SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 7193

SUSTAV DODATAKA ZA DOHVAT PODATAKA O IP ADRESAMA I DRUGIM MREŽNIM ARTEFAKTIMA IZ IZVORA NA INTERNETU

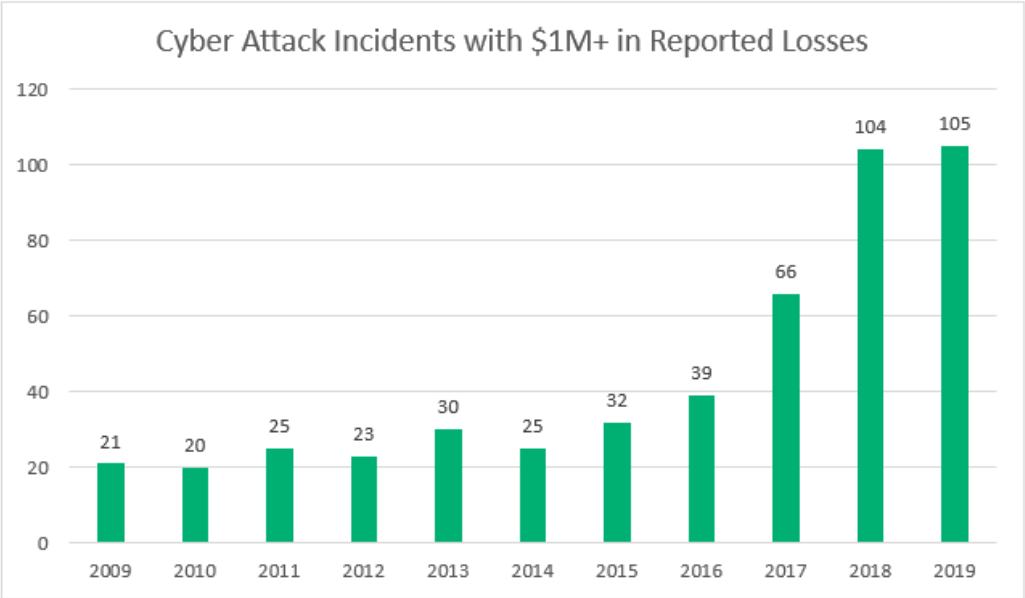
Nika Brašnović

Zagreb, lipanj 2021.

Sadržaj

Uvod

U današnje vrijeme kibernetska sigurnost se suočava s mnogobrojnim izazovima - obilje neprovjerenih informacija pristiže na internet svake sekunde, korisnika je sve više, kontrola sadržaja i sudionika postaje sve teža. Stručnjaci u tom području brojčano su nadjačani onima protiv kojih se bore – napadačima.



Slika 1. Prikaz porasta cyber napada s velikom financijskom štetom kao posljedicom u periodu 2009. – 2019. [1]

Kao što slika 1. prikazuje, cyber napadi su sve češća pojava i ne možemo očekivati da će u skorije vrijeme nestati. Iz tog razloga, internetska sigurnost postaje prioritet mnogima. Jedna od grana kibernetske sigurnosti je CTI – *Cyber Threat Intelligence.* CTI je znanje utemeljeno na dokazima o prijetnji ili opasnosti na skupu podataka koji se provjerava uključujući pokazatelje, implikacije, kontekst i mehanizme, a nekada i savjete kako bi se donijela odluka kako pravilno i na najbolji način reagirati na prijetnju.

Za izradu CTI sustava ili aplikacije potrebno je skupiti što više korisnih i pouzdanih podataka o potencijalnim prijetnjama koje se istražuju.

CTI aplikacija koja se izrađuje podrazumijeva učitavanje tekstualnih datoteka preko sučelja koje sadržavaju, između ostalog, IP adrese koje su poslale zahtjev nekom URL-u. IP adrese se tada pohranjuju u bazu podataka i analiziraju jedna po jedna (ili u grupi) te se dobiveni podatci pohranjuju ponovno u bazu podataka tako da su pohranjene i međusobne veze između podataka.

Iako se dohvaćanje podataka čini kao prilično jednostavan zadatak (mnogi su koristili razne usluge za dohvaćanje podataka o svojoj, ili nekim drugim IP adresama) izazov je pronaći uslugu koja omogućava pretragu velikog broja IP adresa, u prihvatljivom vremenu, i s pouzdanim podatcima. Uz to, preferira se korištenje usluga koje su besplatne i nude API kojim se šalje IP adresa a vraćaju podatci o njoj.

Postoje razne besplatne usluge na internetu koje pružaju informacije o IP adresama (OSINT alati) ali, kao i sve takve stranice i usluge, korištenje samo jedne ograničava rad aplikacije te količinu i vrstu informacija koje se dobivaju i prisiljava korisnika da koristi isključivo odabranu uslugu. Takav je pristup manjkav iz navedenih razloga, ograničava daljnji razvoj aplikacije i, općenito, nije dobra praksa prilikom izrade sustava koji će se nadograđivati.[2] Iz tog je razloga stvoren sustav dodataka *(eng. plugin)* za aplikaciju, izrađen u Python modulu *Pluggy*.

Sustav dodataka omogućuje korisniku (i razvojnim inženjerima koji rade na aplikaciji) da prilagode dohvaćanje podataka svojim potrebama. Uz pisanje programskog koda za dodatak, potrebno je odabrati koji dodatak će se koristiti prilikom analize IP adresa.

Prilikom izrade aplikacije korištene su tri usluge za koje su izrađeni dodatci, po načinu pristupa vrlo slične, kao i po podatcima koje vraćaju.

Funkcionalnost ovakve aplikacije i izrade dodataka je prvi korak CTI procesa – dohvaćanje pouzdanih podataka o IP adresama na temelju kojih se kasnije može dolaziti do zaključaka o IP adresi i namjeri interakcije IP adrese s određenom web stranicom.

1. CTI aplikacija za analizu IP adresa

Izazovno područje CTI procesa i OSINT pretraga inspiriralo je izradu aplikacije za analizu zapisnika koji sadrže IP adrese i web sadržaje kojima iste pokušavaju pristupiti. Iako je to relativno mali pothvat, analiza IP adresa, aplikacija uključuje znanje programskog inženjerstva (sama izrada aplikacije), sigurnosti (CTI) ali i maštovitost i snalažljivost (OSINT).

1.1. CTI – *Cyber Threat Intelligence*

CTI je vrlo uzbudljivo područje IT sigurnosti, kojemu je glavni cilj stvoriti što lukaviji i intuitivniji proces obrade podataka i donošenja zaključaka nad novim informacijama. Temelji se u potpunosti na podatcima koje dobiva i pokušava donijeti što oprezniju, ali i točniju procjenu prirode pojave o kojoj dobiva podatke. [3]

Kao što slika 1.1 prikazuje, životni ciklus CTI procesa je petlja koja se sastoji od pet faza [4]:

1. Planiranje i zahtjevi

   Potrebno je definirati jasnu misiju CTI programa – koje odluke će trebati donijeti. Ovaj korak započinje novi ciklus i vrlo je bitno svaki puta ponovno definirati ciljeve budući da se oni mogu mijenjati s obzirom na to kakav je ishod ciklusa prije.

2. Prikupljanje i obrada

   Korištenjem raznih strategija prikupljanja podataka, treba se osigurati dovoljan broj i odgovarajuća vrsta podataka koja će biti potrebna za analizu i donošenje odluke u CTI procesu.

