih okvira ili biblioteka. *Wappalyzer* prepoznaje ove uzorke i identificira koja je tehnologija korištena, omogućavajući korisnicima detaljan uvid u tehničku pozadinu mrežne stranice. Na temelju ovih podataka, korisnici mogu donositi bolje odluke o potencijalnim sigurnosnim prijetnjama i strategijama zaštite.

```
1   app.post('/webtech', async (req, res) => {
2     let domain = req.body;
3     try {
4       const url = domain.domain;
5
6       const options = {
7         debug: false,
8         delay: 500,
9         headers: {},
10        maxDepth: 3,
11        maxUrls: 10,
12        maxWait: 14000,
13        recursive: false,
14        probe: false,
15        userAgent: 'Wappalyzer',
```

```
16        htmlMaxCols: 2000,
17        htmlMaxRows: 2000,
18        noScripts: false,
19      };
20
21      const wappalyzer = new Wappalyzer(options)
22
23      try {
24        await wappalyzer.init()
25
26        // Optionally set additional request headers
27        const headers = {}
28
29        const site = await wappalyzer.open(url, headers)
30
31        // Optionally capture and output errors
32        site.on('error', console.error)
33
34        const results = await site.analyze()
35
36        return res.json(results);
37
38      } catch (error) {
39        console.error(error)
40      }
41
42      await wappalyzer.destroy()
43
44
45
46    }
47    catch (e) {
48      console.log(e)
49    }
50  })
51    try {
52      const shodan_get_data = await axios.get(`https://internetdb.shodan.io/${ip}`);
```

```
53      let shodan_data = (shodan_get_data.data)
54
55
56      let shodan_pass_data = {
57        your_email: req.jwt.email,
58        date: Date(),
59        country_name: shodan_data.country_name,
60        city: shodan_data.city,
61        ip: ip,
62        ports: shodan_data.ports,
63        os: shodan_data.os,
64        isp: shodan_data.isp,
65        vulns: shodan_data.vulns,
66
67      }
68
69      await db.collection('shodan_data').insertOne(shodan_pass_data)
70
71      return res.json({ shodan_pass_data })
72    }
73    catch (e) {
74      return res.json({ error: "Task failed" })}})
75
```

Programski kod 6. Krajnja točka (Endopoint) za identificiranje tehnologija određene internet stranica

Directory Listing Scanner je napravljen nakon alata *Web Technology Identifier***.** Razvijen je kako bi otkrio sigurnosne propuste na mrežnim stranicama ili direktorijumima koji ne bi trebali biti javno dostupni ili koji sadrže informacije koje bi se potencijalno mogle iskoristiti za napad na mrežnu aplikaciju. Cilj ovog alata je identificirati skrivene direktorijume ili datoteke koje su nenamjerno izložene i koje bi mogle predstavljati sigurnosni rizik.

Alat funkcionira na principu *brute-force* metode, koristeći unaprijed pripremljen rječnik (wordlist) s mogućim ranjivim stranicama ili kranjim točkama (endpoint). Svaka stavka iz rječnika se testira slanjem zahtjeva prema ciljanoj mrežnoj stranici. Ako server odgovori

sa statusnim kodom 200 (što znači da je stranica ili direktorijum dostupan), alat prikazuje tu stranicu ili direktorijum kao potencijalno ranjivu točku.

Za komunikaciju između poslužiteljskog sloja i klijentskog sloja ovog alata korištene su programske podrške; softveri, što omogućava interakciju u stvarnom vremenu i brzu obradu rezultata. Isto tako, ova metoda osigurava da korisnici mogu odmah vidjeti koje su stranice ili direktorijumi otkriveni, omogućavajući im da brzo reagiraju i poduzmu potrebne mjere kako bi osigurali svoju mrežnu aplikaciju.

```
1  app.post('/dir', [auth.verify], async (req, res) => {
2   let domain = req.body;
3   let myemail = req.jwt.email;
4   let SocketClientId = req.header("X-Socketio-Id")
5   var array = fs.readFileSync('assets/smaldir.txt', 'utf8').replace(/\r\n/g,'\n').split('\n');
6
7  res.writeHead(200, {
8   "Content-Type": "application/json",
9   "Access-Control-Allow-Origin": "*",
10  "Access-Control-Allow-Headers": "*"
11 });
12 console.log(myemail);
13 for(let i=0; i < array.length; i++){
14      try {
15       if(array[i] == "SVRTLUVORC1PRi1TQ0FOLUFORC1JVC1XSUxMLVNUT1A="){
16         io.emit(myemail, { item:array[i]} );
17         console.log("End of Dir Scaning!")
18         break
19      }
20         const { status } = await axios.get(domain.dns+"/"+array[i],{
21           httpsAgent: new https.Agent({
22             rejectUnauthorized: false
23           })
24         });
25         if (status === 200) {
26             io.emit(myemail, { item:array[i], status});
```

```
27
28 }
29         if(!socektConnection[SocketClientId]){
30           break
31         }
32       } catch (error) {
33         console.error(`Error Occured: ${error}`, array[i]);
34
35       }
36       io.emit(myemail+"interaction",{numberofinteraction:i});
37       await sleep(20);
38 }
39 });
```

Programski kod 7. Krajnja točka (Endopoint) za skeniranje dierektorijuma koji su potencijonalno ranjivi

DNS Lookup u okviru *Analyzeme.dev* alata je dizajniran da jednostavno i efikasno pruža osnovne informacije o domeni, fokusirajući se na povrat IP-adrese povezane s određenom mrežnom aplikacijom. Ovaj alat omogućava korisnicima da unesu domenu i odmah dobiju IP-adresu servera na kojemu se aplikacija nalazi.

