Implementacija Pwnagotchi sustava

Danko, Bukovac

SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
VARAŽDIN

Danko Bukovac

IMPLEMENTACIJA PWNAGOTCHI SUSTAVA

DIPLOMSKI RAD

Varaždin, 2021.

SVEUČILIŠTE U ZAGREBU

FAKULTET ORGANIZACIJE I INFORMATIKE

V A R A Ž D I N

Danko Bukovac

Matični broj: 44021/15–R

Studij: Baze podataka i baze znanja

IMPLEMENTACIJA PWNAGOTCHI SUSTAVA

DIPLOMSKI RAD

Mentor:

Doc. dr. sc. Igor Tomičić

Varaždin, rujan 2021.

Danko Bukovac

Izjava o izvornosti

Izjavljujem da je moj diplomski rad izvorni rezultat mojeg rada te da se u izradi istoga nisam koristio drugim izvorima osim onima koji su u njemu navedeni. Za izradu rada su korištene etički prikladne i prihvatljive metode i tehnike rada.

Autor potvrdio prihvaćanjem odredbi u sustavu FOI-radovi

Sažetak

Tema ovog rada je open source program naziva Pwnagotchi implementiran na Raspberry Pi (kraće RPi) platformi što tvori Pwnagotchi sustav. Opisan je Raspberry Pi Zero W (kraće RPi0W) kao platforma na koju se program implementira i njegove karakteristike relevantne za rad sa Pwnagotchijem. Objašnjene su sastavnice programa kao što je bettercap framework za napade na WiFi i bluetooth uređaje, te implementacija umjetne inteligencije i način kako ona uči. Za probijanje lozinki koristio se hashcat te je objašnjen način na koji radi i kako se koristio s Pwnagotchijem. Rad također obuhvaća hvatanje WiFi handshakeova sa Pwnagotchijem i prikazan je put kako je korišteni Pwnagotchi učio i prilagođavao se okolišu. Kao doprinos zajednici kreirane su prevedene poruke koje Pwnagotchi ispisuje korisniku, lista probijenih lozinki, te poboljšanje wigle plugina.

Ključne riječi: sigurnost, open source, infosec, Pwnagotchi, WiFi Hacking, hardware, RPi

Sadržaj

1. Uvod

Pojam *Wireless Fidelity (kraće WiFi)* kao što ga kolokvijalno poznajemo označava skupinu bežičnih mrežnih protokola baziranih na *Institute of Electrical and Electronics Engineers (kraće IEEE)* 802.11 protokolima, koji se koriste za lokalno umrežavanje uređaja i omogućavanje pristupa internetu i sve to korištenjem radio valova tj. bez uporabe žica [1]. Veza se ostvaruje zapravo djelomično bez korištenja žica, jer se uređaj koji hoće pristup mreži ili internetu mora spojiti na router koji nudi vezu i koji je spojen žicom na internet. Pravo značenje pojma WiFi je odredila neprofitna organizacija WiFi-Alliance kao pridjev za uređaj koji je certificiran od strane ove organizacije da se bežično može spajati s ostalim WiFi uređajima, ali zbog jednostavnosti prenošenja poruke u radu koristit ću pojam WiFi kao skupinu mrežnih protokola i oblik konekcije koja se njima postiže. Bežične ili WiFi konekcije su danas najzastupljeniji oblika konekcija u svijetu, no to nije uvijek bilo tako [2]. Računalne mreže su do 1997. postojale već duži niz godina, ali te se godine puštanjem WiFi-ja u korištenje stvorio alternativni način spajanja žičnoj konekciji koja je do tog trenutka bila jedini način spajanja uređaja u mrežu. WiFi način spajanja je danas najpopularniji način spajanja na mreže samo zbog činjenice da za spajanje nije potrebna žica. 802.11 protokol danas, a ni u kojem trenutku od svoje kreacije nije mogao postići brzinu veću od Ethernet protokola kojim se vrši žična konekcija, niti je takav način konekcije sigurniji od žične konekcije [3]. Niža razina sigurnosti WiFi konekcije proizlazi iz činjenice da je veza bežična preko radio valova za koje nije moguće sakriti od svih ostalih uređaja koji slušaju na toj frekvenciji. Iz te niže razine sigurnosti nastalo je više oblika napada na vezu između uređaja i routera, jedni gdje se samo prisluškuje veza i skupljaju podaci, drugi gdje se podaci presretnu te izmijenjeni proslijede primatelju [4]. Danas je glavni protokol za osiguravanje WiFi veze WiFi zaštićeni pristup (engl. *WiFi Protected Acccess - WPA*). Taj protokol štiti podatke koji se razmjenjuju između uređaja i routera kriptiranjem čime vanjskom promatraču u obliku uređaja podaci postanu nečitljivi. Jedina velika slabost koja je preostala je činjenica da se javnom vezom, što je zapravo WiFi veza, mora poslati šifra za spajanje na router i ona čak iako se šalje kriptirana ne znači da ju je nemoguće dekriptirati, što znači da se onda neželjeni uređaji mogu spojiti na tu mrežu [5].

Tema ovog rada, sustav Pwnagtochi, nastao je iz želje kreatora da kroz korištenje sustava korisnicima na zabavan način omogući bolje upoznavanje Wifi mreža i njihovih slabosti, rada umjetne inteligencije, te natjerati korisnike da izađu iz kuće i idu prošetati sa svojim Pwnagotchijem [6]. Kreator tu ima potpuno pravo u vezi upoznavanja WiFi-ja i umjetne inteligencije, jer za početi koristiti Pwnagotchi nije nužno imati puno znanja u tim poljima, ali se kroz kontinuirano druženje s tim temama vrlo brzo upije znanje, što je jedna od motivacija za izbor ove teme. Pwnagotchi naravno nije jedina automatizirana platforma za napade na WiFi mreže, ali ima vrlo simpatičan i relativno jednostavan pristup toj temi [7]. Pwnagotchijevo ime nastaje spajanjem riječi "pwn" i nastavka "gotchi". Gotchi dolazi od kraja imena igračke Tamagotchi koja je nastala u Japanu i vezana je na određeni sa Pwnagotchijem. Naime igračka Tamagotchi je napravljena s ciljem da se korisnik brine o biću koje živi u igrački tako da ga hrani i drži sretnim, a može prepoznati potrebe igračke preko lica koje se pokazuje na ekranu. Veza između igračke i programa je što Pwnagotchi isto ima lice kojime pokazuje svoje stanje, a stanje ovisi o broju

WiFi rukovanja (engl. *handshake*) koje je Pwnagotchi "pojeo" jer mu je to "hrana". Korisnik ga može hraniti tako što šeta i upoznava Pwnagotchija s novim WiFi routerima preko kojih se on hrani. Simone Margaritelli ili Evilsocket (prema korisničkom imenu na GitHubu) Pwnagotchijev kreator osmislio je osobnost za Pwnagotchija tako što je iskoristio mogućnosti umjetne inteligencije preko koje on uči kako najučinkovitije jesti WiFi rukovanja i preko koje može izraziti svoje "emocije". Kao tijelo za svojeg "ljubimca" Evilsocket odlučio je iskoristiti Raspberry Pi Zero W, "mozak" se temelji na *Advantage-Actor-Critic* (kraće A2C) vrsti potpomognutog učenja (engl. *Reinforcement Learning - RL*), a "ruke" kojima "jede" je program bettercap, program za napade na WiFi i Bluetooth. Kroz rad objasnit će se sastavnice Pwnagotchija, proučiti način na koji on uči kroz analizu prilagodbe različitim sredinama, proučiti njegove mogućnosti probijanja šifri koje "pojede", te prikazati moj doprinos Pwnagotchi zajednici.

2. Metode i tehnike rada

Prvi dio rada vezan je za implementaciju Pwnagotchija na Raspberry Pi Zero W. Za početak implementacije potreban je program BalenaEtcher koji se koristi za zapisivanje slike s potrebnim temeljnim podacima na SD karticu koju koristi Raspberry. Alat koji je instaliran sa Pwnagotchijem je bettercap čije se sučelje koristi izravno za praćenje podataka o uhvaćenim WiFi rukovanjima, a neizravno se koriste akcije tog programa za hvatanje tih rukovanja. Program hashcat se koristi za probijanje lozinki koje Pwnagotchi uhvati. Pregled, izmjena i dodavanje podataka i datoteka u Pwnagotchi vrši se preko SSH veze za koju je potrebna SSH klijentska aplikacija PuTTY, a za dodatnu razmjenu podataka koristi se WinSCP program. Dio rada vezan za davanje doprinosa zajednici obuhvaća korištenje razvojnog okruženja programskog jezika Python za izradu skripti koje se uključuju u Pwnagotchi.

3. Opis Pwnagotchi sustava i uređaja koje podržava

Pwnagotchi je sustav koji kombinira korištenje više manjih programa za ostvarivanje svoje funkcije i predviđen je da bude instaliran na određenu hardversku podlogu koja ima potrebne performanse. Hardverska podloga koja je predviđena za Pwnagotchi je Raspberry Pi Zero W, Raspeberry Pi 3 ili 4 koji će biti opisani svome potpoglavlju. U svojim zasebnim potpoglavljima bit će opisani bettercap kao program preko kojeg Pwnagotchi napada WiFi i umjetna inteligencija koja ga pogoni, a temelji se na Advantage-Actor-Critic vrsti reinforcement učenja.

3.1. Hardverska podloga

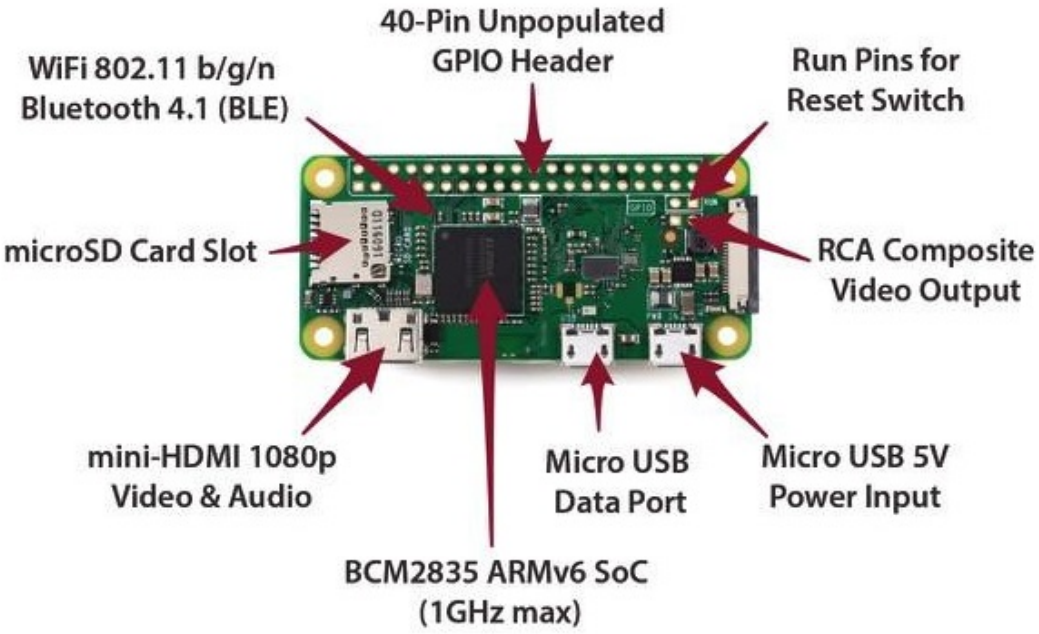
Službene upute za korištenje Pwnagotchija navode da je hardverska podloga temeljena na Raspberry Pi malim računalima i da je za najbolje rezultate i najlakše instaliranje potrebno koristiti Raspberry Pi Zero W [8]. Zbog sličnosti u načinu funkcioniranja navedeno je da su neki korisnici uspjeli instalirati Pwnagotchi na računala Raspeberry Pi 3 i 4, a autor navodi da bi u principu uz dovoljno podešavanja Pwnagotchi mogao raditi na bilo kojem računalu s Linux operativnim sustavom i WiFi sučeljem. U poglavlju usredotočit ću se na navedena Raspberry Pi računala i njihova svojstva, te navesti ću hardverske dodatke koji se preporučuju spojiti na Raspberry Pi računala.