3. Analiza

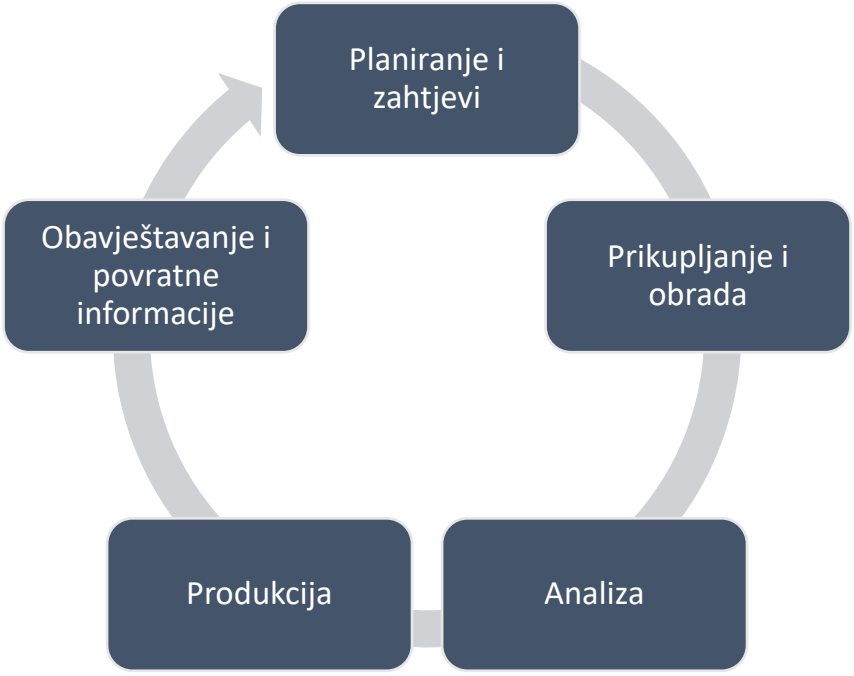
   Analiza podataka, kao srce CTI operacije, osigurava ispravne i korisne podatke za donošenje odluke o pojavama koje se promatraju.

4. Produkcija

   Korak produkcije podrazumijeva tehničke izvještaje ili sažetke koji su pravovremeni, relevantni i djelotvorni, a izrađuju se na temelju analize.

5. Obavještavanje i povratne informacije

   Obavještavanje o novim spoznajama i rezultatima analize i produkcije završni je korak jednog ciklusa CTI procesa.



Slika 1.1 Prikaz životnog ciklusa CTI procesa

Opisani koraci CTI procesa čine jedan ciklus, koji, kada završi, ponovno započinje novim spoznajama i zaključcima koji utječu na fazu planiranja i zahtjeva, a samim time i na ostale faze.

CTI aplikacija o kojoj je riječ jedan je od primjera dijela ciklusa CTI procesa – sastoji se od sljedećih faza:

- Planiranje i zahtjevi – odluka koji podatci se pretražuju, kako izgleda ulazna datoteka sa sirovim podatcima, odabir dodatka za pretraživanje IP adresa
- Prikupljanje i obrada – učitavanje datoteke, prikupljanje podataka o IP adresama
- Analiza – iako aplikacija ne dolazi sama do zaključaka, grupira podatke po nekim ključnim karakteristikama
- Produkcija – izvještaji u obliku tablica i grafova o analiziranim podatcima
- Obavještavanje i povratne informacije – kako aplikacija ne dolazi sama do zaključaka, ovaj korak se ne provodi

Glavni zadatak stvaranja CTI aplikacije i procesa jest prikupljanje podataka i obrada, što je izazovan proces koji će biti detaljnije opisan.

1.2. OSINT – Open Source Intelligence

Ranije spomenuti OSINT (*open source intelligence*) alati su vrlo moćni pružatelji relevantnih informacija o prirodi podataka koji se analiziraju. OSINT je praksa prikupljanja podataka iz javno dostupnih izvora. OSINT pretrage, bilo da ih provode stručnjaci za sigurnost ili zlonamjerni pojedinci, koriste napredne tehnike pretraživanja mora podataka koji su svima dostupni. OSINT je jedan od ambicioznih pokušaja pronalaska i kategoriziranja važnih podataka (važnih u kontekstu onoga što se smatra ciljem OSINT pretrage) te odvajanja relevatnog od nepotrebnog. Iako je relativno mlado područje sigurnosti (OSINT se prvi puta spominje 1980-ih godina u kontekstu vojnih i obavještajnih metoda prikupljanja podataka) [5], vrlo je široko i sveobuhvatno područje upravo zato što podrazumijeva prikupljanje i obradu podataka koji su svima dostupni, u svakom trenutku – koliko je izvora informacija na internetu, toliko je i mogućnosti.

CTI aplikacija, u sklopu faze prikupljanja i obrade, mora pronaći podatke o IP adresama – odabir usluga i izvora iz kojih te podatke pronalazi, koristeći OSINT alate, je opisana u nastavku rada.

2. Dohvat podataka o IP adresama

Aplikacija koja obavlja analizu i prikaz podataka o IP adresama ima jasna pravila korištenja – nakon prijave i kreiranja računa, korisnik može napraviti prijenos datoteke u tekstualnom obliku (ekstenzije *.txt*). Takve datoteke, odnosno zapisnici, sastoje se od neodređenog broja linija, od kojih je svaka linija jedinstvena kombinacija podataka o pokušaju pristupa određene IP adrese određenom URL-u (u određenom trenutku).

Jedna linija zapisnika sadrži:

- IP adresu koja je poslala zahtjev
- Datum i vrijeme kada je zahtjev poslan
- Vrstu zahtjeva i URL zahtjeva
- Status kod koji je vraćen
- Duljina odgovora
- Trajanje zahtjeva

Ovako izgleda primjer jednog takvog zapisnika:

216.244.66.231- - [30/Aug/2020:03:24:31 +0200] "GET /education/is/lv/Manual/xpce.html HTTP/1.1" 200 4016 2661 0

46.229.168.153--[30/Aug/2020:03:24:39 +0200] "GET /predmeti/de/de_labosi@zemris.fer.hr HTTP/1.1" 404 315 1843 0

178.154.200.44 - - [30/Aug/2020:03:26:44 +0200] "GET /~golub/clanci/S/iis2004.pdf HTTP/1.1" 200 53736 89591 0

66.249.64.126 - - [30/Aug/2020:03:27:34 +0200] "GET /predmeti/de/pdfs/DE_Zbirka/DIGEL_Zbirka_0.pdf HTTP/1.1" 304 - 2600 0

46.229.168.162 - - [30/Aug/2020:03:28:09 +0200] "GET /predmeti/mr/materijali/index.shtml HTTP/1.1" 200 2700 2956 0

178.154.200.44 - - [30/Aug/2020:03:29:13 +0200] "GET /~golub/clanci/ceciis2010.pdf HTTP/1.1" 200 164797 246171 0

Iz primjera zapisnika, format jednog retka je jasan :

- [IP adresa] - - [vrijeme zahtjeva] "[vrsta zahtjeva] [URL kojemu je pokušala pristupiti] [HTTP]" [status kod] [duljina odgovora] [trajanje zahtjeva] 0

Zapisnici koji su korišteni prilikom izrade i testiranja aplikacije bili su raznih duljina, od 10 do 100 000 redaka. Prilikom korištenja aplikacije, kako bi se osiguralo brzo analiziranje i prikazivanje podataka, preporučeno je koristiti zapisnike s manjim brojem redaka.