Iako je alat jednostavan, njegova funkcionalnost je izuzetno korisna za brzu identifikaciju infrastrukture mrežne stranice, što može poslužiti kao polazna točka za daljnju analizu sigurnosnih aspekata ili mrežnu konfiguraciju.

```
1  app.post('/dnslookup', (req, res) => {
2   let DNS = req.body
3   let stringdns = JSON.stringify(DNS.DNS).replace(/"/g, "");
4   console.log(stringdns)
5   try{
6     dns.lookup(stringdns, function (err, address) {
7       if (err) {
```

```
            console.log(err);
            res.status(500).json({ error: err });
        } else {
            res.json({ address });
        }
    });


}
catch(e){
  console.log(e);
}
})
```

Programski kod 8. Krajnja točka (Endopoint) za osonvone podatke o domeni

Port Scanner je alat razvijen za otkrivanje otvorenih računalnih ulaza na određenoj mrežnoj aplikaciji ili IP-adresi. Ovaj alat omogućava korisnicima da brzo identificiraju koji računalni ulazi su otvoreni na ciljanom serveru, što je ključno za razumijevanje koje su mrežne usluge dostupne i potencijalno ranjive na napade.

Port Scanner funkcionira tako što šalje zahtjeve prema različitim računalnim ulazima na ciljanom serveru, a zatim analizira odgovore kako bi utvrdio koji su računalni ulazi aktivni i dostupni. Otkrivanjem tih ulaza, korisnici mogu dobiti uvid koje su aplikacije i servisi pokrenuti na serveru te što im omogućava da prepoznaju moguće sigurnosne prijetnje i ranjivosti.

```
//Search for open Port
app.post('/openport', async (req, res) => {
  let ipData = req.body;
  let ip = JSON.stringify(ipData.ip).replace(/[\"\\]/g, "");
  let port = parseInt(ipData.ports)
```

```
6
7
8
9   try {
10    const portScanOnIp = await nodePortScanner(ip, [port]);
11    console.log(portScanOnIp)
12    return res.json(portScanOnIp.ports);
13
14
15  }
16  catch (e) {
17
18
19
```

Programski kod 9. Krajnja točka (Endopoint) za skeniranje otovrenih mrežnih ulaza (Port) internet stranice

3.2. Integracija API-ja u vlastite projekte

Osim što je *Analyzeme.dev* platforma s raznim alatima za penetracijsko testiranje, cijeli sustav je osmišljen kao otvoren API koji se može integrirati u druge projekte i aplikacije. Ovaj API pruža pristup svim alatima unutar aplikacije *Analyzeme.dev*, uključujući *DNS Lookup, Port Scanner, Subdomain Finder, Directory Listing Scanner, Web Technology Identifier, WordPress Admin Usernames Scanner*, i *Web Status and Vulnerabilities Scanner*.

Analyzeme.dev API omogućava programerima da koriste ove alate unutar vlastitih aplikacija, skripti ili automatiziranih procesa. Na taj način, API može biti integriran u različite sustave za kontinuiranu integraciju (CI/CD), alate za nadzor mreže, ili sigurnosne sustave, pružajući im mogućnost da u stvarnom vremenu prate sigurnosno stanje svojih mrežnih aplikacija i infrastrukture.

Na primjer, koristeći *Port Scanner API*, programeri mogu automatski provjeravati otvorene mrežne ulaze na svojim serverima kao dio sigurnosnih provjera tijekom razvoja aplikacije. *DNS Lookup* API može se koristiti za brzo dobivanje IP-adresa tijekom mrežne analize, dok *Subdomain Finder* API može pomoći u otkrivanju novih poddomena koje treba zaštititi.

Kao otvoreni API, *Analyzeme.dev* pruža fleksibilnost i mogućnost prilagodbe, omogućujući korisnicima da implementiraju i automatiziraju sigurnosne procese na način koji najbolje odgovara njihovim potrebama. To čini *Analyzeme.dev* ne samo alatom za penetracijsko testiranje, već i ključnim dijelom sigurnosne infrastrukture za sve vrste projekata.

Osim integracije u druge projekte putem otvorenog API-ja, *Analyzeme.dev* također nudi funkcionalnost povijesti korištenja. Svaki korisnik ima mogućnost pregledati detaljnu povijest svih alata i testova koje je koristio unutar aplikacije (Madden, 2020).

3.3. Izrada grafičkog sučelja

Za izradu grafičkog sučelja *Analyzeme.dev* platforme korišten je klijentski sloj *Vue.js*, koji je poznat po svojoj jednostavnosti, fleksibilnosti, i moći u izradi interaktivnih korisničkih sučelja. *Vue.js* omogućava razvoj aplikacija temeljenih na komponentama, što olakšava organizaciju koda i omogućava ponovnu upotrebu elemenata sučelja kroz cijelu aplikaciju.

Ključne karakteristike okvira *Vue.js* su reaktivnost, komponentno zasnovan razvoj, jednostavna integracija i virtualni DOM. Reaktivnost *Vue.js* omogućava vezu između podataka i prikaza, što znači da se promjene u podacima automatski odražavaju u

korisničkom sučelju bez potrebe za dodatnim rukovanjem. Komponentno zasnovan razvoj odnosi se na svojstvo aplikacije da se može podijeliti na manje, neovisne komponente što olakšava razvoj, održavanje i testiranje koda. *Vue.js* se lako integrira u postojeće projekte i može se koristiti zajedno s drugim bibliotekama ili okvirima, čineći ga fleksibilnim izborom za razne vrste aplikacija. Koristi virtualni DOM koji poboljšava performanse aplikacije tako što minimizira izravne promjene na stvarnom DOM-u.

Za stiliziranje korisničkog sučelja korišten je *PicoCSS*, minimalistički CSS okvir koji je usmjeren na jednostavnost i brzinu razvoja. *PicoCSS* omogućava brzo i jednostavno stiliziranje bez potrebe za dodavanjem previše prilagođenog CSS-a, čineći sučelje čistim i modernim.