Raspberry Pi računala su serija računala s jednom pločom (engl. *Single-board computer - SBC*) koja su razvijena u Ujedinjenom Kraljevstvu u organizaciji Raspberry Pi Foundation u suradnji s tvrtkom Broadcom [9]. Dizajn i proizvodnja imala je cilj omogućiti učenje osnova računarstva u zemljama u razvoju [10]. Ubrzo nakon početka prodaje Raspberry računala postala su dosta popularnija ne zbog svoje prvotne svrhe, nego jer su pogodna za druge svrhe gdje su potrebne funkcije koje računalo nudi. Ono što Raspberry računala nude je da rade kao stolna računala s operacijskim sustavom Linux, ali su jeftina i mala, te uz to mogu na sebe spojiti različite vanjske uređaje.

Sva Raspberry Pi računala su temeljena na sustav u čipu (engl. *System on a chip - SoC*) arhitekturi gdje se u jednom čipu nalazi procesor, grafički procesor, radna memorija, procesor za ulaz i izlaz tj. sve što treba jednom računalu da ispravno radi [11] [12]. Takva vrsta arhitekture dozvoljava da računalo ostane malo, a da ima sve potrebne funkcije, naravno zbog malog prostora koje ima za komponente performanse nisu vrlo visoke. Naravno i kod SoC arhitekture nije moguće sve nagurati u mali prostor tako da RPi računala imaju različite performanse i mogućnosti usporedno sa svojom veličinom. SoC sustav koji se koristi na RPi računalima proizvodi tvrtka Broadcom i prvi SoC koji je stavljen na RPi imao je 700MHz ARM procesor 16Kb L1 predmemorije, 128 KB memorije i performansama je bio sličan 300 MHz Pentium II procesoru. Kroz razvojnu povijest performanse procesora su rasle i sadašnji najjači model ima 1.5 GHz četverojezgreni procesor sa 1MB predmemorije. Ove vrijednosti nisu nužno finalne, jer je moguće overclockati procesore za bolje performanse nauštrb trajnosti. Najraniji modeli RPi računala koristili su 256MB radne memorije, a danas mogu imati i do 8GB radne memorije. Za umrežavanje RPi računala imaju različite mogućnosti ovisno o svojoj verziji. Za

umrežavanje mogu se koristiti Ethernet sučelja, USB Ethernet sučelja, WiFi sučelja i Bluetooth sučelja. Ovisno o verziji neka računala imaju neka i nemaju druga sučelja, iz razloga kako bi se računala pojednostavila i učinila jeftinijima. Što se tiče komunikacije s računalnim periferijama, to ovisi o konektorima koje računalo ima, a ne ovisi o driverima, jer se bilo koji driveri mogu instalirati na RPi. Neki od konektora koji se koriste: USB 2.0 ili 3.0, Micro USB, Ethernet, Display DSI, HDMI, 3.5mm audio izlaz, utor za SD karticu i izlazno-ulazni konektor opće namjene (engl. *General purpose input-output - GPIO*). Grafički procesor na SoC može generirati prikaz normalne televizijske rezolucije što znači da mu starije rezolucije također nisu problem, ali pošto je grafički procesor malen, ne može pouzdano prikazivati rezolucije veće od 2048×1152 piksela. RPi računala dolaze s Raspberry Pi OS ili Raspbian operacijskim sustavom (kraće OS) koji je 32-bitni OS temeljen na Debian distribuciji Linux-a. Podržava različite open-source operacijske sustave kao: FreeBSD, OpenBSD, Windows 10 IoT Core itd., a podržava i različite distribucije Linux-a kao: Kali, Fedora, Arch Linux itd.[13] [14] [15] [16]

3.1.1. Raspberry Pi Zero W



Slika 1: Raspberry Pi Zero W (Izvor: PixelLytical.com, 2021)

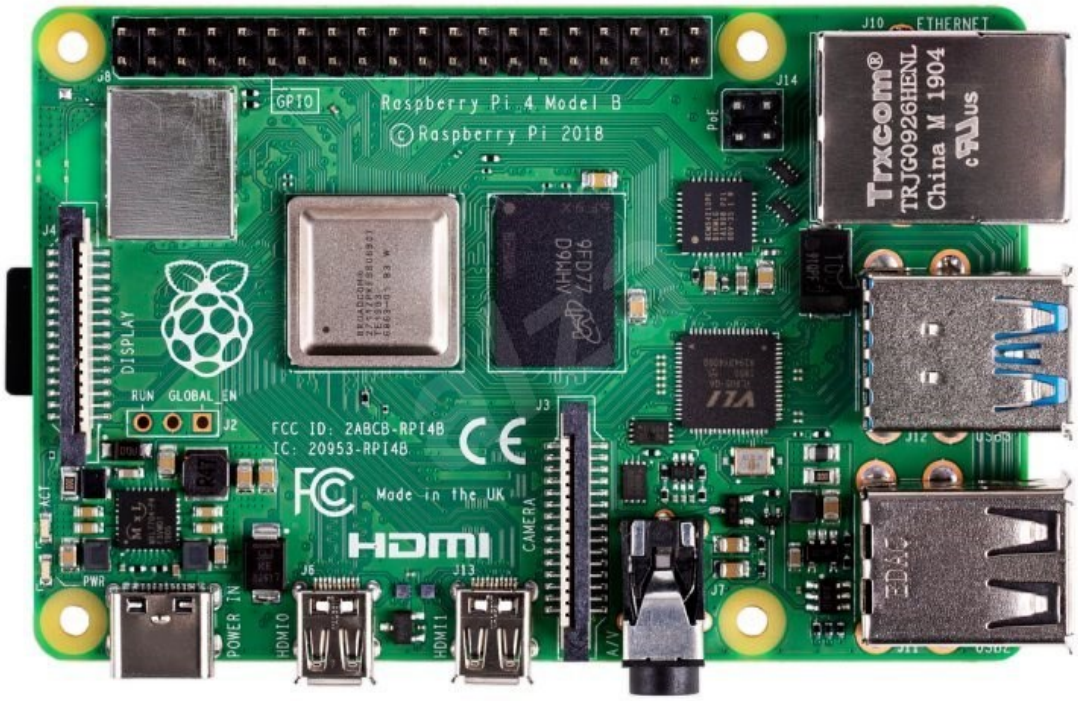
Raspberry Pi Zero W je naveden u Pwnagotchi dokumentaciji kao polazna hardverska podloga za implementaciju sustava iz više razloga. RPi0W je najmanji RPi model koji se nudi i najjeftiniji, a podržava sve funkcije koje trebaju Pwnagotchiju i ima sva potrebna sučelja. Na slici iznad mogu se vidjeti dijelovi koji čine RPi0W i kao što se može primijetiti to SoC i pridružena sučelja [17]. Procesor je dio Broadcomovog BCM2835 SoC-a, ima jednu jezgru radnog takta do 1GHz i temeljen je na ARM arhitekturi. U SoC integriran je grafički procesor Broadcom VideoCore IV radnog takta 250MHz. Raspberry Pi Zero W jedan je od modela koji nema Ethernet sučelje nego za umrežavanje koristi WiFi sučelje, može ostvariti internet vezu tako da mu se podijeli internet preko Bluetooth sučelja ili se postavi Micro USB Data Port kao

Ethernet sučelje. Ethernet sučelje je izostavljeno kako bi se smanjila pločica, jer Ethernet sučelje zahtijeva svoj zasebni čip. Micro USB 5V Power Input služi isključivo za napajanje, dok drugo USB sučelje služi i za prijenos podataka. Mini-HDMI 1080p Video & Audio očito služi za video i audio izlaz na monitor ili televiziju. MicroSD Card Slot služi za prihvat SD kartice bez koje RPi0W ne bi radio, jer mu ona služi kao mjesto za trajnu pohranu tj. kao hard-disk [8]. Preko SD kartice se vrši instalacija operacijskog sustava tako da se na SD karticu "zaprži" slika operacijskog sustava i stavi SD karticu u "bootable" stanje gdje se ponaša kao CD koji sadrži podatke za instalaciju operacijskog sustava. DSI Display sučelje je onaj duguljasti bijeli plastični dio desno od USB power inputa, tamo se može spojiti mali ekran vrste eInk koji troše malo energije i koriste se kao lagani ekrani za ispis podataka direkt sa RPi0W-a da se zaobiđe potreba da se koristi veći, manje prenosivi način prikaza. 40-Pin GPIO Header je sučelje za spajanje velike selekcije vanjskih senzora i LED lampica, te nudi RPi0W-u mogućnost obavljati mnoštvo funkcija.

Raspberry Pi Zero W je korisnicima jako zanimljiv zbog svoje cijene koja nominalno iznosi 10$ i manje, svoje veličine i svoje svestranosti. RPi0W može se koristiti u svakakvim *Internet of Things (kraće IoT)* projektima zbog mogućnosti spajanja Bluetooth-om, WiFi-jem i zbog njegovog GPIO sučelja. Različiti projekti su temeljeni na RPi0W kao moderna inačica stari prijenosnih igraćih konzola npr. Nintendo DS. Može se koristiti kao i obično računalo s monitorom, tipkovnicom i mišem, međutim današnji programi, npr. web preglednik i web stranice su prilagođeni za jače procesore i ovisno od zadatka RPi0W će trebati 5 do 10 puta duže od iPhone-a 6s da obavi zadatak. Robotika je još jedno područje gdje je RPi0W koristan, ali samo za one jednostavnije projekte gdje se radi o jednostavnoj kontroli motora ili skupljanje podataka od senzora [18].

3.1.2. Raspberry Pi 3 i 4

Raspberry Pi 3 i 4 stavljam u istu grupu jer su međusobno relativno slični što se tiče namjene i performansi. Odvojio sam ih od RPi0W jer je RPi0W napravljen da bude što manji i jeftiniji nauštrb performansi, dok RPi 3 i 4 nude podosta bolje performanse uz naravno veću cijenu i relativno malo povećanje veličine pločice. Kad kaže da su RPi 3 i 4 relativno slični mislim na to da dijele skoro sva sučelja uz dodatak nekih rjeđih sučelja na RPi 4 i to što su istog oblika i veličine [14] [15]. Njihova najveća razlika je u performansama gdje je RPi 4 bolji jer je to zadnja verzija koju je Raspberry Pi Foundation izdao. Treba napomenuti da postoje dvije vrste RPi 3, verzije A i B, gdje je jedina razlika što verzija A nema Ethernet sučelje. U performansama RPi 3 i 4 su na razini kao i pravo računalo, za razliku od RPi0W koji je sporiji od današnjeg prosječnog mobitela [18]. RPi 3 ima četverojezgreni procesor radnog takta 1.4GHz, a RPi 4 četverojezgreni procesor radnog takta 1.5GHz i oba su 64-bitni procesori. Ti procesori kombinirani sa 1GB DDR3 SD radne memorije za RPi 3 i od 1GB do 8GB DDR4 SD radne memorije za RPi 4 čine ih potpuno sposobnim raditi kao obično računalo. To prati i cijena gdje RPi3 je nominalno 35$, a RPi4 može biti od 35$ do 75$. Zbog svojih boljih performansi potrošnje energije im je također veća, od 5 do 10 puta više od RPi0W što je značajno za Pwnagotchi gdje je cilj da sustav bude što lakše prenosiv i da duže traje baterija. RPi 3 i 4 imaju



Slika 2: Raspberry Pi 4 B (Izvor: Alzashop.com, 2021)

sučelja kao i RPi0W uz dodatak ostalih kao što su: USB 2.0 i 3.0, 3.5mm audio izlaz i Ethernet. Zajedničko sa RPi0W im je također način instalacije operacijskog sustava na računalo i način pohrane preko SD kartice. To je prigodno za naš slučaj s korištenjem Pwnagotchija na RPi 3 ili 4, jer se prilikom instalacije puno ne razlikuju [8]. Prate se isti koraci, jedino se za ispravan rad mora izabrati izvor energije koji ima odgovarajuću amperažu. Preporučuje se underclockati procesor za duže trajanje baterije, a Ethernet sučelje olakšava spajanje i dijeljenje internet veze Pwnagotchiju.