2.1. Podatci o IP adresama

Analiza zapisnika fokusira se na IP adrese, odnosno na podatke koje može dobiti o IP adresama. Procijenjeno je da je to najbitniji dio zapisnika u prvoj verziji aplikacije, dok bi za zaključke o namjeri IP adrese ili statistici zahtjeva i URL-ova bilo potrebno razmotriti i analizirati i druge dijelove zapisnika. Na primjer, povezanost vraćenih 403 statusa i IP adresa mogli bi značiti da IP adresa pokušava „ući“ negdje bez odobrenja, povezanost 403 statusa i određenih URL-ova bi mogli značiti da su neke stranice podložnije pokušajima napada, itd. Iako su mogućnosti zaista raznolike, aplikacija se trenutno bavi analizom isključivo IP adresa – pokušava geografski smjestiti IP adresu.

Budući da će aplikacija imati dodatke i omogućiti korisnicima da „napišu“ vlastite dodatke, podatci koji se dohvaćaju trebaju biti takvi da ih većina usluga koje će se koristiti mogu pružiti.

To su osnovne informacije poput:

- ime hosta - *Hostname*
- grad u kojem se računalo nalazi- *City*
- regija u kojoj se računalo nalazi - *Region*
- država u kojoj se računalo nalazi (kratica) - *Country*
- država u kojoj se računalo nalazi (puni naziv) - *Countryname*
- internetski poslužitelj i autonomni sustav - *Org*
- poštanski broj - *Postal*
- vremenska zona - *Timezone*
- geografska širina lokacije - *Latitude*
- geografska dužina lokacije – *Longitude*

2.2. Izvori o podatcima o IP adresama

Budući da je odabir izvora o podatcima IP adresa raznolik i obilan, istraženi su razni servisi i rješenja koja ih pružaju.

Prilikom odabira izvora, kriteriji koji su poštivani su:

- Alat je besplatan za korištenje (ima besplatnu verziju koju je moguće koristiti neko vrijeme ili za određeni broj upita – broj upita koji je omogućen mora biti dovoljan potrebama rada i korištenja aplikacije)
- Alat ima biblioteku (Python biblioteku, budući da je aplikacija izrađena u Django okviru) ili API koji se može koristiti

Istraženi su mnogi alati, te odabrani oni koji su najbolje odgovarali potrebama aplikacije i koji su ispunjavali kriterije.

2.2.1. Whois

Whois je jedan od omiljenih alata za pretraživanje koji vraća mnogo informacija o pružatelju usluga pristupa internetu, a primjer odgovora na zahtjev izgleda ovako [6] [7]:


```
{"response":{"registrant":"REDACTED FOR
PRIVACY","registration":{"created":"1998-08-
02","expires":"2027-08-01","updated":"2020-01-
09","registrar":"eNom,
LLC","statuses":["clientTransferProhibited"]},"name_servers":
["DNS1.P04.NSONE.NET","DNS2.P04.NSONE.NET","DNS3.P04.NSONE.NE
T","DNS4.P04.NSONE.NET"],"whois":{"date":"2021-01-
19","record":"Domain Name: domaintools.com\nRegistry Domain
ID: 1697312_DOMAIN_COM-VRSN\nRegistrar WHOIS Server:
WHOIS.ENOM.COM\nRegistrar URL: WWW.ENOM.COM\nUpdated Date:
2020-01-09T23:06:29.00Z\nCreation Date: 1998-08-
02T04:00:00.00Z\nRegistrar Registration Expiration Date: 2027-
08-01T04:00:00.00Z\nRegistrar: ENOM, INC.\nRegistrar IANA ID:
48\nDomain Status: clientTransferProhibited
https:\/\/www.icann.org\/epp#clientTransferProhibited\nRegist
rant Name: REDACTED FOR PRIVACY\nRegistrant Organization:
REDACTED FOR PRIVACY\nRegistrant Street: REDACTED FOR
PRIVACY\nRegistrant Street: \nRegistrant City: REDACTED FOR
PRIVACY\nRegistrant State\/Province: WA\nRegistrant Postal
Code: REDACTED FOR PRIVACY\nRegistrant Country: US\nRegistrant
Phone: REDACTED FOR PRIVACY\nRegistrant Phone Ext:
\nRegistrant Fax: REDACTED FOR PRIVACY\nRegistrant Email:
https:\/\/tieredaccess.com\/contact\/5d76a496-a836-430c-8de3-
eff978313f54\nAdmin Name: REDACTED FOR PRIVACY\nAdmin
Organization: REDACTED FOR PRIVACY\nAdmin Street: REDACTED FOR
PRIVACY\nAdmin Street: \nAdmin City: REDACTED FOR
PRIVACY\nAdmin State\/Province: REDACTED FOR PRIVACY\nAdmin
Postal Code: REDACTED FOR PRIVACY\nAdmin Country: REDACTED FOR
PRIVACY\nAdmin Phone: REDACTED FOR PRIVACY\nAdmin Phone Ext:
```

\nAdmin Fax: REDACTED FOR PRIVACY\nAdmin Email: REDACTED FOR PRIVACY\nTech Name: REDACTED FOR PRIVACY\nTech Organization: REDACTED FOR PRIVACY\nTech Street: REDACTED FOR PRIVACY\nTech Street: \nTech City: REDACTED FOR PRIVACY\nTech State\/Province: REDACTED FOR PRIVACY\nTech Postal Code: REDACTED FOR PRIVACY\nTech Country: REDACTED FOR PRIVACY\nTech Phone: REDACTED FOR PRIVACY\nTech Phone Ext: \nTech Fax: REDACTED FOR PRIVACY\nTech Email: REDACTED FOR 3 PRIVACY\nName Server: DNS1.P04.NSONE.NET.\nName Server: DNS2.P04.NSONE.NET.\nName Server: DNS3.P04.NSONE.NET.\nName Server: DNS4.P04.NSONE.NET.\nDNSSEC: unsigned\nRegistrar Abuse Contact Email: ABUSE@ENOM.COM\nRegistrar Abuse Contact Phone: +1.4259744689\nURL of the ICANN WHOIS Data Problem Reporting System: HTTP:\/\/WDPRS.INTERNIC.NET\/\n"},"record_source":"domaintools.com"}}

Kao što se iz primjera može vidjeti, informacije koje se dobivaju zanimljive su i potencijalno vrlo korisne, no whois alat pruža besplatnu verziju API-ja s tisuću (1000) besplatnih zahtjeva mjesečno, što je premalo za potrebe izrade i korištenja aplikacije. Iako ima dostupan API koji je vrlo elegantno rješenje, mali broj omogućenih zahtjeva razlog je zašto whois nije korišten u većoj mjeri.

2.2.2. GeoIP2

GeoIP2 alat pruža detaljne informacije o IP adresi koristeći API. Nudi par servisa – bazu zemalja, gradova, anonimnih IP adresa, ISP-a, domena, i vrsti veze [8].