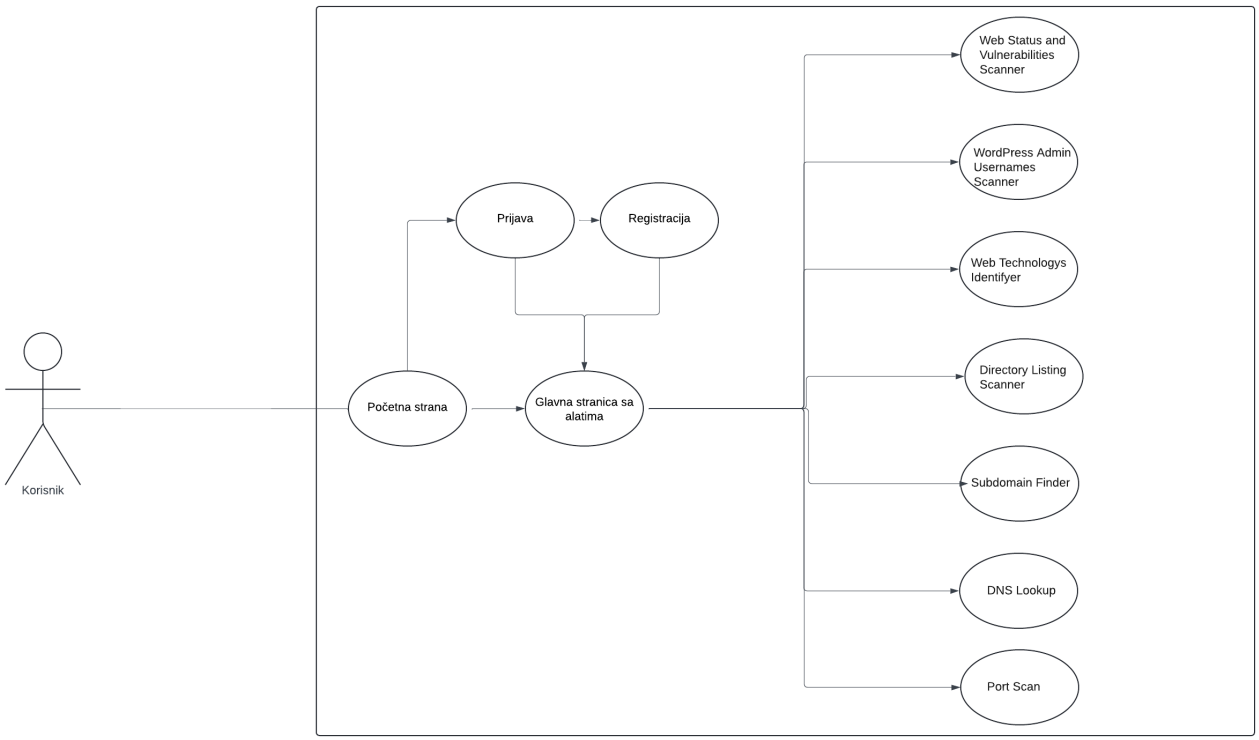
PicoCSS koristi "classless" dizajn, što znači da osnovni HTML elementi dolaze s predefiniranim stilovima, smanjujući potrebu za dodatnim stiliziranjem. To omogućava brzu izradu privlačnih i responzivnih korisničkih sučelja, savršenih za aplikacije poput *Analyzeme.dev*, gdje je fokus na funkcionalnosti i jednostavnosti korištenja (Verou, 2015).

3.4. Proces razvoja grafičkog sučelja

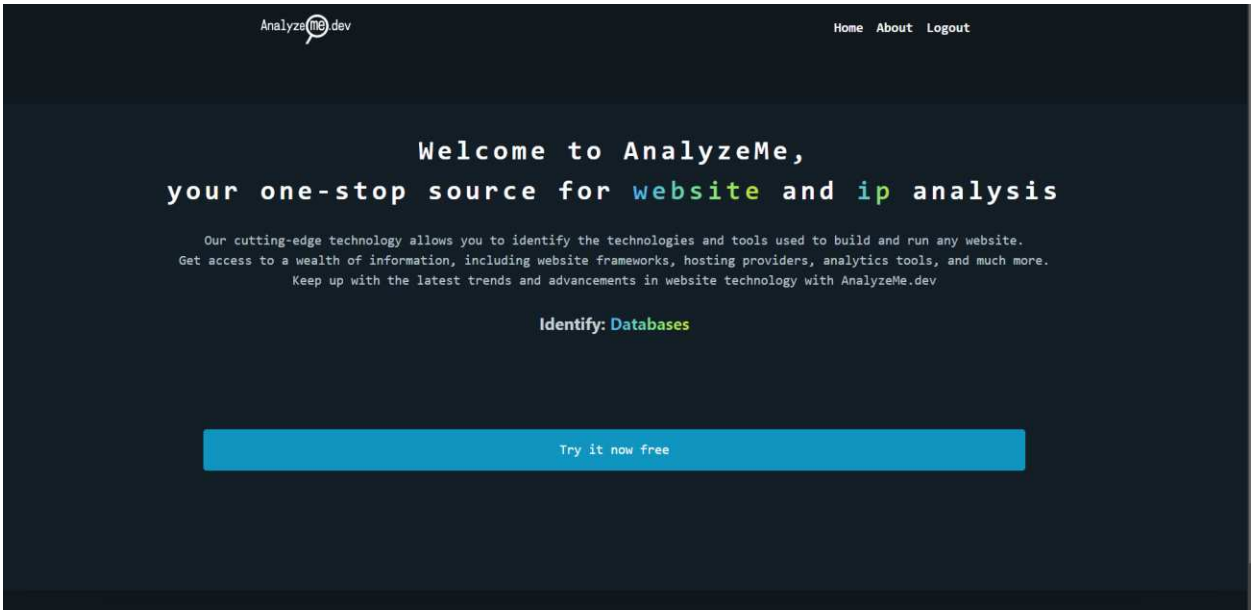
Prvo je izrađena početna stranica aplikacije koja služi kao prezentacija *Analyzeme.dev* platforme. Ova stranica je jednostavna i fokusirana na prikaz osnovnih informacija o platformi, uključujući njezine ključne značajke i prednosti.