3.2. Bettercap

Bettercap je jedna od dvije programske komponente koje čine Pwnagotchi, gdje je druga komponenta umjetna inteligencija. Kad ih gledamo u tom odnosu bettercap bi bio ona komponenta koja "radi", a umjetna inteligencija bi bila komponenta koja "misli". Ono što Pwnagotchi "radi" i što mu bettercap omogućava je hvatanje WiFi rukovanja koji se koriste u WPA sigurnosnom protokolu za WiFi, najraširenijem protokolu tog tipa. Bettercap je nastao prije Pwnagotchija od strane istog autora, ali njegove funkcije su šire nego one koje se koristi Pwnagotchi. Bettercap je najbolje opisao njegov kreator rekavši da je bettercap snažni, lako proširiv i prijenosni framework napisan u Gou koji nudi sigurnosnim inženjerima rješenje koje je sveobuhvatno i lako za koristiti u svrhu obavljanja izviđanja i napada na WiFi mreže, uređaje bazirane na tehnologiji bluetooth niske energije (engl. *Bluetooth Low Energy - BLE*), bežične uređaje i IPv4/IPv6 mreže [20]. Nabrojat ću specifično koje akcije bettercap nudi vezano za tehnologije, a detaljnije ću objasniti one koje koristi Pwnagotchi.

Bettercapov WiFi modul ostvaruje svoje akcije kroz WiFi sučelje koje podržava nadzorni (engl. *monitor*) načina rada i ubrizgavanje paketa (engl. *packet injection*). WiFi modul omogućava skeniranje i pronalaženje svih uređaja koji komuniciraju 802.11 protokolom, omogućava napadanje odspajanjem (engl. *deauthentication attack, deauth attack*) klijente spojene na WiFi pristupnu točku (engl. *access point - AP*), omogućava izvođenje RSN PMKID napada na ranjive APove, automatski sluša i sprema podatkovni dio WPA rukovanja ili PMKID paketa vezan za lozinku i može kreirati lažne pristupne točke ili klijente [21].

BLE modul ima akcije za otkrivanje BLE uređaja i akcije za prepoznavanje servisa i karakteristika BLE uređaja u dometu. Modul za bežične uređaje radi s bežičnim *Human interface device (kraće HID)* vrstom uređaja. HID uređaji su oni uređaji preko kojih čovjek unosi informacije u računalo i preko njega mu se vraćaju informacije iz računala. Najčešće se pod HID nazivom misli na uređaje obitelj uređaja koji služe za razmjenu informacija između čovjeka i računala i spajaju se na računalo USB sučeljem, a to su na primjer: računalni miševi i tipkovnice. Bettercapov modul može napasti te uređaje ako rade na frekvenciji od 2.4GHz koristeći USB pojačivače signala za bežične HID uređaje. Nakon otkrivanja takvih uređaja bettercapov bežični modul može poslati skriptu na takav uređaj i preuzeti kontrolu nad njime. Modul za napade na IPv4/IPv6 mreže nudi mogućnost pasivnog i aktivnog otkrivanja takvih mreža, pasivnog prikupljanja korisničkih podataka, skeniranje portova, te kreiranje lažnih korisnika i poslužitelja [22] [23] [24].

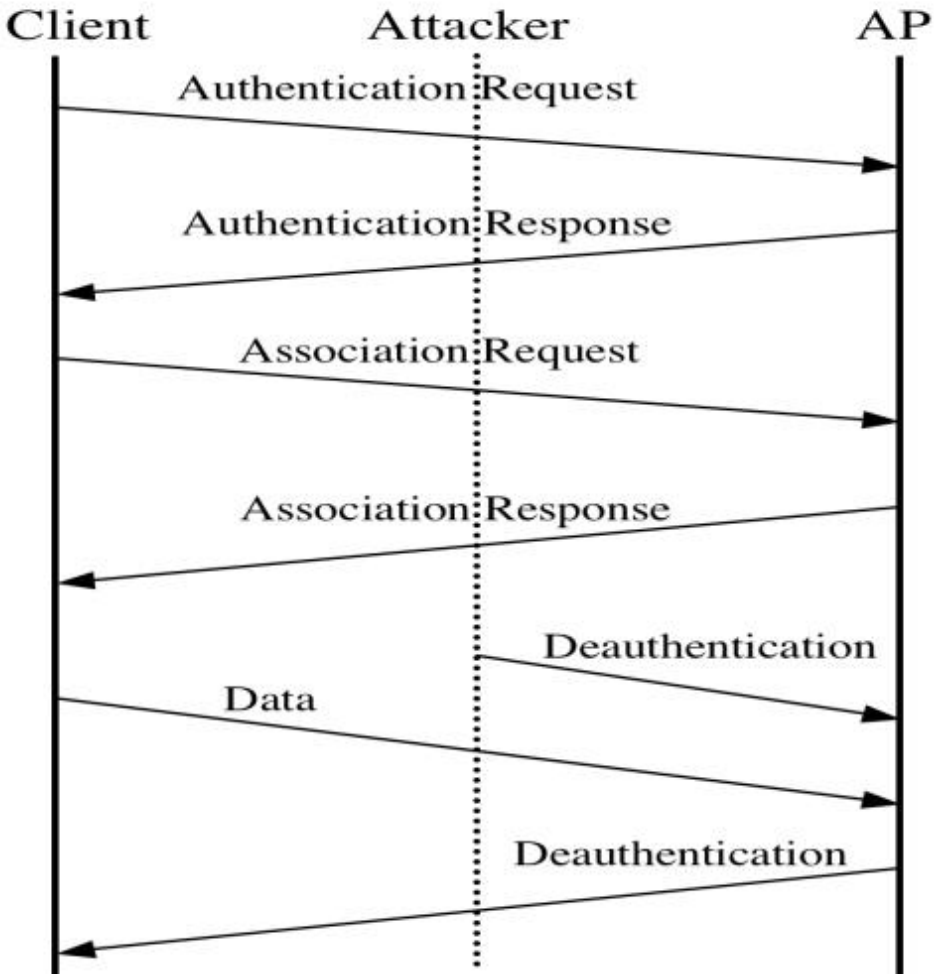
3.2.1. Bettercap WiFi modul

Bettercapov WiFi modul nam je važan jer je to jedini dio bettercapovih mogućnosti koje Pwnagotchi koristi. Svi napadi koje je moguće izvršiti na WiFi uređajima temelje se ne WiFi-jevoj osnovnoj pretpostavki ranjivosti: podaci se prenose preko javne veze. Pošto se podaci prenose preko javne veze nije teško otkriti APove i u principu je uređaju koji ima instaliran bettercap potrebno samo WiFi sučelje s monitor načinom rada i da može raditi packet injection. Monitor način rada omogućava prisluškivanje WiFi komunikacije između APa i klijenata, a packet injection omogućava da stvorimo WiFi podatkovne pakete i time što ih šaljemo određenim uređajima kontroliramo njihovo ponašanje u odnosu na WiFi mrežu.

3.2.1.1. Deauthentication napad

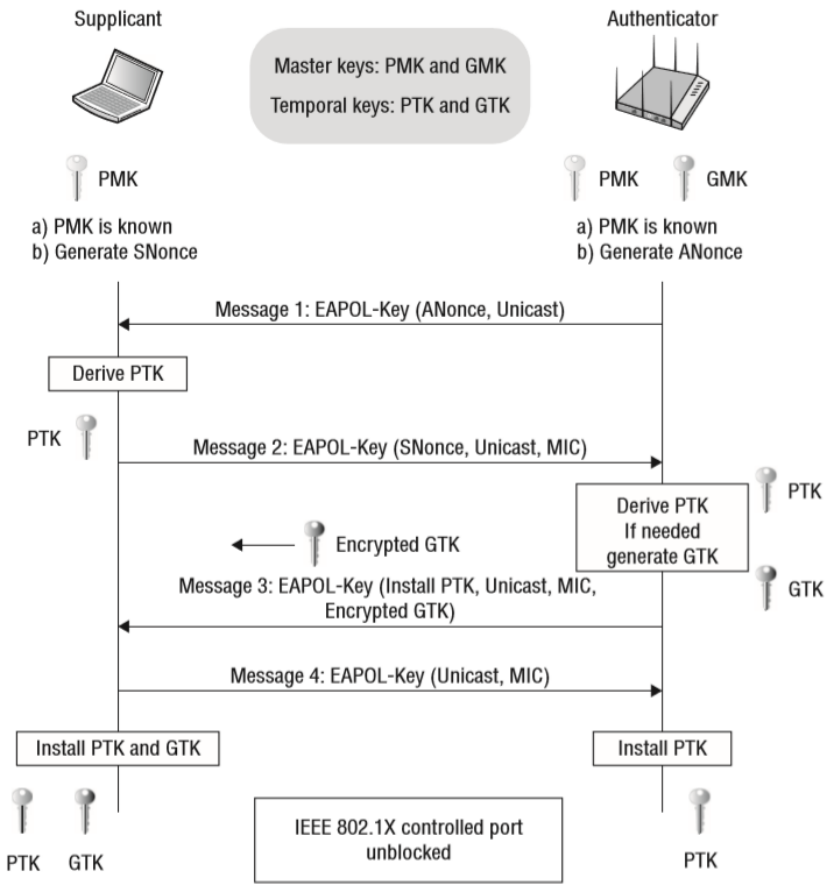
Deauthentication napad je napad vrste uskraćivanja usluge (engl. *Denial of Service - DoS*) kojime se napada jednog ili sve klijente spojene na WiFi AP i uskraćuje im se veza na mrežu koju pruža napadnuti AP [25]. DoS napad je vrsta cyber napada gdje se zbog otvorenosti internetske veze između pružatelja usluga i konzumenta usluga može tome konzumentu uskratiti usluga. WiFi veza i radio veza prenose podatke preko komunikacijskog kanala kojega čine radio valovi određene frekvencije. Deauth napad u ovom kontekstu ima istu svrhu kao i stvaranje smetnji na određenom radio kanalu i time se onemogućuje klijentu s prijemnom stanicom slušanje određene frekvencije. Međutim deauth napad ide korak dalje i koristi jednu određenu akciju koja je ugrađena u 802.11 protokol. Deauth protokol koristi deauth paket koji pošalje klijentu koji je postao "neželjen" na toj mreži i time ga "pristojno" isključi s mreže. Još

jedna slabost je potrebna da ovaj napad funkcionira, a to je da se samo podatkovni paketi kriptiraju za prijenos preko 802.11 protokola. To za deauth napad znači da se bez problema može poslati paket na AP koji sadrži adresu za pristup medijima (engl. *media access control address - MAC address*) od klijenta kojeg želimo odspojiti, a tu njegovu adresu nije teško dobiti, već se to vidi prilikom početnog izviđanja WiFi mreže.