Primjer odgovora na zahtjev izgleda ovako [9]:


```
{ "city": { "geoname_id": 3186886, "names": { "es": "Zagreb",
"fr": "Zagreb", "ja": "ザグレブ", "pt-BR": "Zagreb", "ru":
"Загреб", "zh-CN": "萨格勒布", "de": "Zagreb", "en": "Zagreb" }
}, "continent": { "code": "EU", "geoname_id": 6255148,
"names": { ... } }, "country": { "is_in_european_union": true,
"iso_code": "HR", "geoname_id": 3202326, "names": { ... } },
"location": { "accuracy_radius": 100, "latitude": 45.8293,
"longitude": 15.9793, "time_zone": "Europe/Zagreb" },
"postal": { "code": "10000" }, 5 "registered_country": {
"is_in_european_union": true, "iso_code": "HR", "geoname_id":
3202326, "names": { ... } }, "subdivisions": [ { "iso_code":
"21", "geoname_id": 3337532, "names": { ... } } ], "traits":
{ "autonomous_system_number": 34594,
"autonomous_system_organization": "OPTIMA TELEKOM d.d.",
"domain": "optinet.hr", "isp": "Optima Telekom",
"organization": "Optima Telekom", "ip_address": "REDACTED",
"network": " REDACTED " }, "represented_country": { "names":
{} } }
```


Iako GeoIP2 nudi biblioteku za Python, korištenje baza koje pružaju informacije o IP adresama nije besplatno- ovisno o tome koji podatci se žele koristiti, potrebno je „kupiti“ bazu [10]. Iz tog razloga, GeoIP2 nije odabran kao primjeren izvor podataka.

2.2.3. Ipinfo

Ipinfo je izvor podataka koji je zanimljiv jer nudi API zahtjeve ali i biblioteku za Python [11].

Primjer odgovora na zahtjev izgleda ovako [11]:


```
{ "ip": "8.8.8.8", "hostname": "dns.google", "anycast": true, "city": "Mountain View", "region": "California", "country": "US", "loc": "37.4056,-122.0775", "postal": "94043", "timezone": "America/Los_Angeles", "asn": { "asn": "AS15169", "name": "Google LLC", "domain": "google.com", "route": "8.8.8.0/24", "type": "business" }, "company": { "name": "Google LLC", "domain": "google.com", "type": "business" }, "privacy": { "vpn": false, "proxy": false, "tor": false, "hosting": false }, "abuse": { "address": "US, CA, Mountain View, 1600 Amphitheatre Parkway, 94043", "country": "US", "email": "network-abuse@google.com", "name": "Abuse", "network": "8.8.8.0/24", "phone": "+1-650-253-0000" }, "domains": { "ip": "8.8.8.8", "total": 11606, "domains": [ "41.cn", "onionflix.cc", "newmax.info", "ftempurl.com", "itempurl.com" ] } }
```


Ipinfo je alat koji je besplatan uz stvaranje računa i generiranje tokena koji se mora proslijediti prilikom slanja zahtjeva, a nudi sto tisuća (100 000) besplatnih zahtjeva, bez ograničenja na broj pretraga u minuti/sekundi/danu. Uz to, nudi API i Python biblioteku. Budući da zadovoljava kriterije (ima ograničenje samo na sveukupan broj zahtjeva) odabran je kao jedan od alata koji se koristi u aplikaciji.

2.2.4. Ipstack

Ipstack je alat koji nudi vrlo slične podatke kao i ostali spomenuti alati[12].

Primjer odgovora na zahtjev izgleda ovako [12]:

{ ip: "95.178.244.186" type: "ipv4" continent_code: "EU" continent_name: "Europe" country_code: "HR" country_name: "Croatia" region_code: "19" region_name: "Dubrovačko-Neretvanska" city: "Dubrovnik" zip: 20236 latitude: 42.67477035522461 longitude: 18.115510940551758 location: Object {} geoname_id: 3201047 capital: "Zagreb" languages: Object {} country_flag: "https://assets.ipstack.com/flags/hr.svg" country_flag_emoji: "HR" country_flag_emoji_unicode: "U+1F1ED U+1F1F7" calling_code: "385" is_eu: true time_zone: Object{} id: "Europe/Zagreb" current_time: "2021-06-01T06:51:56+02:00" gmt_offset: 7200 code: "CEST" is_daylight_saving: true currency: Object{} code: "HRK" name: "Croatian Kuna" plural: "Croatian kunas" symbol: "kn" symbol_native: "kn" connection: Object{} asn: 34594 isp: "Optima Telekom D D" security: Object{} is_proxy: false proxy_type: null is_crawler: false crawler_name: null crawler_type: null is_tor: false threat_level: "low" threat_types: null }

Ipstack nudi zanimljive podatke o IP adresi, poput „is_crawler“ i „threat_level“, što bi moglo biti od pomoći prilikom donošenja zaključaka o prirodi IP adrese i njenim namjerama.

Ipstack nudi besplatnu verziju računa i korištenje API-ja s pet tisuća (5 000) zahtjeva, što je relativno malo, ali budući da postoji primarni izvor (ipinfo), odabran je kao još jedan izvor podataka za koji će biti napravljen primjer dodatka [13].

2.2.5. Free IP Geolocation API – FreeGeoIP

Vrlo jednostavan izvor, FreeGeoIP nudi vrlo slične podatke kao i ostali alati[14].

Primjer odgovora na zahtjev izgleda ovako [14]:


```
{"ip":"REDACTED","country_code":"HR","country_name":"Croatia"
,"region_code":"11","region_name":"Pozesko-Slavonska
Zupanija","city":"Požega","zip_code":"34000","time_zone":"Eur
ope/Zagreb","latitude":45.3403,"longitude":17.6853,"metro_cod
e":0}
```


FreeGeoIP nudi petnaest tisuća (15 000) zahtjeva i korištenje API-ja [14], što je za potrebe razvoja aplikacije i korištenje kao jednog od dodataka dovoljno, pa je odabran kao još jedan izvor o IP adresama.

2.2.6. Greynoise

Greynoise je platforma za cybersigurnost koja analizira promet napada i internet [15]. Podatke koje prikupi nudi preko API-ja, a ima i biblioteku za Python.

Primjer odgovora na zahtjev izgleda ovako [16]:


```
{"data":{"ip":"71.6.135.131""seen":true"classification":"beni
gn""first_seen":"2018-01-28""last_seen":              "2018-2-
28""actor":"Shodan.io""tags":[0:"Mirai"1:"Telnet
Worm"]"spoofable":true"cve":[              0:"Mirai"1:"Telnet
Worm"]"vpn":true"vpn_service":"IPVANISH_VPN""metadata":{"coun
try":"United                                              States"
"country_code":"US""city":"Seattle""region":"Seattle""organiz
ation":"DigitalOcean,            LLC""rdns":"crawl-66-249-79-
17.googlebot.com""asn":"AS521""tor":false"category":"educatio
n""os":"Windows
7/8"}"raw_data":{"scan":[0:{"port":80"protocol":"TCP"}]"web":
{"paths":[0:"/robots.t    t"]    "useragents":[0"Mozilla/5.0
(compatible;                                    Googlebot/2.1;
+http://www.google.com/bot.html)"]}"ja3":[0:{"fingerprint":"c
3a6cf0bf2e690ac8e1ecf6081f17a50""port":443}]"hassh":[0:{"fing
erprint":"51cba57125523ce4b9db67714a90bf6e""port":2222}]}
}"message":"ok""results":1}
```


Iako nudi mnoge zanimljive informacije, poput procjene sigurnosti IP adrese (*classification*), Greynoise je besplatan samo 14 dana, što je za potrebe aplikacije premali vremenski period [17]. Greynoise tim je vrlo ljubazan i pristupačan, pa su za razvoj aplikacije omogućili korištenje premium računa na dva mjeseca. Tijekom razvoja, Greynoise je korišten, no budući da je licenca trajala samo dva mjeseca, nije uključen u aplikaciju kao dostupan dodatak.