Na početnoj stranici nalazi se također i istaknuti gumb koji korisnicima omogućava prelazak na glavnu stranicu s alatima. Ovaj dizajn je osmišljen kako bi pružio posjetiteljima brz i jasan pregled onoga što *Analyzeme.dev* nudi te ih potaknuo da istraže platformu i iskoriste alate koji su na raspolaganju.

Izrada početne stranice bila je ključna faza u razvoju aplikacije, jer je postavila temelje za daljnji dizajn korisničkog sučelja i omogućila jednostavan i intuitivan ulazak u svijet penetracijskog testiranja putem *Analyzeme.dev* platforme.



Slika 2. Dijagram klijentskog sloja platforme “analyzeme.dev”

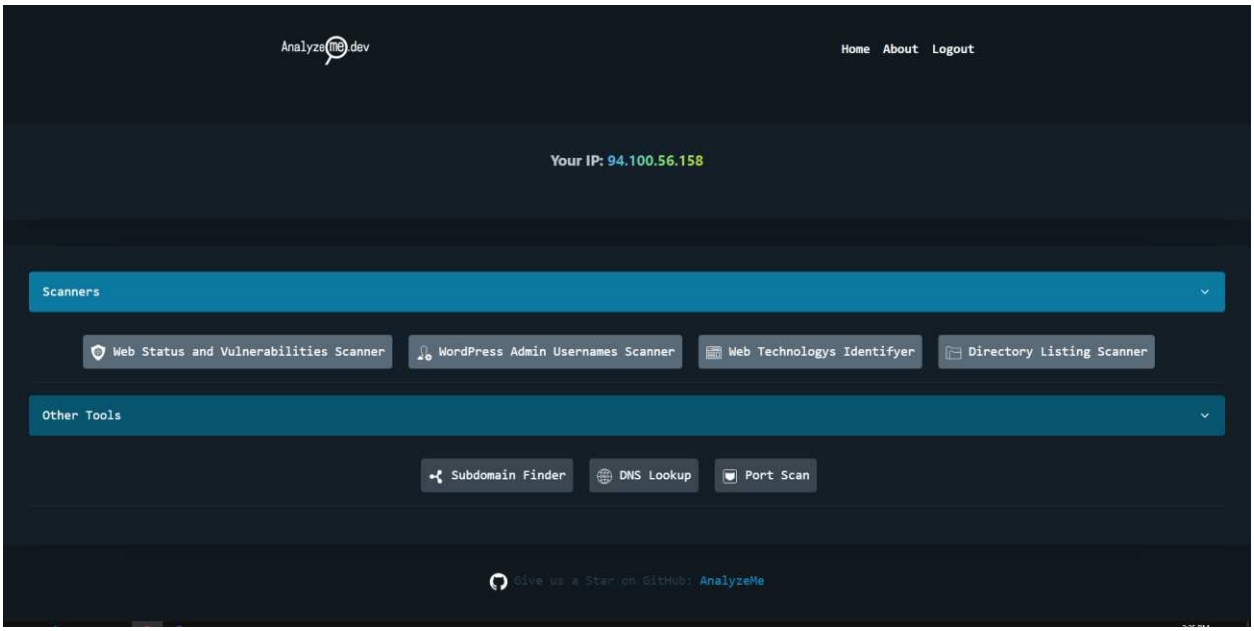


Slika 3. Naslovna strana

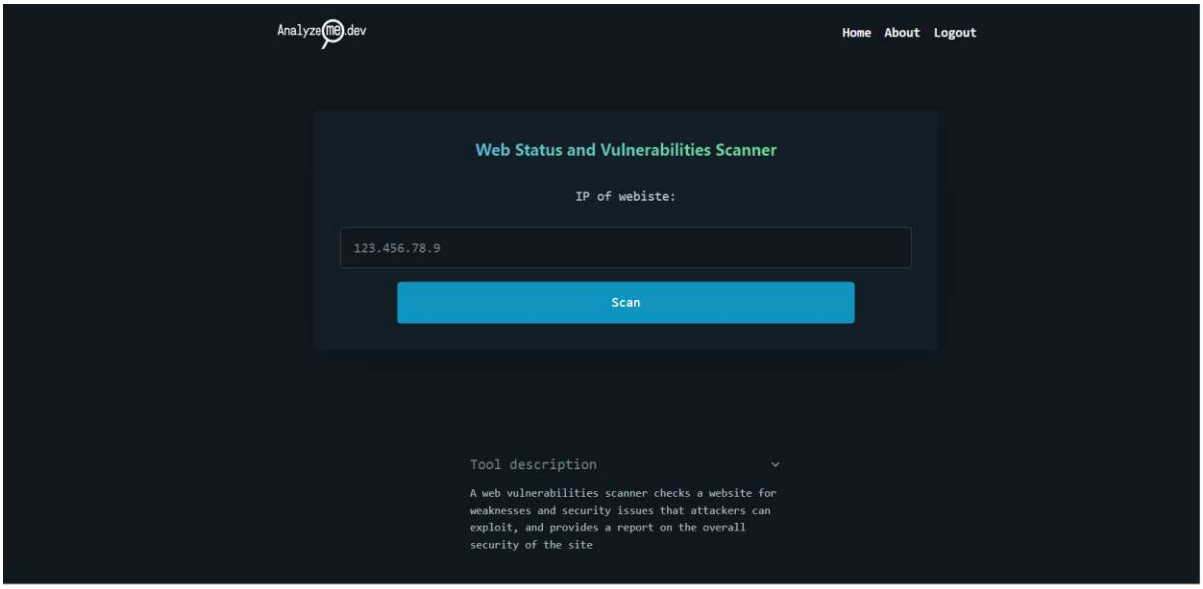
Nakon izrade početne stranice, fokus se prebacio na razvoj glavne stranice koja okuplja sve alate unutar *Analyzeme.dev* platforme. Ova stranica je centralno mjesto gdje korisnici mogu pristupiti različitim alatima za penetracijsko testiranje.