Slika 3: Deauthentication napad (Izvor: Haitham Ameen, Shahidan M. i Haydar Imad, 2015)

Na slici gore vidi se kako napadač pošalje APu deauth paket i prema navedenoj MAC adresi AP odspoji klijenta. Postoji par razloga zašto bi napadač izveo ovaj napad. Ovaj napad se može konfigurirati preko napadačkih alata da konstantno šalje istu MAC adresu u deauth paketu, a pošto je MAC adresa fiksna za uređaje od momenta kreiranja, onda bi to značilo da bi onemogućili određenom uređaju pristup toj WiFi mreži za vrijeme trajanja napada. Kako sam naveo moguće je preko bettercapa, ali i drugih napadačkih alata stvoriti lažni AP i koristiti to da dajemo klijentu krive podatke ili da krademo njegove. Nekad je potrebno natjerati klijenta da se spoji na naš lažni AP, a to možemo lako napraviti tako što ga deauth napadom odspojimo njegovog trenutnog APa. Nama najvažnija svrha deauth napada je hvatanje WPA rukovanja, ali trebalo bi prije objasniti što je WPA rukovanje.



Slika 4: WPA rukovanje (Izvor: Wifi-professionals, 2019)

Slika gore prikazuje WPA rukovanje koje kroz 4 koraka sigurno razmjeni ključeve za sigurni prijenos podatak između klijenta i APa preko nesigurne veze. Ovo rukovanje dogodi se prije nego na slici 3 krene razmjena podataka između klijenta i APa. Podaci koji su kriptirani na ovaj način nije moguće efikasno dekriptirati tako da nam ovo rukovanje nije zanimljivo iz tog razloga. Jedna mana ovakvog rukovanja je što klijent mora poslati PTK tj. Pairwise Transient Key za pristup APu koji je izveden od PMK tj. Pairwise Master Key a on je hash vrijednost stvorena od šifre u čitljivom obliku [7]. Dovoljan je samo jedan paket od 4 u razmjeni da se može doći do šifre pristupa APu. Može se doći do šifre jer je poznata procedura hash enkripcije i dostupni su svi podaci osim šifre u čitljivom obliku, u tom slučaju hash enkripcija nije vrlo jaka, nju se može probiti u relativno kratko vrijeme preko napada grubom silom (engl. *Brute-force attack*) u usporedbi koliko brute-force napadu treba za probiti ostale vrste enkripcije. Najbrži način za probiti hash enkripciju je preko napada rječnikom (engl. *Dictionary attack*) gdje imamo listu čestih riječi i kad od njih stvorimo hash, hash koji smo mi stvorili i onaj koji smo uhvatili će se poklapati ako su prije kriptiranja sadržavali istu riječ, te smo time dobili šifru pristupa u izvornom obliku.

3.2.1.2. RSN PMKID napad

RSN je skraćenica od Remote Security Network što je identifikator jednog polja u poruci asocijacije koja dolazi klijentu od APa, a PMKID je identifikator PMK ključa koji se razmjenjuje kroz rukovanje u 4 koraka. Ova dva identifikatora jesu dio napada koji bettercap može izvesti i za razliku od deauth napada, jedini cilj mu je dobiti hash vrijednost šifre mreže koja se gore navedenim metodama može dekriptirati. Ova vrsta napada otkrivena je relativno nedavno 2018. godine od strane admina stranice hashcat. Oni su shvatili da puno APa prilikom koraka asocijacije u paketu šalje polje RSN koje sadrži PMKID. Za razliku od deauth napada koji zahtjeva da je bar jedan već autorizirani klijent spojen na AP, ovaj napada se može izvesti na AP koji nitko nije spojen. Dovoljno je da klijent, a u ovom slučaju napadač, pošalje asocijacijski zahtjev i može dobiti PMKID. Iz PMKIDa može se izvući hash vrijednost šifre mreže, jer je PMKID kriptiran kroz formulu "PMKID = HMAC-SHA1-128(PMK, "PMK Name" | MAC_AP | MAC_STA)", a iz te formule znamo svaku varijablu osim PMKa što je dovoljno podataka za izvući hash vrijednost šifre mreže. Ova novija vrsta napada smatra se boljom opcijom kada želimo samo pokupiti kriptiranu šifru mreže, jer ostavlja manji trag u smislu da ne mora odspojiti klijenta s mreže, ne mora postojati spojeni klijent i zapravo velika većina današnjih AP-ova je ranjiva na ovaj oblik napada [28].

3.2.1.3. Pasivno prikupljanje WPA rukovanja ili PMKID podataka

Treća vrsta napada koje bettercap nudi je pasivno slušanje okoliša. Kako su podaci koje se šalje WiFijem javni samo pasivnim promatranjem mogu se prikupiti podaci iz okoliša. Gore navedeni napadi su aktivnog i invazivnog tipa, gdje postoji namjerna interakcija između napadača i APa, ali uz dovoljno volje i vremena, sve što deauth i RSN PMKID napadi nude može se postići pasivnim promatranjem. U slučaju pasivnog promatranja rezultati ovise potpuno od okoliša u kojem se nalazimo i ovisimo o akcijama ljudi oko nas, znači podaci o WPA rukovanju i PMKID mogli bi se prikupiti samo kada neki klijent u našoj blizini se ide spojiti na AP [7].

3.3. Umjetna inteligencija

Umjetna inteligencija (engl. *Artificial Intelligence - AI*) je znanstveno polje koje kombinira računalno inženjerstvo i robusne skupove podataka za rješavanje problema [29]. Obuhvaća polja strojnog učenja i dubokog učenja koja su podskupine umjetne inteligencije. Kada se govori da negdje postoji umjetna inteligencija koja radi nešto, točnije bi bilo reći da je implementiran neki od AI algoritama koji radi predviđanja ili klasifikaciju podataka. AI kojeg koristi Pwnagotchi služi da bi bolje predvidio kako postupiti da se skupi što više WPA šifri. Taj AI koristi duboko učenje (engl. *Deep Learning*) kao način učenja koji koristi neuralne mreže za proces učenja. Duboko učenje razlikuje se od strojnog učenja što algoritmi dopuštaju da stroj automatski prepoznaje podatke koji mu se daju i zna kako učiti iz njih za razliku od strojnog učenja gdje čovjek intervenira davanje strogo strukturiranih skupova podataka za učenje.

Pwnagotchijev AI koristi *Reinforcement learning (kraće RL)* algoritam iz skupine *Deep learning* algoritama. RL je način učenja za AI gdje je proces učenja vrlo sličan onome kako ljudi i životinje uče. Temelji se na činjenici da osoba ili životinja ili program u AI slučaju naprave neku akciju i osjete i procjene posljedicu te akcije. U terminima RLa agent se kreće kroz stanja tako što poduzima akcije zbog nagrade koju dobije i procjenjuje [30]. Pwnagotchi koristi A2C model učenja koji spada pod RL algoritam, ali postoje još dva modela, model temeljen na vrijednostima i model temeljen na pristranosti. A2C model zapravo kombinira ova zadnja dva model u dvije neuralne mreže koje koristi za donošenje odluka [31]. Model temeljen na vrijednostima pridaje stanju i akciji prema tom stanju vrijednost i bira se akcija koja vodi prema stanju s najvećom vrijednosti. Model temeljen na pristranosti upravlja agentom prema prijašnjim iskustvima s određenim kombinacijama akcije i stanja. Ono što je karakteristično za Deep learning je da prema iskustvima algoritmi mijenjaju težišne vrijednosti da daju drugačiji izlaz tj. uče, ali ovisno o modelu uče na drugačiji način.

A2C koristi dvije neuralne mreže gdje jedna mreža uzdržava model temeljen na pristranosti koji se u A2C zove vođa (engl. *Actor*), a druga neuralna mreža uzdržava model temeljen na vrijednostima koji se u A2C zove kritičar (engl. *Critic*). A2C s dvije neuralne mreže radi ciklički tako što za ulazne podatke prima senzorne podatke iz svoje okoline, koja može biti fizička kao sa Pwnagotchijem ili virtualna, te daje dva izlazna podatka svaki za jednu neuralnu mrežu. Izlazni podatak koji generira kritičar je procjena koliko je nagrade moguće očekivati od trenutne točke nadalje, a izlazni podatak koji generira vođa je preporuka koju akciju poduzeti prema prijašnjim iskustvima [30]. Nakon odabira akcije ta akcija rezultira nekom nagradom koja se bilježi za kasniju analizu. Vrijeme za analizu podataka prema A2C modelu dolazi nakon svakih X broja obavljenih akcija, što je nešto drugačije od Monte Carlo modela koji analizira svoje postupke tek na kraju puta ili drugih modela koji analiziraju svoje postupke nakon svake akcije. Analiza akcija važan je trenutak, jer se rezultatima analize mijenjaju težišne vrijednosti algoritma tj. doživljava se trenutak učenja. Monte Carlo model uči na kraju puta jer tako sa sigurnošću može znati koja je bila prava vrijednost svakog stanja kroz koje je prošao, jer je vrijednost zadnje stanja 0 i od tog stanja do prvoga sumiraju se sve dobivene nagrade i dobijemo vrijednost prvog stanja. Analiza na kraju puta međutim dovodi do visoke varijance u rezultatu, jer izostavlja puno ostalih puteva kojima se moglo proći. Ono što se dobiva periodičkom analizom jer mogućnost beskonačne operacije i smanjuje se varijanca procjene. To je moguće jer se ne koristi zadnje stanje puta gdje je vrijednost 0 već se u zadnjem stanju jedne procjene uzima procjena vrijednosti tog stanja kao točka od koje se računa. Sada kad se sumiraju nagrade unazad do kraja prošle procjene dobiju se "prave" vrijednosti stanja. Te vrijednosti su za kritičara prave međutim one se odnose na njegove procjene tako da se time unosi pristranosti u rezultat analize, ali i mogućnost beskonačnog izvođenja.

Prilikom izbora akcije vođa iznosi postotke pouzdanosti u pojedine akcije i uglavnom se odabere ona se najvećom pouzdanosti, ali radi omogućavanja AIu da eksperimentira, nekad odabir nije onaj s najvećim postotkom i unosi se entropija u postotke pouzdanosti. Time AI nekad odabere naizgled manje idealnu opciju koja ispadne da donese bolje nagrade. Kada se AI nađe u negativnoj situaciji zbog svojih odluka A2C regulira analizu tako da se može izvući najbolje iz loše situacije. Ako se procjeni da stanje donosi negativnu nagradu, ali nakon akcije

ispadne da je nagrada bila pozitivnija nego očekivano, vođa nauči da akciju koju je odabrao i koja je donesla negativnu nagradu nije loša već zapravo dobra. To je u suprotnosti s modelom temeljenom na pristranosti, jer taj model nema pomoć kritičara za procjene i on svaku akciju s negativnom nagradom smatra lošom i smanjuje vjerojatnost da će se ta akcija opet izabrati. Iz ove situacije je naučio i kritičar da je krivo procijenio negativnost situacije tako da i on može u ovoj negativnoj situaciji poboljšati svoje procjene.

AI koji radi na *Reinforcement learning* algoritmu mora se jedino izložiti podražajima vanjskim podražajima koje može osjetiti i od tog momenta on će se početi razvijati i učiti. Od te točke dalje on se može samostalno nastaviti učiti prema podražajima koje dobije u smjeru na koji mu mi odredimo što je se boduje kao nagrada. Iskustvom postaje bolji i precizniji u procjeni nagrade stanja, uvjereniji u svoje izbore akcija i rjeđe će se naći u situaciji gdje su ga iznenadili rezultati akcije.

4. Korištenje Pwnagotchija

U ovom poglavlju obuhvatit će se sve procesi koji su izvedeni od instalacije do korištenja i učenja Pwnagotchija. To uključuje: stvari potrebne za instalaciju Pwnagotchija, proces instalacije, početno podešavanje, učenje korištenja Pwnagotchija, plugini koji su korišteni i kako, te kako se učilo Pwnagotchija da skuplja rukovanja.