2.2.7. Spyse

Spyse, kao alat za Cyber Intelligence, bio je vrlo zanimljiv kao potencijalni dodatak [18].

Primjer odgovora na zahtjev, zbog svoje veličine, nije uključen u rad ali je dostupan na službenim stranicama [19]. Spyse je vrlo moćan alat koji povezuje domene sa IP adresama, web stranice s IP adresama, prepoznaje potencijalni rizik i označava IP adrese ako taj rizik uistinu i postoji.

Spyse je alat koji nudi API, no besplatna verzija uključuje samo pretraživanje pojedinačnog IP-a preko web stranice [20]. Spyse tim također je vrlo ljubazan i vrlo rado se uključuju u projekte fakulteta tako što daju besplatnu premium licencu studentima, pa su za razvoj aplikacije omogućili korištenje premium računa na tri mjeseca. Tijekom razvoja, Spyse je korišten, no budući da je licenca trajala samo tri mjeseca, nije uključen u aplikaciju kao dostupan dodatak.

2.2.8. Ip-api

Ip-api je besplatan alat koji nudi pretraživanje na stranici, kao i korištenje API-ja [21].

Primjer odgovora na upit izgleda ovako [21]:


```
[ { "country": "США", "countryCode": "US", "city": "Чикаго",
"query": "208.80.152.201" }, { "status": "success", "country":
"United States", "countryCode": "US", "region": "VA",
"regionName": "Virginia", "city": "Ashburn", "zip": "20149",
"lat": 39.03, "lon": -77.5, "timezone": "America/New_York",
"isp": "Google LLC", "org": "Google Public DNS", "as": "AS15169
Google LLC", "query": "8.8.8.8" }, { "status": "success",
"country": "Canada", "countryCode": "CA", "region": "QC",
"regionName": "Quebec", "city": "Montreal", "zip": "H1S",
"lat": 45.5808, "lon": -73.5825, "timezone":
"America/Toronto", "isp": "Le Groupe Videotron Ltee", "org":
"Videotron Ltee", "as": "AS5769 Videotron Telecom Ltee",
"query": "24.48.0.1" } ]
```


Iako je ip-api korištenje API-ja besplatno, ne dozvoljava više od 45 upita u minuti, što je za potrebe aplikacije količinski ipak premalo. Međutim, u ponudi postoje verzije s više zahtjeva, no one se plaćaju [22]. Iz tih razloga, ip-api nije korišten prilikom izrade dodataka.

2.2.9. Ostali izvori informacija o IP adresama

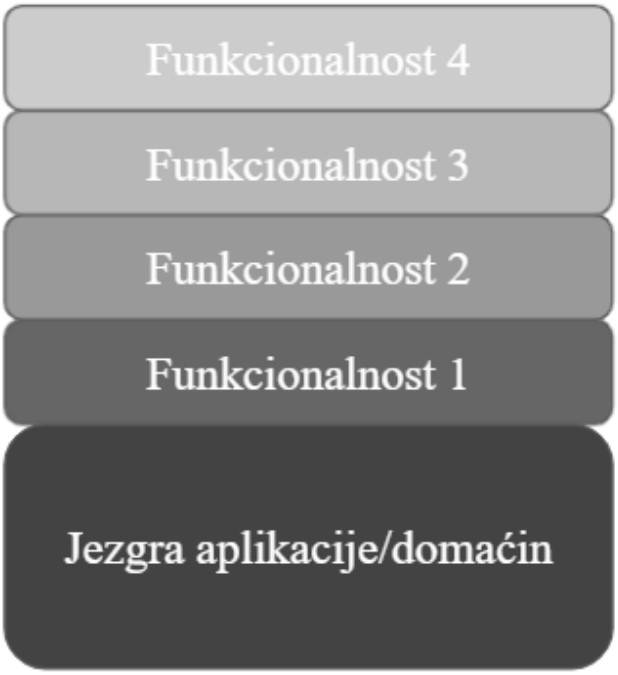
Prilikom odabira izvora informacija o IP adresama istraživani su još neke usluge koje neće biti detaljnije opisane. To su:

- geoPlugin [23]
- Ipregistry [24]
- Shodan [25]
- Viewdns.info [26]
- BuiltWith [27]
- Dnslytics [28]

Većina spomenutih usluga nema API koji se može koristiti ili se korištenje plaća, te nisu odabrane kao potencijalni dodatci u aplikaciji.

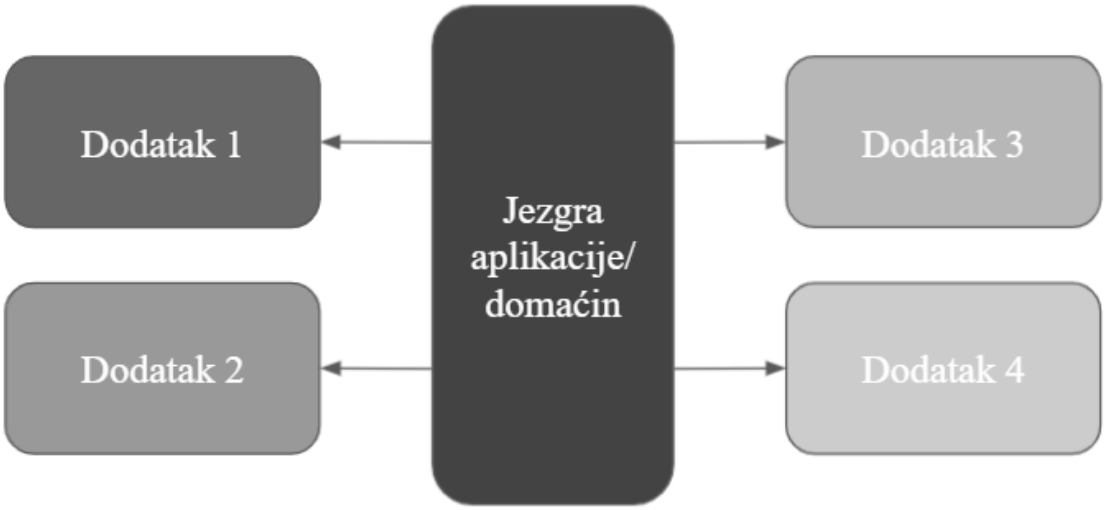
3. Pluggy modul

Klasična arhitektura raznih aplikacija podrazumijeva jezgru koda/aplikacije te nadograđivanje jezgre raznim funkcionalnostima (Slika 3.1.). Jezgra se često mora mijenjati prilikom razvoja novih funkcionalnosti, a ako se same funkcionalnosti mijenjaju, to postaje problematično u kompleksnom kodu na kojem radi više programskih inženjera ili kada se radi o kodu koji uvijek mora biti stabilan. Jezgra ovisi o radu funkcionalnosti, i obrnuto.



Slika 3.1 Prikaz klasične arhitekture bez dodataka

Za razliku od klasične arhitekture, arhitektura s dodatcima (Slika 3.2) omogućava relativno nezavisan razvoj i rad jezgre i dodataka. Iako, naravno, jezgra mora postojati kako bi dodatci postojali, ona se mijenja i nadograđuje bez da ovisi o razvoju dodataka. I obrnuto vrijedi, dodatci se razvijaju nezavisno od jezgre, te ako postoji potreba za promijenom dodataka ili nadogradnjom, ne ugrožava se funkcioniranje jezgre.