Na vrhu glavne stranice nalazi se prikaz IP adrese korisnika, što omogućava brz uvid u mrežne informacije relevantne za testiranje. Ispod ovog prikaza, alati su organizirani u dva glavna sučelja: "Skeneri" i "Ostali alati". U prvoj grupi nalaze se alati specifično dizajnirani za skeniranje mrežnih stranica i mreža uopće, poput *WordPress Admin Usernames Scanner*, *Web Status and Vulnerabilities Scanner*, i *Directory Listing Scanner.* U ostale alate spadaju alati vezani za mrežnu analizu, kao što su *DNS Lookup* i *Subdomain Finder*. Ovi alati pružaju dodatne mogućnosti za analizu mrežnih struktura i sigurnosnih postavki, čime upotpunjuju funkcionalnosti skenera.

Ovakva organizacija alata omogućava korisnicima jednostavan pristup svim potrebnim funkcijama olakšavajući proces penetracijskog testiranja i mrežne analize. Glavna stranica je dizajnirana s ciljem da sve ključne informacije i alati budu lako dostupni na jednome mjestu, čime se poboljšava efikasnost korištenja platforme.



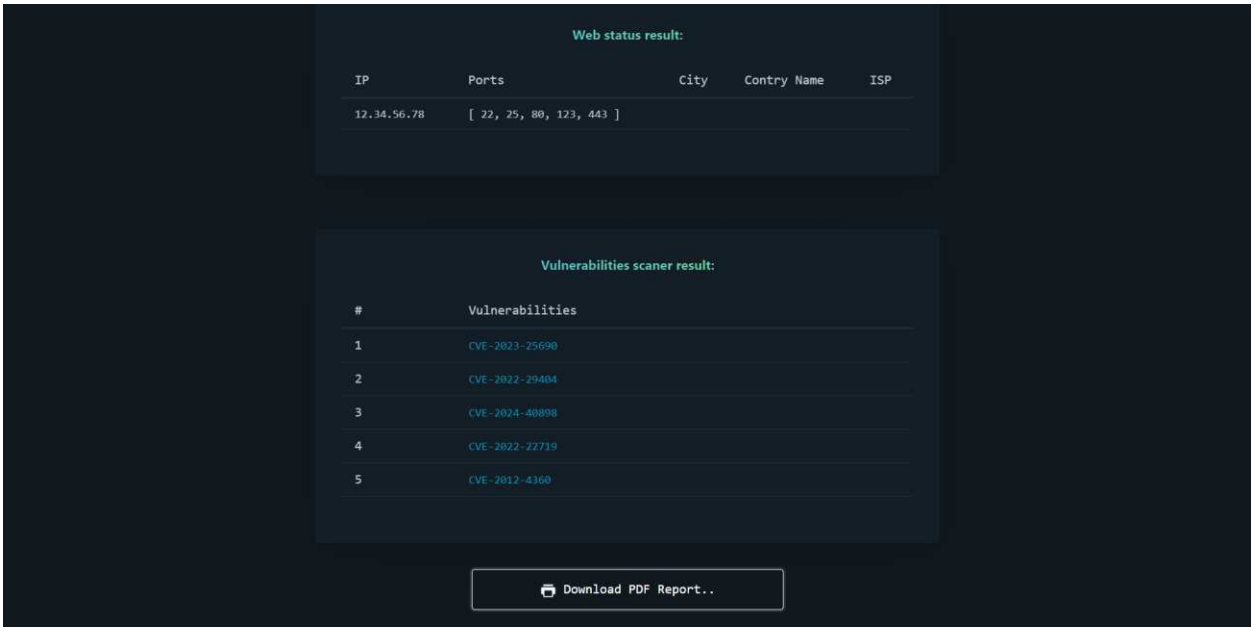
Slika 4. Početna strana sa alatima



Slika 5.Stranica „web status and vulnerabilities scanner“

Nakon što je skeniranje završeno, korisnici dobivaju detaljan pregled svih pronađenih propusta i ranjivosti. Ovi rezultati su jasno prikazani na stranici, omogućujući korisnicima da brzo pregledaju identificirane sigurnosne probleme.

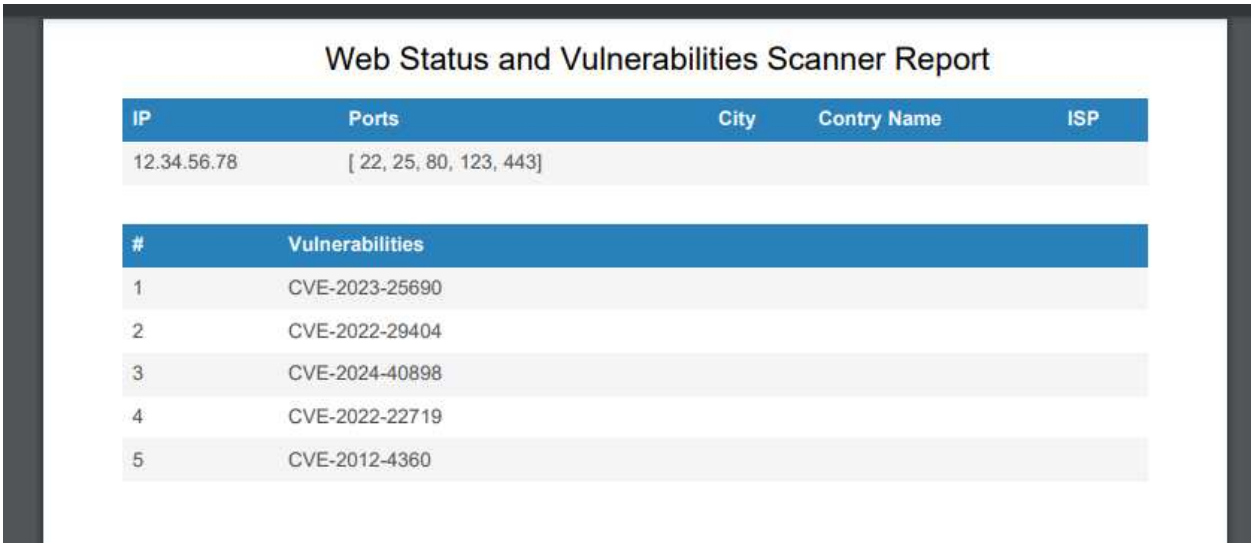
Na kraju stranice nalazi se gumb koji omogućava korisnicima da preuzmu rezultate skeniranja u PDF formatu. Ovo omogućava jednostavno spremanje i dijeljenje izvještaja o pronađenim ranjivostima. (Kim, 2014).