4.1. Instalacija

Proces instalacije objašnjen je na internetu na Pwnagotchi stranici, međutim proći će se kroz proces jer se neke stvari razlikuju od službenog vodiča. Kako bi se uopće moglo razmišljati o instalaciji potrebno je nabaviti Raspberry Pi, gdje to može biti 3, 4 ili ZeroW, a za ovaj projekt odabran je RPi0W kao što je preporučeno na stranici. Za započeti raditi sa RPi0W potrebno je nabaviti kvalitetnu SD karticu koja će služiti kao tvrdi disk za RPi i mora biti kvalitetna, jer od mnogo pisanja i brisanja može postati neiskoristiva, što ne želimo da nam se dogodi na početku projekta [8]. Na SD karticu se zaprži slika u .iso formatu koja se prethodno treba skinuti s GitHub stranice za Pwnagotchi. Skinuta je verzija 1.5.3 koja nije bila najnovija verzija u tom trenutku, jer postoji problem da 1.5.5 verzija ne radi ispravno ako se započne instalacija s njome. Za zapržiti .iso sliku na SD karticu potrebno je koristiti jedan od programa specijaliziranih za to, a iskorišten je bio besplatni BalenaEtcher program. Prilikom korištenja postoji jedan detalj, a to je da program nije dao uspješnu poruku nakon što je zapržio sliku na SD karticu, ali je ipak sve ispravno napravio što je očito neki problem u samom programu. Potrebno je napomenuti da treba imati adapter za čitanje SD kartice, jer je ona mala a čitači kartica imaju drugačije sučelje od same SD kartice, u ovom projektu je za čitanje korišten ugrađeni čitač kartica od vlastitog laptopa. Potrebno je odraditi još jedan korak prije nego se SD kartica može prebaciti u RPi0W, a to je dodati inicijalnu konfiguracijsku datoteku u korijenski direktorij SD kartice koja se prikazuje na računalo kao disk particija. Datoteka koja sadrži konfiguraciju je .toml formata i ja sam u nju upisao sljedeće podatke koji su jako slični preporučenima:

```
main.name = "pwnagotchi"
main.lang = "en"
main.whitelist = []

main.plugins.grid.enabled = true
main.plugins.grid.report = true
main.plugins.grid.exclude = []

ui.display.enabled = true
ui.display.type = "waveshare_2"
ui.display.color = "black"
```

Ovaj korak je vrlo važan zato što ako se propusti ubaciti .toml datoteka s konfiguracijom onda se Pwnagotchijev operacijski sustav sa SD kartice neće pravilno instalirati i trebat će ponoviti proces stavljanja slike na SD karticu.

U ovom trenutku SD kartica je spremna za ubacivanje u RPi0W. Nakon što se ubaciti SD kartica u njezino ležište potrebno je spojiti RPi na izvor energije. Za to može se koristiti jedna od navedenih baterija iz službenog vodiča za Pwnagotchi koja se može spojiti lemljenjem na pločicu od RPija ili USB kabelom na jedan od portova na RPiju. Korišteni su bili USB portovi i spojen je RPi na računalo, te tada je počela instalacija. Potrebno je pričekati par minuta prije nego se Pwnagotchi instalira i pokrene na RPiju i ako nam je spojen na računalo preko Data USB porta onda ga naše računalo registrira kao Ethernet USB uređaj. Iznimno je važno da se ne isključuje RPi iz struje dok vrši instalaciju.

4.2. Konfiguracija Pwnagotchija

Nakon prethodnog koraka instalacije Pwnagotchi je potpuno spreman za započeti skupljati WPA rukovanja, međutim ako želimo imati kontrolu nad pluginima koje koristi ili mijenjati nešto u konfiguraciji potrebno je napraviti nekoliko koraka da si to omogućimo. Pwnagotchi nema konekciju na internet preko svog "tijela" RPija, nego može pristupiti internetu preko dvije točke: računala ili mobitela. Pristupanje internetu preko računala ima dvojaku ulogu, dobivanje interneta, ali i spajanje na Pwnagotchi SSH vezom koja omogućuje izmjenu, instalaciju i dodavanje datoteka. Preko mobitela korištenjem bluetootha može se podijeliti internet i vidjeti Pwnagotchijevo web sučelje koje se može vidjeti i s računala. S mobitela se u teoriji mogu slati datoteke, ali je jako nepraktično i nitko od drugih korisnika niti u ovom projektu nije isproban taj način.

4.2.1. Spajanje s računalom i mobitelom

Prvo će se opisati postupak potreban da se ostvari veza s računalom. Kod ove veze iako u vodiču piše da se može podijeliti internet, ipak nije bilo moguće što se može vidjeti iz toga što više drugih korisnika isto nije uspjelo u tome nego su preporučili preko mobitela podijeliti internet Pwnagotchiju. Razlog zašto je bitno napraviti ovaj postupak je zato što će omogućiti korištenje SSH veze. U mrežnim postavkama gdje su prikazana sva mrežna sučelja potrebno je naći novo Ethernet sučelje koje je vrste *USB Ethernet/RDNIS Gadget* i u IPv4 postavkama postaviti fiksnu IP adresu na "10.0.0.1", *Subnet mask* na "255.255.255.0" i može se probati postaviti *Preferred DNS Server* na "8.8.8.8" iako to nema toliko uloge pošto se neće ostvariti veza prema internetu. Nakon toga može se SSH vezom spojiti se na Pwnagotchi ciljajući na adresu 10.0.0.2 s korisničkim imenom "pi" i šifrom "raspberry", što su početne vrijednosti koje se kasnije mogu promijeniti i preporučeno ih je promijeniti.

Ostvarivanje veze s mobitelom je jednostavnije, ali potrebna je izmjena datoteke konfiguracije. Zbog te činjenice potrebno je imati ostvarenu konekciju s računalom prvo ili postaviti sve postavke u datoteku konfiguracije na SD kartici prije nego je instaliran Pwnagotchi. Prije pokušaja spajanja treba podesiti nekoliko stvari na mobitelu. Uređaj mora biti vidljiv drugim bluetooth uređajima i mora biti uključena opcija dijeljenja internet veze preko bluetootha. To bi većina novijih mobitela trebala imati tako da vjerujem da to nije problem za većinu korisnika Pwnagtochija. Tada putem računala i prije putem konfiguracijske datoteke treba ubaciti sljedeće

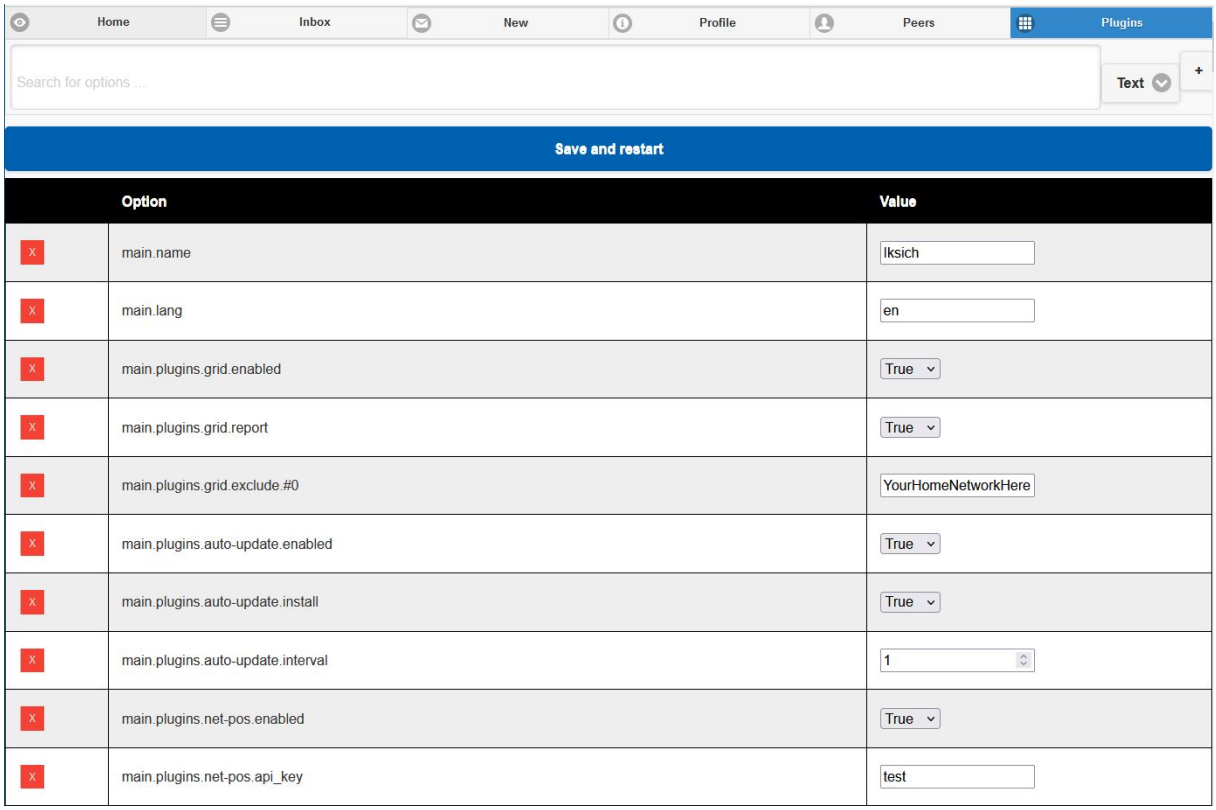
redove u config.toml datoteku kako bi Pwnagotchi mogao komunicirati s Android mobitelom:

```
main.plugins.bt-tether.enabled = true
main.plugins.bt-tether.devices.android-phone.enabled = true
main.plugins.bt-tether.devices.android-phone.search_order = 1
main.plugins.bt-tether.devices.android-phone.mac = "94:D0:0D:16:0B:25"
main.plugins.bt-tether.devices.android-phone.ip = "192.168.44.44"
main.plugins.bt-tether.devices.android-phone.netmask = 24
main.plugins.bt-tether.devices.android-phone.interval = 1
main.plugins.bt-tether.devices.android-phone.scantime = 10
main.plugins.bt-tether.devices.android-phone.max_tries = 0
main.plugins.bt-tether.devices.android-phone.share_internet = true
main.plugins.bt-tether.devices.android-phone.priority = 99
```

Ovo su postavke za Android mobitele, a za iOS nisu stavljane pošto to nije isprobano, iako se razlikuju samo u IP adresi i naravno MAC adresi. Kod ovih postavki potrebno je ubaciti MAC adresu svoga uređaja u polje *main.plugins.bt-tether.devices.android-phone.ip*, to polje nije teško naći, ali vjerojatno će način dolaska do te informacije varirati između modela i proizvođača mobitela. Nakon ponovnog pokretanja Pwnagotchija trebao bi se pojaviti zahtjev za uparivanje na mobitelu i nakon potvrde konekcija bi trebala biti ostvarena. U ovom projektu nije bilo tako već je bilo potrebno napraviti nekoliko dodatnih koraka. Preko SSH veze s programom PuTTY koji je program komandne linije bilo je potrebno ući u modul koji radi s bluetooth sučeljima i ručno skenirati uređaje koje detektira. Nakon pronalaska MAC adresu mobitela bilo je potrebno nad tom adresom izvesti naredbu za spajanje s tim uređajem i tek tada dobiva se poruku za uparivanje na mobitelu. Potrebno je bilo i izvesti naredbu da Pwnagotchi vjeruje mobitelu kako ne bi bilo problema s internet vezom. Nakon postavljanja tih parametara i izvođenja potrebnih korake nije više bilo potrebno ponavljati ih, jer Pwnagotchi periodički provjerava detektira li poznate bluetooth uređaje i ako da automatski se spoji na njih.