Slika 3.2 Prikaz arhitekture s dodatcima

Pluggy je jezgra upravljanja dodatcima za razvoju okolinu *pytest* [29]. Omogućava proširenje funkcionalnosti za više od 500 dodataka i prilagođavanje osnovne funkcionalnosti *pytest*-a [30].

Korisniku omogućava proširenje funkcionalnosti programa domaćina (eng. *Host program*) instalacijom dodatka koji se izvodi jednako kao i ostatak programa. Omogućava zakačivanje (eng. *Hooking*) dodatka za glavni program.

Iako postoje mnoge metode za promijenu ponašanja programa, poput nadjačavanja metoda, dodavanja funkcionalnosti u originalnom programu, izrada dodataka je bolja praksa budući da više dodataka može mijenjati isti dio izvornog koda bez da utječu jedan na drugi [29]. Naravno, to podrazumijeva pisanje izvornog koda tako da se dodatak metodama može jednostavno zakačiti.

U razvoju Pluggy dodataka potrebno je implementirati:

- Domaćina (eng. *Host*)
- Dodatak (eng. *Plugin*)
- Specifikacije kuke (eng. *Hook specification*)
- Implementacije kuke (eng. *Hook implementation*)
- Upravitelja dodataka (eng. *Plugin manager*)

Ovisno o složenosti željenih funkcionalnosti koje se implementiraju pisanjem dodataka, složenost Pluggy dodatka može se povećati i uključivati upraljavnje iznimkama, blokiranje dodataka i slično [29].

3.1. Program domaćin (eng. *Host*)

Domaćin je dio aplikacije (jedna Python datoteka), na primjer host.py. To je jezgra dodatka koja upravlja tijekom programa te otkriva, registrira i zove dodatke.

U domaćinu se uključuje pluggy kao biblioteka (na početku koda, u prvoj liniji) te postoji poziv upravitelju dodataka.

3.2. Specifikacija kuke

Specifikacija kuke je definicija koja se koristi za provjeru valjanosti svake implementacije kuke. Time se osigurava da je dodatak točno definirao implementaciju funkcije povratnog poziva (eng. *Callback*).

Specifikacija kuke se definira pomoću slično označenih funkcija, međutim pohranjuje se i analizira samo njezin potpis – ime i imena svih argumenata.

3.3. Implementacija kuke

Implementacija kuke je zapravo samo funkcija povratnog poziva (eng. *Callback*) koja je odgovarajuće označena, te ju tako označenu pluggy zna prepoznati i pozvati u pravom trenutku.

3.4. Upravitelj dodataka

Upravitelj dodataka odgovoran je za otkrivanje, registraciju specifikacije kuke i njenu implementaciju.

3.5. Primjer Pluggy dodatka

Jedan ilustrativni primjer pluggy dodatka, radi preglednosti napisan samo u jednoj datoteci.

Kod prikazan na slici 3.3. prikazuje jednostavan Pluggy primjer.

```python
import pluggy

hookspec = pluggy.HookspecMarker("CTI")
hookimpl = pluggy.HookimplMarker("CTI")


class SpecifikacijaKuke:

    @hookspec
    def kuka(self, arg1, arg2):


class Plugin_1:
    """Funkcionalnosti prvog dodatka"""

    @hookimpl
    def kuka(self, arg1, arg2):
        print("Plugin 1)")
        return arg1 + arg2


class Plugin_2:
    """Funkcionalnosti drugog dodatka"""

    @hookimpl
    def kuka(self, arg1, arg2):
        print("Plugin 2")
        return arg1 - arg2


# stvaranje upravitelja i dodavanje specifikacije kuke
pm = pluggy.PluginManager("CTI")
pm.add_hookspecs(SpecifikacijaKuke)
# registracija pluginova
pm.register(Plugin_1())
pm.register(Plugin_2())
# poziv `kuka`
rezultat = pm.hook.kuka(arg1=1, arg2=2)
print(rezultat)
```

Slika 3.3 Primjer osnovnih elemenata pluggy dodatka

Na početku (na slici 3.3 linije 1-4) se uključuje pluggy, definiraju oznake specifikacije i implementacije kuke (proizvoljno ime, ime aplikacije ili dodatka)

Specifikacija kuke (na slici 3.3 linije 7-10), koja se može prilagoditi, mora postojati i primati argumente koji se onda prenose dodatku, ali može biti prazna (kao u ovom slučaju).

Svaki je od dodataka (na slici 3.3 linije 13-28) definiran u svojoj klasi, kao metoda koja predstavlja implementaciju kuke. On prima iste argumente kao i specifikacija kuke i glavni je dio dodatka, u njemu se odrađuje posao koji se očekuje od svakog dodatka.

Slijedi (na slici 3.3 linije 31-33) stvaranje upravitelja istog imena kao i marker specifikacije i implementacije kuke. Upravitelju se dodaje specifikacija kuke i na taj način je dodatak povezan.

Zatim dolazi (na slici 3.3 linije 35-37) registracija dodatka, što predstavlja vrlo korisnu mogućnost jer se dodatci koji se ne žele koristiti mogu „isključiti“ tako da ih se jednostavno ne registrira.

Na kraju (na slici 3.3 linije 38-39), dolazi sam poziv kuke (zapravo specifikacije kuke) i predaja argumenata.

Ovaj dodatak bi nam ispisao sljedeći rezultat:

```
$ python CTI/primjer/CTI_dodatak.py
Plugin 2
Plugin 1
[-1, 3]
```

4. Implementacija dodataka

CTI aplikacija koja je izrađena uključuje tri dodatka – IpInfo, ipstack i freegeoip.

Svaki od ta tri dodatka definiran je u vlastitoj implementaciji kuke i pozvan samo ako ga se odabere. Korisnik preko sučelja aplikacije ne može odabrati dodatak, već je to trenutno moguće napraviti isključivo u samom kodu.

Odabrana baza podataka za aplikaciju je Noe4j, prvenstveno zbog svoje fleksibilnosti. Budući da je Neo4j graf baza, kada se odluči promijeniti podatke, novi čvorovi i nove veze između njih se definiraju i aplikacija nastavlja s radom [31]. Budući da je baza kreirana i razvijana neovisno o razvoju dodataka, odlučeno je da će se koristiti podatci koji su dogovoreni : *Hostname*, *City*, *Region*, *Country*, *Countryname*, *Org*, *Timezone*, *Latitude*, *Longitude.*

4.1. Ipinfo dodatak

Ipinfo alat koristi token za pristup kako bi se kontrolirao broj iskorištenih zahtjeva. Prilikom razvoja aplikacije, kreiran je jedan token i upisan u varijablu u kodu kako bi se mogao koristiti neovisno o tome tko pokreće aplikaciju. U suprotnome, svaki korisnik aplikacije bi morao kreirati korisnički račun na stranici alata i koristiti svoj token koji se šalje u URL-u zahtjeva.

Slika 4.1 prikazuje implementaciju kuke za Ipinfo dodatak. Kao argument, implementaciji kuke se šalje IP adresa koja se analizira. Budući da je Ipinfo biblioteka u Pythonu, koriste se gotove metode poput ipinfo.getHandler, gdje se kao argument predaje samo token. Podatci se vraćaju u varijabli *details* u JSON obliku.

```
@hookimpl
def myhook(self, ip_address):
    print("Using ipinfo plugin")
    access_token = 'xxxxxxxxx'
    handler = ipinfo.getHandler(access_token)
    details = handler.getDetails(ip_address)
    return details
```

Slika 4.1 Implementacija kuke dodatka Ipinfo

4.2. Ipstack dodatak

Ipstack alat također koristi token za pristup kako bi se kontrolirao broj iskorištenih zahtjeva. Prilikom razvoja aplikacije, kreiran je jedan token i upisan u varijablu u kodu kako bi se mogao koristiti neovisno o tome tko pokreće aplikaciju.