Slika 6. Rezultat alata „web status and vulnerabilities scanner“

U PDF-u se nalazi tablica s propustima koja nudi detaljan prikaz svih pronađenih ranjivosti. Osnovne informacije**,** ako su dostupne, uključuju podatke poput otvorenih računalnih ulaza, lokacije servera, i njegovog Internetskog servisnog provajdera (ISP).

Ovaj format omogućava korisnicima da dobiju sve relevantne informacije o sigurnosnim propustima na jasan i strukturiran način te ih lako dijele s kolegama ili koriste za daljnje analize.



Web Status and Vulnerabilities Scanner Report

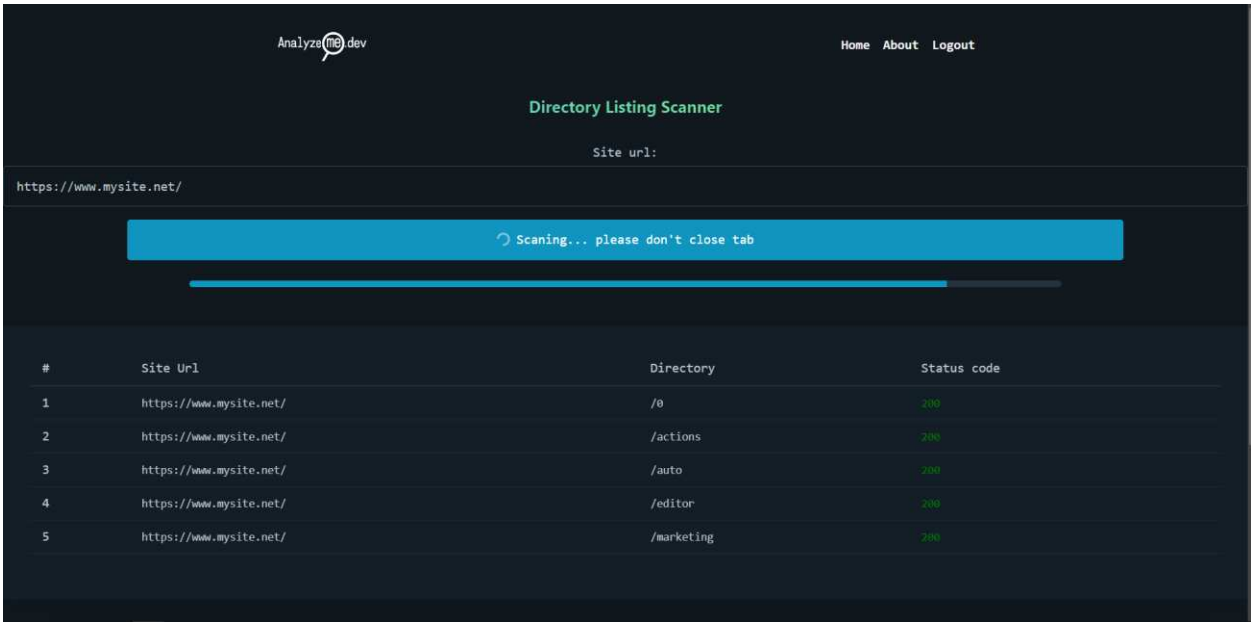
IP	Ports	City	Contry Name	ISP
12.34.56.78	[22, 25, 80, 123, 443]			

#	Vulnerabilities
1	CVE-2023-25690
2	CVE-2022-29404
3	CVE-2024-40898
4	CVE-2022-22719
5	CVE-2012-4360

Slika 7. PDF Rezultata alata „web status and vulnerabilities scanner“

Directory Listing Scanner je sljedeći alat koji je razvijen u sklopu aplikacije.

Alat razvijen za otkrivanje ranjivih direktorijuma i stranica na mrežnim serverima. Na početku sučelja alata, ispod naziva "*Directory Listing Scanner*," nalazi se polje za unos (input) u koje korisnici mogu unijeti URL stranice koju žele skenirati (SiteUrl). Ovaj unos omogućava korisnicima da specificiraju cilj skeniranja. Ispod polja za unos smješten je gumb koji pokreće proces skeniranja. Klikom na ovaj gumb, alat započinje pretraživanje navedene mrežne stranice kako bi otkrio potencijalno ranjive direktorijume i stranice. Ispod gumba nalazi se kratak opis alata koji objašnjava njegovu svrhu i način rada, naglašavajući da je alat dizajniran za identifikaciju direktorijuma koji su javno dostupni, a koji ne bi trebali biti te stranica koje sadrže osjetljive informacije koje bi se mogle iskoristiti za napade. Alat koristi *WebSocket* za komunikaciju s poslužiteljskim slojem, omogućujući brzu i učinkovitu razmjenu podataka u stvarnom vremenu.



Slika 8. Rezultat alata „Directory listing scanner“

Izrada grafičkih sučelja za alate poput *WordPress Admin Usernames Scanner*, *Subdomain Finder*, *DNS Lookup* i *Port Scanner* slijedila je sličan pristup kao kod prethodnih alata, s prilagodbama za svaku specifičnu funkcionalnost.