4.2.2. Izmjena konfiguracije

Nakon spajanja s računalom izmjene u konfiguraciji moguće je raditi na dva načina. Prvi način je direktna izmjena u config.toml datoteci korištenjem PuTTY programa i pisanjem putem nano naredbe direktno u datoteku ili otvaranjem datoteke putem WinSCP programa i mijenjanjem datoteke. Ovaj način je dobar kada se koristi dio nečije tuđe konfiguracije pa je lakše samo kopirati i zalijepiti te linije u datoteku. Drugi način izmjene je način koji je ovdje češće korišten, a to je preko web sučelja i jednog plugina. Važno je napomenuti da je preko web sučelja moguće mijenjati samo config.toml datoteku konfiguracije, ostale datoteke potrebno je mijenjati preko prve metode koje su ovdje navedena. Za promijene u konfiguracijskoj datoteci lakše mi je bilo koristiti web sučelje jer ima filtriranje opcija prema imenu i ograničuje korisnika da unese podatke u pravilnom obliku (string, int ili bool). Mijenjanje konfiguracije za pojedine plugine moguće je preko sučelja raditi tek kad se uključe plugini, jer tek se onda učitaju u config.toml datoteku konfiguracijske linije iz njihovih .toml datoteka.



Slika 5: Webcfg plugin (Izvor: Osobna izrada)

Na slici je prikaz *webcfg* plugina koji omogućuje izmjene u config.toml datoteci. Lakše je koristiti ovo sučelje nago ručno mijenjati putem PuTTYa ili WinSCPa. Još jedna prednost ovog sučelja je što se može koristiti preko mobitela. To je posebno dobro kada treba nešto promijeniti dok se šeta sa Pwnagotchijem daleko od laptopa. Negativna strana toga je što za neke promjene treba ponovno pokrenuti Pwnagotchija i ako je bio u AI načinu rada onda će opet trebati proći 20 minuta da se prebaci iz AUTO načina na AI način rada.

4.3. Pwnagotchi načini rada

Pwnagotchi ima više načina operacije: MANU, AUTO i AI. AUTO i AI načini rada su dosta slični i vezani su za hvatanje WPA rukovanja, a MANU način rada vezan je za period održavanja Pwnagotchija.

4.3.1. MANU

Ovaj način rada aktivira se kad je Pwnagotchi spojen Data USB kablom na računalo i pretpostavlja se da će se nad Pwnagotchijem raditi "održavanje". Održavanje nije u principu potrebno, jer bi u suprotnom prestao raditi nego je onda omogućen prijenos datoteka i funkcije koje se događaju u pozadini ili čekaju znak da je MANU način rada. Neke funkcije koje bi stvorile veliki napor da rade s hvatanjem rukovanja pokreću se tek u ovom načinu rada. U ovom načinu rada pokreću se funkcije koje šalju podatke na web stranice ili skidaju podatke s

web stranica. U ovom načinu rada ne skupljaju se WPA rukovanja niti se bilo kako prati WiFi.

4.3.2. AUTO

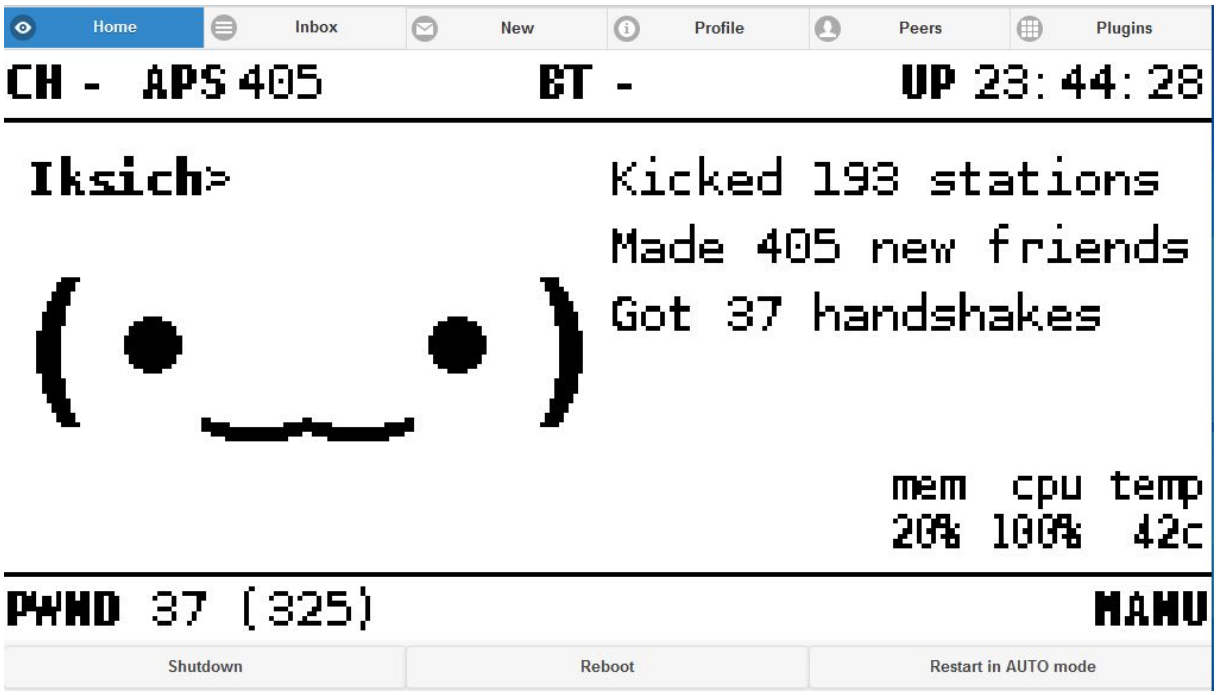
AUTO način rada je u suštini osnovna Pwnagotchijeva funkcija, skupljanje WPA rukovanja. Ovaj način rada aktivan je kada se Data USB kabelom spoji na USB port zadužen za napajanje ili kada se prijenosna baterija spoji na bilo koji od dva USB porta RPija. U ovom načinu rada Pwnagotchi neselektivno prolazi kroz sve kanale WiFija i odspaja uređaje i svim uređajima šalje PMKID napade.

4.3.3. AI

Kada je AI aktiviran i Pwnagotchi provede otprilike 20 minuta u AUTO načinu rada napokon se učitaju neuralne mreže potrebne za opsluživanje A2C *Reinforcement learning* algoritma kojeg koristi Pwnagotchi. Tada Pwnagotchi počinje koristiti svoja "sjećanja" koja je dosad napravio i selektivnije počne skupljati WPA rukovanja. Iskustvo koje Pwnagotchi ima formira se u obliku naklonosti određenim vrijednostima radnih parametara u odnosu na obilježja okoliša koje osjeća oko sebe. Radni parametri prema kojima Pwnagotchi radi objašnjeni su u potpoglavlju "Učenje Pwnagotchija".

4.4. Vizualno sučelje

Pwnagotchi je zamišljen da se koristi s malim Waveshare eInk 2.13 inčnim ekranom koji se spaja na RPi0W ili s drugim eInk ekranima. Oni su izabrani zbog male potrošnje električne energije. Na tom ekranu prikazuje se Pwnagotchijevo lice i osnovne informacije uz dodatak drugih potencijalno važnih informacija uz prethodnu aktivaciju plugina. Prilikom korištenja Pwnagotchija nije bio korišten ekran već je korišteno web sučelje za prikaz svih potrebnih informacija. Kreator Pwnagotchija omogućio je da ljudi mogu koristiti Pwnagotchi bez ekrana tako da sve što se prikazuje na ekranu, prikazuje se i na web sučelju uz neke dodatne opcije.



Slika 6: Web sučelje (Izvor: Osobna izrada)

Na slici koja je izrezana iz slike ekrana mog laptopa vidimo web sučelje uz dodatak gornje trake s opcijama i donje trake s opcijama, tih traka nema na prikazu preko eInk ekrana. Počevši od gornjeg lijevog kuta objašnjavat ću stvari od gore prema dolje i s lijeva na desno. Prije objašnjavanja ostalih oznaka važno je napomenuti da je na slici Pwnagotchi u MANU načinu rada kao što se vidi prema oznaci dolje lijevo na slici. Oznaka "CH" prikazuje koji WiFi kanal Pwnagotchi trenutno pretražuje, a kako je sada u MANU načinu rada ne vrši se pretraživanje kanala. Oznaka "APS" prikazuje koliko je Pwnagotchi do sada osjetio WiFi APa, u slučaju da je u AUTO ili AI načinu rada tu se onda prikazuje broj APa u trenutnoj sesiji. Vezano za sesije isto vrijedi za broj uhvaćenih rukovanja i šifri. Oznaka "BT" pojavljuje se kad je aktiviran plugin za bluetooth i na slici prikazuje da preko bluetootha Pwnagotchi nije spojen na niti jedan uređaj. Sljedeća oznaka je "UP" što daje podatak o vremenu koliko je Pwnagotchi bio aktivan. U mome slučaju taj podatak nije ispravan, jer ono što računala i mobiteli imaju, a Pwnagotchi nema je unutarnji sat. Zato vremena koja Pwnagotchi pridodaje svojim akcijama i drugim stvarima u mojem slučaju nisu točna. Moguće je kupiti i ugraditi unutarnji sat u Pwnagotchi, ali to nije učinjeno, a da je podaci o vremenu bili bi ispravni. Oznaka "Iksich" je ime koje korisnik može dati svome Pwnagotchiju gdje je prema vlastitom izboru odabrano Iksich. Tim imenom se prijavljuje na neke servise gdje se bilježi njegove akcije. Ispod imena nalazi se lice Pwnagotchija koje nam sugerira kako se on "osjeća" i može iz njega vidjeti koja se akcija trenutno izvodi. Lice prikazuje događaje što su momenti u vremenu obilježeni nekom akcijom Pwnagotchija i prikazuje trenutno stanje Pwnagotchija koje obuhvaća jedan period vremena. Pwnagotchi ima lica koja su: sretna, jako sretna, ravnodušna, tužna i jako tužna. Ovi tipovi lica odnose se na okolinu gdje je Pwnagotchi sretniji što je više APova u blizini i tužniji ako ne detektira nove APove. Postoji lice za detekciju APa, obavljanje napada, hvatanje rukovanja i slanje podataka na server. Desno od lica pojavljuju se poruke o trenutnim akcijama Pwnagotchija koje

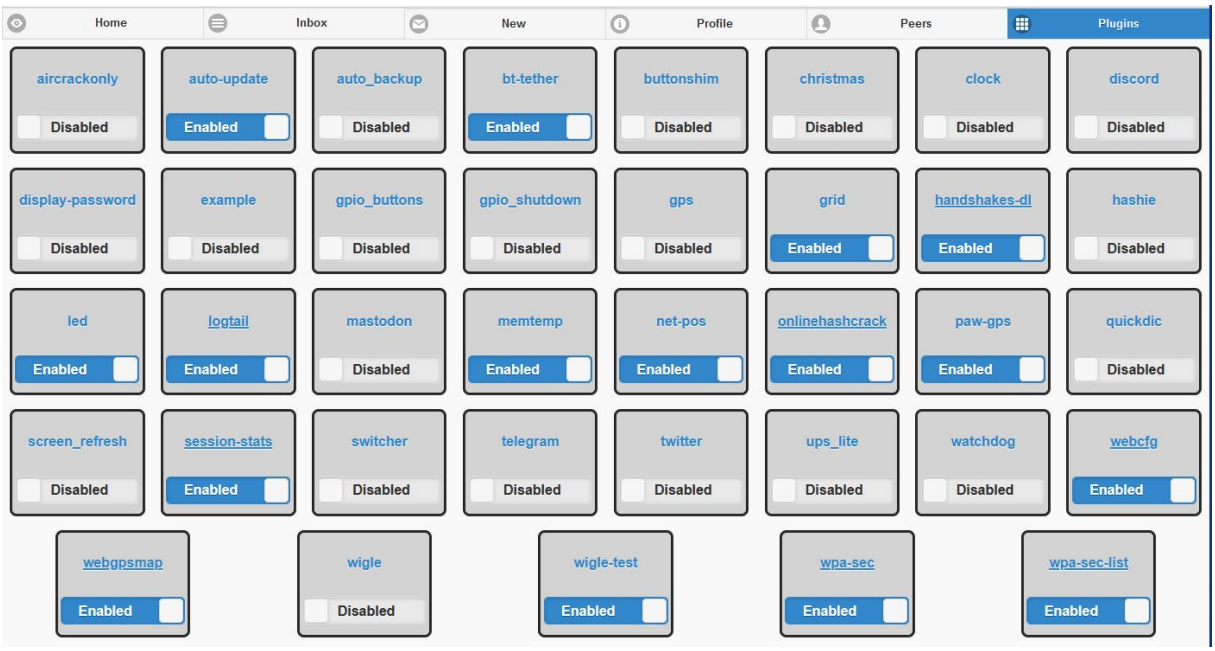
prate lica, ali lakša su za razumjeti dok se korisnik bolje ne upozna sa svojim Pwnagotchijem. Poruka koja se vidi na slici glavna je poruka za MANU način rada gdje se prikazuje broj "šutnutih" APa, broj registriranih APa i broj uhvaćenih rukovanja. Broj rukovanja odnosi se na zadnju sesiju, a ostali brojevi odnose se na kompletni životni vijek Pwnagotchija. U donjem lijevom kutu ponavlja se podatak o uhvaćenim rukovanjima u prošloj sesiji i unutar zagrada stoji broj uhvaćenih tijekom cijelog životnog vijeka Pwnagotchija. Iznad oznake za način rada što je na slici MANU, stoje dodatni podaci koje sam ja samostalno uključio i odnose se na opterećenje na hardver Pwnagotchija tj. koliko se radne memorije koristi, koliko se snage procesora koristi i kolika je temperatura SoCa ili glavnog procesora RPija.