Slika 4.2 prikazuje implementaciju kuke za Ipinfo dodatak. Za razliku od Ipinfo, Ipstack nema biblioteku za Python, pa se koristi GET request poziv i predaje mu se URL koji se sastoji od adrese, IP adrese koja se analizira te tokena koji služi za pristup i dobivanje podataka. Budući da je odgovor u tekstualnom obliku, pretvara ga se u JSON i podatci koji se mogu dobiti tim alatom (svi osim hostname, timezone i org) se spremaju u mapu.

```
class Plugin_2:

    @hookimpl
    def myhook(self, ip_address):
        print("Using ipstack plugin")
        url = "http://api.ipstack.com/" + ip_address + \
            "?access_key=xxxxxxxxxxxxxxxx"
        response = requests.request("GET", url)
        parsed_json = (json.loads(response.text))
        details = {'ip_address': parsed_json['ip'],
           'hostname': 'No hostname',
           'city': parsed_json['city'],
           'region': parsed_json['region_name'],
           'country': parsed_json['country_code'],
           'countryname': parsed_json['country_name'],
           'org': 'No ORG',
           'postal': parsed_json['zip'],
           'timezone': 'No timezone',
           'latitude': parsed_json['latitude'],
           'longitude': parsed_json['longitude']
           }
        return details
```

Slika 4.2 Implementacija kuke dodatka Ipstack

4.3. FreegeoIP dodatak

FreegeoIP alat ne koristi token za pristup kako bi se kontrolirao broj iskorištenih zahtjeva.

Slika 4.3 prikazuje implementaciju kuke za Ipinfo dodatak. Za razliku od Ipinfo ali slično kao i Ipstack, FreegeoIP nema biblioteku za Python, pa se koristi GET request poziv i predaje mu se URL koji se sastoji od adrese, IP adrese koja se analizira te tokena koji služi za pristup i dobivanje podataka. Budući da je odgovor u tekstualnom obliku, pretvara ga se u JSON i podatci koji se mogu dobiti tim alatom (svi osim hostname i org) spremaju se u mapu.

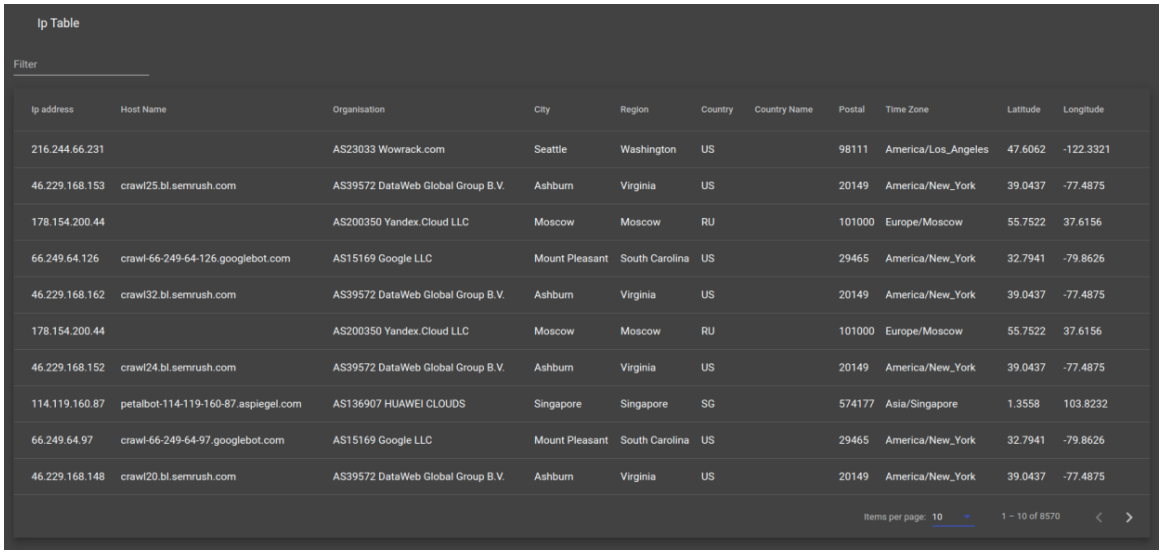
```
class Plugin_3:

    @hookimpl
    def myhook(self, ip_address):
        print("Using freegeoip plugin")
        url = "https://freegeoip.app/json/"+ ip_address
        headers = {
            'accept': "application/json",
            'content-type': "application/json"
            }
        response = requests.request("GET", url, headers=headers)
        parsed_json = (json.loads(response.text))
        details = {'ip_address': parsed_json['ip'],
           'hostname': 'No hostname',
           'city': parsed_json['city'],
           'region': parsed_json['region_name'],
           'country': parsed_json['country_code'],
           'countryname': parsed_json['country_name'],
           'org': 'No ORG',
           'postal': parsed_json['zip_code'],
           'timezone': parsed_json['time_zone'],
           'latitude': parsed_json['latitude'],
           'longitude': parsed_json['longitude']
           }
        return details
```

Slika 4.3 Implementacija kuke dodatka FreegeoIP

4.4. Vraćeni podatci o IP adresama

Nakon analize zapisnika, u sučelju aplikacije može se otvoriti zablica IP adresa te se u njoj nalaze analizirani podatci (slika 4.4.). Sve informacije o IP adresi koje je mogao pronaći, dodatak je poslao metodi koja stvara neo4j čvor u bazi, te se iz baze stvara tablica koju korisnik može vidjeti.



Slika 4.4 Prikaz podataka o IP adresama dobivenim korištenjem Pluggy dodatka

Zaključak

Podataka na internetu ima u obilju, korisnih, sigurnih, ali i lažnih, opasnih ili nebitnih. Skupljati podatke i na temelju njih donositi ispravan zaključak jest glavni cilj CTI-a, ali prepreka na putu je mnogo – koje informacije pretraživati, kojim izvorima vjerovati, kako skupljati i kategorizirati podatke i kojom logikom zaključivati o dobivenim podatcima, i još mnoga slična pitanja si postavlja svatko tko se upusti u kibernetsku sigurnost. Iako su dostupni mnogi OSINT alati, koje može koristiti svatko s računalom i internetskom vezom, korištenje i donošenje zaključaka nije trivijalan posao.

U pokušaju stvaranja relativno malene CTI aplikacije koja analizira IP adrese, a za to koristi OSINT alate, zaključeno je da se OSINT alati sami po sebi često mijenjaju, te da, ako sustav želi raditi stabilno i pouzdano, mora biti omogućeno uključivanje drugih OSINT alata. To je ostvareno uz pomoć dodataka, a modul u kojemu su se dodatci radili bio je Python modul Pluggy. Izrađena aplikacija, osim prikupljanja podataka i analize, omogućava korisniku da filtrira, pretražuje i grupira podatke i tako dolazi do zaključaka, te je tako ostvaren jedan model CTI aplikacije.

Literatura

[1] Crane C., *42 Cyber Attack Statistics by Year: A Look at the Last Decade*, Sectigo, (2021, veljača). Poveznica: https://sectigostore.com/blog/42-cyber-attack-statistics-by-year-a-look-at-the-last-decade/ ; pristupljeno 10. ožujka 2021.