Za WordPress Admin Usernames Scanner, na vrhu stranice, prikazan je naziv alata "WordPress Admin Usernames Scanner". Ispod naziva nalazi se polje za unos URL-a *WordPress* stranice koju korisnici žele skenirati. Ispod tog polja nalazi se gumb koji pokreće skeniranje administrativnih korisničkih imena. Ispod gumba je kratak opis alata koji objašnjava da alat pronalazi korisnička imena administrativnih naloga na *WordPress* stranicama. Naziv alata "*Subdomain Finder*" nalazi se na vrhu stranice. Ispod naziva nalazi se polje za unos glavne domene koju korisnici žele istražiti. Gumb za pretragu pokreće proces pronalaženja subdomena povezanih s tom domenom. Kratak opis ispod gumba objašnjava da alat traži subdomene povezane s glavnim domenom.

DNS Lookup alat ima naziv "*DNS Lookup*" na vrhu stranice. Polje za unos omogućava korisnicima da unesu naziv domene čije DNS zapise žele pregledati. Ispod polja za unos nalazi se gumb koji pokreće analizu DNS zapisa, a kratak opis ispod gumba objašnjava da alat pruža IP adresu domene.

Za *Port Scanner*, naziv alata "*Port Scanner*" nalazi se na vrhu stranice, a polje za unos omogućava korisnicima da unesu IP-adresu koju žele skenirati. Gumb za skeniranje pokreće ispitivanje otvorenih portova na toj IP-adresi. Ispod gumba nalazi se kratak opis alata koji objašnjava da alat otkriva otvorene računalne ulaze i potencijalne ranjivosti.

Svi alati su dizajnirani s jasno definiranim nazivom, poljem za unos, gumbom za pokretanje funkcija i kratkim opisom koji objašnjava njihovu svrhu. Ovaj pristup omogućava korisnicima da jednostavno koriste svaki alat za specifične potrebe u evaluaciji sigurnosti.

4. Implementacija

Nakon što su grafička sučelja za sve alate bila dovršena, online alat je bio spreman. Preostao je samo korak implementacije, koji uključuje postavljanje alata na server i osiguravanje njegove funkcionalnosti u stvarnom okruženju. Implementacija omogućuje korisnicima pristup alatu putem interneta i omogućava rad svih funkcionalnosti. Prvo se započelo s kupovinom domene.

Domena je jedinstvena adresa na internetu koja omogućava korisnicima da pronađu i pristupe vašoj mrežnoj stranici. To je kao virtualna adresa vaše stranice koja se koristi za navigaciju na mreži, umjesto da se koristi IP-adresa, koja može biti složena i teža za pamćenje. Na primjer, umjesto da koristite IP-adresu poput 123.456.7.8, koristite lako pamtljivu adresu poput example.com. DNS sistem funkcioniše tako što prevodi imena domena u IP adrese koje računari koriste za međusobnu komunikaciju, što omogućava jednostavno povezivanje korisnika s odgovarajućim mrežnim stranicama (Liu, Albitz, 2006).

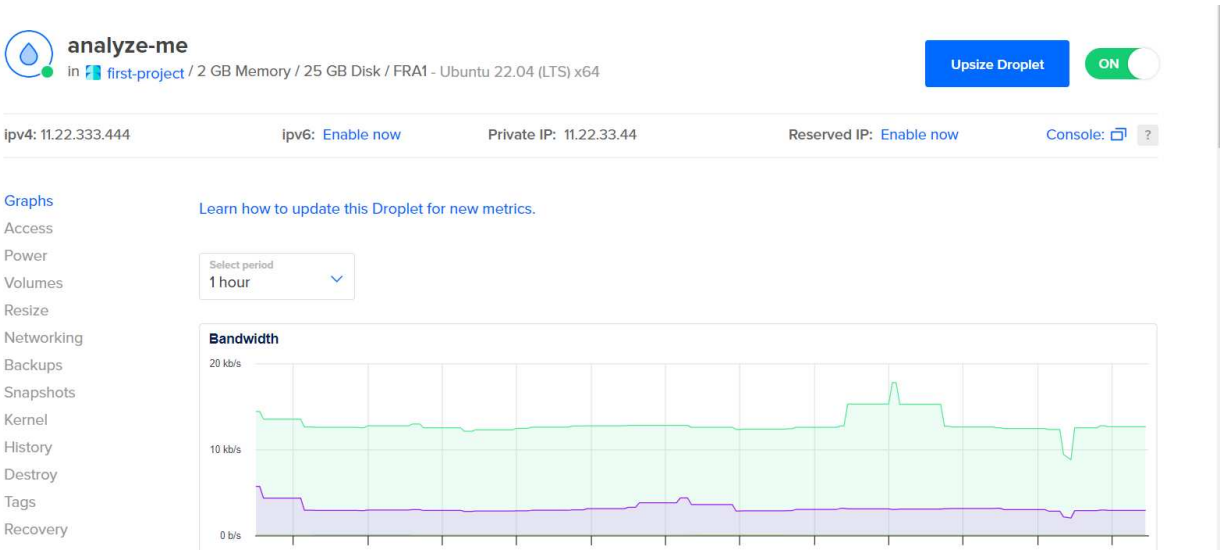
Analyzeme.dev domena je kupljena na mrežnoj stranici *Namecheap*, koja je popularna platforma za registraciju domena. *Namecheap* nudi različite usluge povezane s domenama, uključujući registraciju novih domena, upravljanje postojećim domenama, kao i dodatne usluge poput poslužianja internetskih stranica i SSL certifikata. SSL omogućaava šifriranje komunikacije između korisnikovih preglednika i web poslužitelja, što pomaže u zaštiti osjetljivih informacija, kao što su osobni podaci i financijske transakcije, od neovlaštenog pristupa (Vacca, 2012).

Kupljena je domena *Analyzeme.dev* i došao je red da se platforma implementira. Za implementaciju je odabrana *DigitalOcean*[9] platforma.