Sve podatke navedene gore mogu se vidjeti i na eInk ekranu koji bi bio spojen na RPi pločicu. Web sučelje nudi ove opcije koje stoje u gornjoj i donjoj traci. Web sučelju korisnik može pristupiti tako da je spojio računalo ili mobitel s Pwnagotchijem ovisno s kojeg će uređaja pristupati. Proces s računala ili mobitela je isti korisnik mora u web preglednik upisati adresu *http://.local:8080*, gdje je ime Pwnagotchija ono koje mu korisnik postavi u config.toml datoteci. Počevši od donje trake s lijevo na desno prva je tipka za gašenje Pwnagotchija. Ona postoji da se "lagano" ugasi sustav iako nema nikakvih posljedica ako se RPi direktno odspoji s izvora struje. Korisna je tipka "Reboot" koja ponovno pokreće sustav u istom načinu rada, a u ovom projektu je služila najviše dok su testirani plugini koje vlastite izrade. Najdesnija tipka mijenja se ovisno o trenutnom načinu rada Pwnagotchija. Ako se nalazi u MANU načinu rada kao na slici onda ona nudi ponovno pokretanje i početak rada u AUTO načinu rada, a ako je Pwnagotchi u AUTO načinu rada tipka nudi ponovno pokretanje i početak rada u MANU načinu. Korisna je za testiranje, jer nakon izmjene postavki u MANU načinu rada samo se pritisne tipka i prebaci se način rada i kada se želi nakon izleta sa Pwnagotchijem poslati podatke o uhvaćenim šiframa na stranice koje smo odredili. Te stranice mogu biti Pwnagotchijeva stranica za prijavu uređaja i detektiranih mreža, Wigle.net stranica za prijavu detektiranih APova, te stranice OnlineHashCrack i Wpa-sec koje imaju mogućnost probijanja dostavljenih šifri.

Gornja traka s opcijama je izbornik s dodatnim web prikazima za administraciju i dodatne opcije Pwnagotchija. Pritiskom na jedan od opcija na traci prebacuje korisnika na druge stranice web sučelja od kojih su korištene samo dvije: "Home" i "Plugins". Home stranica je ona koja je prikazana slici i prikazuje lice i druge podatke. Stranice "Inbox" i "New" vezane su za mogućnost koja je ugrađena u Pwnagotchi, a to je da može služiti kao e-mail klijent za primanje i slanje e-mailova. Inbox prikazuje primljenu poštu, a New otvara formu za slanje nove pošte. Tu funkcionalnost nije isprobana, jer se ovaj Pwnagotchi nije mogao prijaviti na Pwn poslužitelj zadužen za slanje i primanje pošte koja ide preko Pwnagotchija. Stranica "Profile" sadrži osnovne podatke o Pwnagotchiju kao što je njegovo ime, jedinstveni identifikator, verziju i QR kod. Stranica "Peers" prikazuje podatke o drugim Pwnagotchijima koje je Iksich sreo na svojim putovanjima, nažalost nije sreo drugog Pwnagotchi uređaja tijekom korištenja. "Plugins" je stranica za administraciju plugina koji dođu s Pwngotchijem ili koji se dodatno instaliraju i sadrži listu plugina koji se mogu aktivirati, te koji daju objašnjenje svoje funkcije.

4.5. Korišteni plugini

Plugini za Pwnagotchi su zasebno pakirane Python skripte koje se vežu za Pwngotchijeve funkcije tako što reagiraju na propagacije određenih poruka koje Pwnagotchi oglasi prilikom izvođenja određenih akcija. Na primjer Pwngotchi propagira poruku da se uhvatilo rukovanje ili da se upravo pokrenuo sustav ili kada se Pwnagotchi spojio na internet. Pwnagotchi u svojem osnovnom korištenju samo hvata WPA rukovanja i zapisuje ih, a može to činiti automatski u AUTO načinu rada ili pametno u AI načinu rada. To je donekle složeno gledajući iz perspektive koji trud je trebalo uložiti da sustav radi kako treba, jednome korisniku koji instalira Pwnagotchi na RPi se neće činiti da ovaj sustav radi nešto iznimno posebno i da obavlja mnogo funkcija. Zato je kreator s nekoliko ljudi koji čine uži razvojni tim napravio više plugina koji dolaze sa Pwngotchi sustavom po instalaciji, ali nisu automatski uključeni, nego ih korisnik mora izričito uključiti. Na službenoj stranici i na GitHub stranici piše da su ti plugini dobro testirani od strane kreatora i da dobro rade. S druge strane kreator je dao mogućnost ostalim ljudima da naprave svoj plugin koji neće biti tako rigorozno testiran i imat će svoj zasebni GitHub repozitorij. Ti plugini jesu navedeni na službenoj stranici, ali korisnik ih mora posebno skinuti na svoj Pwnagotchi. Proces za uključiti te plugine koje je napravila zajednica je da se preko komandne linije iz Pwnagotchijevog sustava treba klonirati taj repozitorij s GitHuba i u config.toml ubaciti direktorij sa skinutim pluginima. Taj direktoriji također je najbolje mjesto za staviti plugine koje je korisnik sam razvio i u procesu je testiranja ih.



Slika 7: Plugini (Izvor: Osobna izrada)

Na slici vidi se prikaz stranice "Plugins" gdje su navedeni svi plugini koje Pwnagotchi prepoznaje i koji se mogu uključiti. Plugini koji dolaze prilikom instalacije odmah su prikazani ovdje kada se Pwnagotchi prvi put upali, a ostali se pojave kada i ako se u config.toml datoteci točno navede direktorij s drugim pluginima. Raspored plugina je abecedan i time su pomiješani plugini koji su originalni i oni koji su od zajednice. Klizeća tipka u kućici plugina ispod naziva

pokazuje je li plugin upaljen i pritiskom na nju mijenja se stanje plugina. Plugini koji su upaljeni imaju podcrtano ime jer ako se lebdi iznad imena s mišem dobiti ćemo kratko objašnjenje funkcije plugina i ako taj plugin koristi svoju zasebnu stranicu u web pregledniku onda će nas pritiskom na ime plugina odvesti na tu stranicu. Na slici vidi se da je navedeno mnogo plugina međutim objasnit će se funkcije samo onih koji su korišteni i kakva su bila iskustva s korištenjem tih plugina. Neke plugine nije bilo moguće koristiti jer odgovarajući hardver nije bio raspoloživ kao: "screen_refresh", "ups_lite", "gpio_buttons" i "gps". Neki plugini nisu korišteni, jer su bili ili prejednostavni ili nije bilo svrhe koristiti ih.

4.5.1. Auto-update

Ovaj se plugin aktivira kad je Pwnagotchi u MANU načinu rada i kada postoji veza na internet. Tada on automatski provjerava ako postoji nova verzija Pwnagotchija i u slučaju da postoji automatski skine i instalira novu verziju. Tijekom korištenja nije došla nova verzija tako da se u principu nije uspio iskoristiti ovaj plugin, ali svejedno bio je ostavljen upaljen.

4.5.2. BT-tether

BT-tether je plugin koji je objašnjavan prije što radi, a to je omogućavanje spajanje Pwnagotchija s mobitelom. Ovo je originalni plugin koji dolazi sa Pwngotchijem i korištenje njega je opcionalno, međutim u ovom slučaju korišten je stalno jer je bio jedini način za davanje interneta Pwnagotchiju, a i kad se išlo u šetnju sa Pwnagotchijem korišteno je web sučelje preko mobitela za pratiti što Pwnagotchi radi.

4.5.3. Grid

Grid je plugin koji dolazi sa Pwnagotchijem i ako se koristi config.toml datoteka za početnu konfiguraciju kako sam naveo u potpoglavlju s instalacijom onda će ovo biti jedini aktivni plugin kada se prvi put upali Pwnagotchi. On služi za prijavu Pwngotchija na interni registar svih Pwnagotchi sustava koji su aktivni ili su u jednom trenutku bili aktivni. Ima opciju u svojoj konfiguraciji da se uz identifikator Pwnagothcija pošalju i APovi od kojih je Pwnagotchi pokupio WPA rukovanja, ali bez dijelova sa šifrom. Tijekom korištenja ovaj plugin je iz nekog razloga javljao grešku kada bi se išao prijaviti na registar, a greška se nije uspjela otkriti, tako da je ovaj Pwnagotchi ostao neprijavljen.

4.5.4. Handshakes-dl

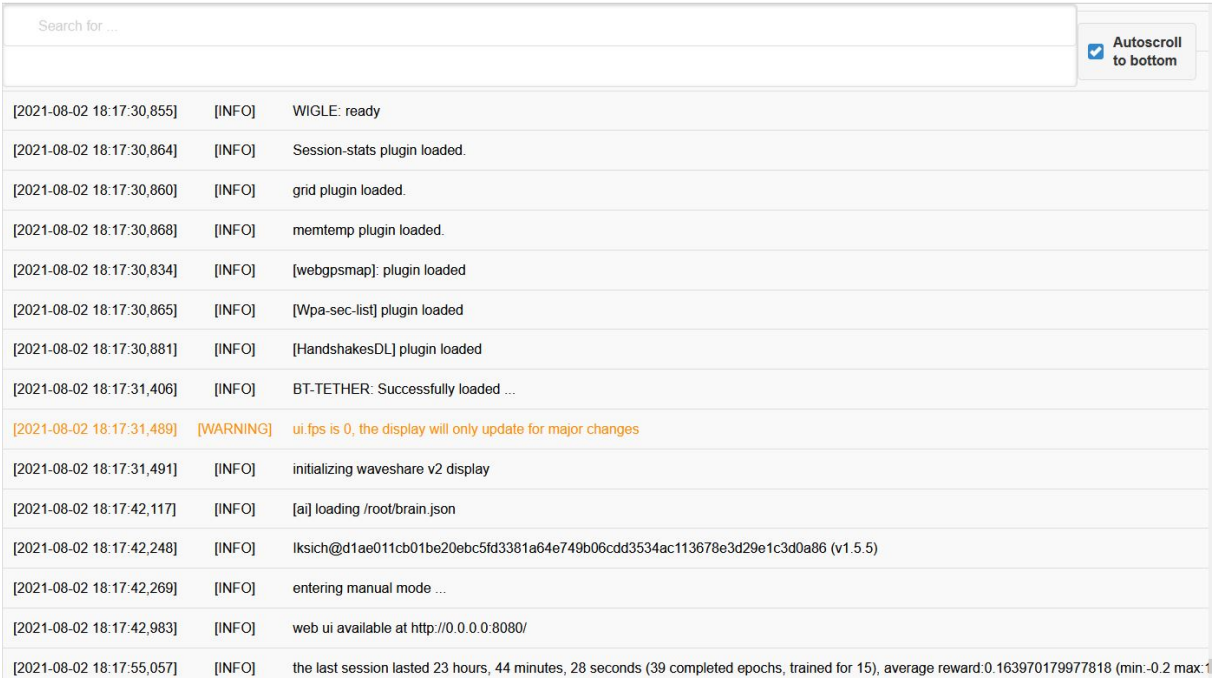
Handshakes-dl je plugin za preuzimanje .pcap datoteka sa Pwnagotchija na računalo ili mobitel. Taj plugin ima svoju stranicu gdje stvori listu svih .pcap datoteka koje zapravo predstavljaju listu svih uhvaćenih rukovanja. Klikom na liniju u tablici pokreće se preuzimanje .pcap datoteke na računalo.