[2] Wright K., *What Is A Plugin? 11 Things to Know*, ithemes, (2017, svibanj). Poveznica https://ithemes.com/what-is-a-plugin/ ; pristupljeno 17. ožujka 2021.

[3] Priyadharshini B., *Cyber Threat Intelligence Tools For Security Professionals – 2021*, Soc Investigation, (2021, veljača). Poveznica https://socinvestigation.com/cyber-threat-intelligence-tools-for-security-professionals-2021/ ; pristupljeno 17. ožujka 2021.

[4] Compton J., *The CTI Process Lifecycle: Achieving Better Results Through Execution*, FireEye, (2017, listopad). Poveznica https://www.fireeye.com/blog/products-and-services/2017/10/cti-process-lifecycle.html/ ; pristupljeno 19. ožujka 2021.

[5] Breeden II. J., Fruhlinger J., *8 top open source intelligence tools*, CSO, (2020, rujan). Poveznica https://www.csoonline.com/article/3445357/what-is-osint-top-open-source-intelligence-tools.html ; pristupljeno 10. ožujka 2021.

[6] Domaintools službene stranice. Poveznica https://whois.domaintools.com/ ; pristupljeno 02.travnja 2021.

[7] Domaintools primjer odgovora. Poveznica https://api.domaintools.com/v1/domaintools.com/whois/ ; pristupljeno 10. ožujka 2021.

[8] GeoIP2 službene stranice. Poveznica https://www.maxmind.com/en/geoip2-services-and-databases ; pristupljeno 10. travnja 2021.

[9] GeoIP2 primjer odgovora. Poveznica https://dev.maxmind.com/geoip/geolocate-an-ip/client-side-javascript ; pristupljeno 10. travnja 2021.

[10] GeoIP2 cjenik. Poveznica https://www.maxmind.com/en/geoip2-databases ; pristupljeno 10. travnja 2021.

[11] Ipinfo službene stranice. Poveznica https://ipinfo.io/ ; pristupljeno 18. travnja 2021.

[12] Ipstack službene stranice. Poveznica https://ipstack.com/ ; pristupljeno 18. travnja 2021.

[13] Ipstack cjenik. Poveznica https://ipstack.com/product ; pristupljeno 18. travnja 2021..

[14] FreeGeoIP službene stranice. Poveznica https://freegeoip.app/ ; pristupljeno 18. travnja 2021.

[15] Greynoise službene stranice. Poveznica https://greynoise.io/ ; pristupljeno 18. travnja 2021.

[16] Greynoise primjer odgovora. Poveznica https://developer.greynoise.io/reference/community-api#ip-lookup-1 ; pristupljeno 18. travnja 2021.

[17] Greynoise cjenik. Poveznica https://greynoise.io/pricing/ ; pristupljeno 18. travnja 2021.

[18] Spyse službene stranice. Poveznica https://spyse.com/ ; pristupljeno 18. travnja 2021.

[19] Spyse primjer odgovora. Poveznica https://spyse-dev.readme.io/reference/ips#ip ; pristupljeno 18. travnja 2021.

[20] Spyse cjenik. Poveznica https://spyse.com/pricing ; pristupljeno 18. travnja 2021.

[21] Ip-api službene stranice. Poveznica https://ip-api.com/ ; pristupljeno 22. travnja 2021.

[22] Ip-api cjenik. Poveznica https://members.ip-api.com/ ; pristupljeno 22. travnja 2021.

[23] GeoPlugin službene stranice. Poveznica https://www.geoplugin.com/ ; pristupljeno 22. travnja 2021.

[24] IpRegistry službene stranice. Poveznica https://ipregistry.co/ ; pristupljeno 22. travnja 2021.

[25] Shodan službene stranice. Poveznica https://www.shodan.io/ ; pristupljeno 22. travnja 2021.

[26] ViewDNS.info službene stranice. Poveznica https://viewdns.info/ ; pristupljeno 22. travnja 2021.

[27] BuiltWith službene službene stranice. Poveznica https://builtwith.com/ ; pristupljeno 22. svibnja 2021.

[28] DNSlytics službene službene stranice. Poveznica https://dnslytics.com/ ; pristupljeno 22. svibnja 2021.

[29] Pluggy dokumentacija. Poveznica https://pluggy.readthedocs.io/en/latest/ ; pristupljeno 22. svibnja 2021..

[30] Popis dodataka koje Pluggy omogućava. Poveznica https://docs.pytest.org/en/latest/reference/plugin_list.html ; pristupljeno 22. svibnja 2021.

[31] Neo4j. Poveznica: https://neo4j.com/ ; pristupljeno 24. svibnja 2021.

Sažetak

Sustav dodataka za dohvat podataka o IP adresama i drugim mrežnim artefaktima iz izvora na Internetu

Dodatci su vrlo popularno proširenje u razvoju aplikacija zbog svoje jednostavne prirode – ne utječu na glavni dio programskog koda, mogu se pozivati i uključivati u rad aplikacije neovisno o ostalim dodatcima, te razvoj novih funkcionalnosti uz pomoć dodataka ne utječe na stabilnost i funkcioniranje do tada izgrađene aplikacije. Ovaj će rad detaljnije opisati rad dodataka uz fokus na Pluggy, Python modul za izradu dodataka, na primjeru CTI (eng. *Cyber Threat Intelligence*) aplikacije koja na temelju zapisnika IP adresa pronalazi podatke uz pomoć OSINT (eng. *Open-source intelligence*) alata dostupnih na internetu. Svaki korišteni OSINT alat ima svoju implementaciju dodatka u Pluggy modulu, te funkcionira neovisno o drugim dodatcima. Također, bit će opisane osnovne karakteristike CTI procesa i OSINT alata, uz analizu raznih izvora o IP adresama koji su trenutno dostupni za korištenje.

Kjučne riječi: Obavještajni rad o kibernetičkim prijetnjama , Pluggy, sustav dodataka, izvori o IP adresama, OSINT alati

Summary

Plugin system for retrieval of data about IP addresses and other network artefacts from sources on the Internet

Plugins are a very popular extension in application development due to their simple nature - they do not affect the main part of the program code, can be called and included in the application independently of other plugins, and the development of new functionalities with the help of plugins does not affect applications. This paper will describe in more detail the work of plugins with a focus on Pluggy, a Python plugin module, on the example of CTI (Cyber Threat Intelligence) application that finds data based on IP address logs using OSINT (Open-source intelligence) tools available on the Internet. Each OSINT tool used has its own plugin implementation in the Pluggy module, and works independently of other plugins. Also, the basic characteristics of the CTI process and the OSINT tool will be described, with an analysis of the various sources on IP addresses currently available for use.

Keywords: Cyber Threat Intelligence, Pluggy, Plugin System, IP Address Sources, OSINT Tools

Skraćenice

Slijedi popis skraćenica koje su korištene u tekstu rada :

API	*Application Programming Interface*	aplikacijsko sučelje
CTI	*Cyber Threat Intelligence*	obavještajni rad o kibernetičkim prijetnjama
HTTP	*Hypertext Transfer Protocol*	protokol transportnog sloja mrežnog sustava
IP	*Internet Protocol*	internetski protokol
JSON	*Javascript Object Notation*	oblik podatka u svrhu razmjene
OSINT	*Open-source intelligence*	obavještajni rad otvorenog koda
URL	*Uniform Resource Locator*	jedinstvena lokacija resursa na web-u