Na platformi je iznajmljena Linux Ubuntu virtualna mašina u oblaku s Intel procesorom, 2 GB radne memorije i 25 GB prostora. Ova virtualna mašina je smještena u Njemačkoj, u gradu Frankfurtu. Interakcija sa virtualnom mašinom vrši se putem komandnog sučelja. Virtualne mašine u oblaku omogućavaju fleksibilno i skalabilno upravljanje resursima, što

[9] DigitalOcean je platforma za oblačno računanje uslugzaupravljanje virtualnim serverima. Url: https://www.digitalocean.com (01.09.2024)

omogućava korisnicima da lako prilagode svoje potrebe za resursima u skladu s promjenama u opterećenju i zahtjevima (Erl, Monroy, 2023).



Slika 9. Platforma „DigitalOcean“ naslovna strana

Nakon podizanja virtualne mašine, instalira se PM2 (Process Manager 2), produkcijski menadžer procesa za *Node.js*. Ključne funkcije PM2 uključuju automatsko balansiranje opterećenja aplikacija, deklarativnu konfiguraciju aplikacija, sistem za postavljanje aplikacija i sustav za praćenje.

Slika 10. Primjer PM2 liste procesa

Pored PM2, korišten je *Nginx*[10], koji služi za serviranje mrežnih stranica, balansiranje opterećenja, obradu zahtjeva za statički sadržaj i kao reverzni *proxy* server.

Za SSL/TLS certifikat je korišten alat *CertBot*[11], alat otvorenog koda koji automatizira proces dobivanja i obnavljanja SSL/TLS certifikata od organizacije *Let's Encrypt*. Ovi certifikati se koriste za omogućavanje HTTPS enkripcije na mrežnim serverima, čime se osigurava sigurna komunikacija između servera i korisnika.

[10] Nginx je web poslužitelj otovrenog koda i reverzni server posrdnik. Razvijen je za visoke performanse i učinkovitost u upravljanju internet prometom. URL: https://nginx.org (05.09.2024).

[11] Certbot je alat otvorenog koda razvijen od strane Electronic Frontier Foundation (EFF) za automatsko dobivanje i obnovu SSL/TLS certifikata. URL: https://certbot.eff.org (05.09.2024).

5. Zaključak

Sustav za pronalazak sigurnosnih propusta *Analyzeme.dev* temelji se na integraciji nekoliko već postojećih alata s ciljem njihove centralizacije, čime se postiže ušteda vremena i povećava učinkovitost. U razvoju ove platforme korišten je programski jezik *JavaScript* te okvir *Express* za poslužiteljski sloj, dok je za korisnički sloj odabran *Vue.js*. Prva faza razvoja obuhvaćala je implementaciju sustava za autentifikaciju korisnika i registraciju novih korisnika. Nakon toga, uslijedila je izrada ključnih alata poput *Web Status and Vulnerabilities Scanner, WordPress Admin Usernames Scanner, Web Technology Identifier*, pri čemu je testiranje funkcionalnosti provedeno uz pomoć programa *Postman.*

Nakon završetka glavnih alata, izrada klijentskog sloja započela je po završetku poslužiteljskog sloja, slijedeći isti način razvoja. *Vue.js*, okvir koji se koristi za izgradnju korisničkih sučelja, pokazao se kao idealno rješenje za klijentski sloj platforme.

Nakon što su oba sloja (poslužiteljski i klijentski) bila dovršena, pristupilo se implementaciji platforme. Prvi korak bio je kupnja domene putem platforme *Namecheap*, a potom je na *DigitalOcean* odabrana virtualna *Ubuntu* Linux mašina za produkciju. Za upravljanje Node.js aplikacijama u produkciji korišten je alat *PM2*, dok je *Nginx* postavljen za poslužitelj mrežne stranice. Kako bi se osigurala sigurnost putem SSL protokola, korišten je *Certbot*, alat otvorenog koda koji automatizira proces dobivanja i obnavljanja SSL/TLS certifikata putem organizacije *Let's Encrypt.*

Popis Slika

Popis programskih kodova

Literatura

1. WEIDMAN, G. (2014). Penetration Testing: A Hands-On Introduction to Hacking, San Francisco, SAD.

2. STUTTARD, D. i PINTO M. (2011).The Web Application Hacker's Handbook: Discovering and Exploiting Security Flaws, Indianapolis, SAD.

3. MUNIZA, J. (2018). The Art of Network Penetration Testing, Birmingham, Velika Britanija

4. MARDANA, A. (2014). Express.js Guide: The Comprehensive Book on Express.js, San Francisco, SAD.

5. MARCAE, C. (2018). Vue.js Up and Running: Building Accessible and Performant Web Apps, Kalifornija, SAD.

6. CHODOROW, K. (2013). MongoDB: The Definitive Guide, Kalifornija, SAD

7. STALLINGSA, W. (2017). Cryptography and Network Security: Principles and Practice,

New Jersey, SAD.

8. WESTERVELDA, D. (2021). API Testing and Development with Postman, Birmingham, Velika Britanija.

9. KRÓL K. (2019). WordPress 5 Complete: Build Beautiful and Feature-Rich Websites from Scratch, Birmingham, Velika Britanija.

10. LIU C. i ALBITZ P. (2006). DNS and BIND, Kalifornija, SAD.

11. VACCA J. (2012). Computer and Information Security Handbook, Amsterdam, Nizozemska.

12. MONROY E. i ERL T. (2023). Cloud Computing: Concepts, Technology, Security, and Architecture, 2nd Edition, New Jersey, SAD.

13. VEROU L. (2015). CSS Secrets: Better Solutions to Everyday Web Design Problems, Kalifornija, SAD.

14. MADDEN N. (2020). API Security in Action, New York, SAD.

15. KIM P. (2014). The Hacker Playbook 2: Practical Guide to Penetration Testing, South Carolina, SAD.