4.5.5. LED

LED plugin prenosi trenutna događanja u sustavu Pwnagotchi tako da na određene načine pali i gasi jednu LED lampicu na RPi0W pločici. Ovaj plugin je korišten kako bi dobili vizualni znak kada je Pwngotchi nešto napravio, te se onda išlo pogledati što se dogodilo u log datoteku.

4.5.6. Logtail

Ovaj plugin je bio jedan od najkorisnijih plugina koji je korišten sa Pwnagotchijem. Dolazi sa Pwnagotchijem prilikom instalacije i služi za pregled log datoteke. Važan je jer se u log datoteci zapisuju sve akcije koje Pwnagotchi obavlja u AUTO i AI načinu rada. To bi značilo da je bilo moguće uživo pratiti što i kako Pwnagotchi radi da sakupi što više WPA materijala. Osim za tu svrhu u log datoteku zapisuju se poruke kada plugin uđe u određeni dio svoga izvođenja, naravno ako je plugin tako programiran. Ova činjenica je važna, jer je to bio jedini način za raditi debug na pluginima koji su bili poboljšavani i razvijani.



Slika 8: Logtail (Izvor: Osobna izrada)

Prikaz na slici sadrži standardne poruke koje se javljaju kada se Pwnagotchi pokrene u MANU načinu rada. Većina plugina je programirano da pokaže poruku kada se učitaju da znamo da su se pravilno učitali i da možemo računati da će započeti svoje akcije. Na slici se još vidi poruka na kojoj adresi je dostupno web sučelje, trenutna verzija i identifikator ovog Pwnagotchija, te nešto podataka o prošloj sesiji.

4.5.7. Memtemp

Ovo je mali plugin koji prikazuje dodatne podatke o stanju Pwnagotchija tj. o njegovom hardveru. Kada se uključi ovaj plugin on pokazuje podatke o zauzeću memorije, procesora i temperaturu čipa, kao što je objašnjeno na prikazu web sučelja.

4.5.8. Net-pos

Net-pos je plugin koji reagira na hvatanje rukovanja i u tom trenutku sprema poziciju Pwnagotchija. Pozicija se u trenutku hvatanja rukovanja sprema u obliku .net-pos.json datoteke koja nema geografske koordinate za lokaciju već sadržava listu WiFi APa gdje se AP identificira sa svojom MAC adresom i uz MAC adresu stoji jačina signala od tog APa. Primjer datoteke:

```
{"wifiAccessPoints": [{"macAddress": "3c:a6:2f:fb:5e:e3", "signalStrength": -71}, {"
    macAddress": "78:94:b4:36:82:82", "signalStrength": -76}, {"macAddress": "3c:f7:
    a4:01:de:37", "signalStrength": -77}, {"macAddress": "0c:8e:29:a6:7e:a9", "
    signalStrength": -77}, {"macAddress": "64:d1:54:16:c1:4f", "signalStrength":
    -78}, {"macAddress": "64:6e:ea:31:e0:a9", "signalStrength": -78}], "ts":
    1627227487}
```

Ova vrsta datoteke nije korisna za lociranje pozicije na karti, nego kada se dobije internet veza ta datoteka pošalje na obradu u *Mozilla Location Services* i preuzme se datoteka s korisnim lokacijskim informacijama. Ovaj plugin se oslanja na projekt od Mozille koji koristi lokacije WiFi APa koje su poslani Mozilli iz druge lokacije da uz pomoć metode triangulacije nađe lokaciju koju tražimo. Prema jačini signala od APova za koje je poznata lokacija može se uz razumljivo odstupanje pretpostaviti lokacija Pwnagotchija u danom trenutku. Datoteka s lokacijskim podacima koja se dobije je ekstenzije .geo.json i uz pravilnu obradu korisnik je može ubaciti u neku od aplikacija za karte i dobiti prikaz lokacije AP od kojeg je uhvaćeno rukovanje na karti. Ove podatke o lokaciji korišteni su uz webgpsmap plugin koji služi za trenutni prikaz uhvaćenih rukovanja na karti.

4.5.9. OnlineHashCrack

Ovaj plugin je jedan od dva plugina preko kojih se može izvući i probiti šifra koja se izvuče iz .pcap datoteke. OnlineHashCrack kada je uključen čeka da se Pwnagotchi prebaci u MANU način rada i da postoji veza na internet, te tada automatski pošalje .pcap datoteke koje još nije poslao na API točku od OnlineHashCrack stranice. Kasnije u radu je detaljnije objašnjen rad ovog plugina i mogućnosti.

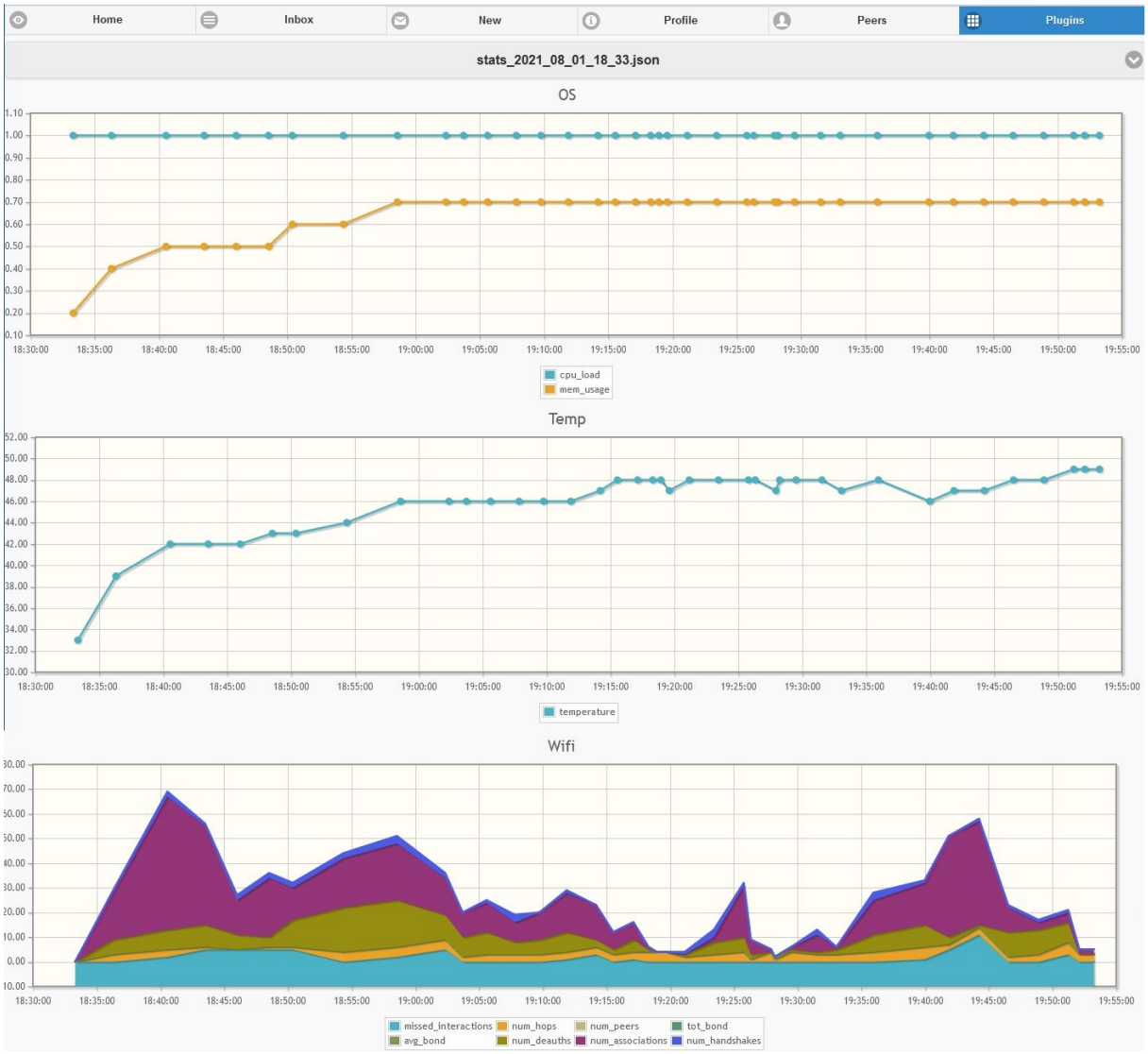
4.5.10. Paw-gps

Paw-gps plugin koristi Pwnagotchijevu vezu s mobitelom da stvori datoteku s lokacijom Pwnagotchija u trenutku hvatanja rukovanja. Datoteka koja se stvori je ekstenzije .paw-gps.json i uz malo ili ništa obrade korisnik je može iskoristiti za lociranje APa od kojeg je uhvaćeno rukovanje na karti. Ovaj plugin koristi se aplikacijom *PAW Server* koju korisnik mora instalirati

na mobitelu da dobije podatke o lokaciji. Rad ovog plugina detaljnije je objašnjen kasnije u dijelu rada vezanog za poboljšanje wigle plugina. Lokacijski podaci korišteni su sa webgpsmap pluginom.

4.5.11. Session-stats

Ovaj plugin služi za prikaz podataka o pojedinoj sesiji na nekoliko grafova gdje svaki graf služi za prikaz drugog tipa podataka. Na slici ispod uzeo sam samo tri grafa koja prikazuju koja su opterećenja hardvera, temperaturu čipa i WiFi aktivnost. Na vrhu stranice mora se odabrati koja će se sesija promatrati, a one su označene prema datumu kada je sesija počela. Najzanimljiviji graf nam je WiFi graf koji pokazuje nekoliko stvari od kojih će navesti najvažnije: tamnoplavo je označen broj uhvaćenih rukovanja, u ljubičastom je broj PMKID napada i u zelenom je broj napada odspajanjem.



Slika 9: Logtail (Izvor: Osobna izrada)

4.5.12. Webcfg

Ovo je drugi jako koristan plugin uz Logtail koji je dosta korišten sa Pwnagotchijem. Webcfg omogućava korisniku da mijenja config.toml datoteku tj. da mijenja konfiguraciju Pwnagotchija preko web sučelja. To olakšava proces izmjena u konfiguraciji, jer se može raditi i preko mobitela, a i zato što ima tražilicu prema imenu konfiguracijske linije. Webcfg ima svoju zasebnu stranicu na koju se dolazi pritiskom na ime plugina i formatirana je u obliku liste konfiguracijskih linija koje stoje u config.toml datoteci. Slika prikaza webcfg stranice stoji u potpoglavlju "Izmjena konfiguracije" pa se neće opet stavljati.

4.5.13. Webgpsmap

Webgpsmap je plugin koji prikazuje lokaciju AP od kojeg je uhvaćeno rukovanje na *OpenStreetMap* karti. Za prikaz koristi .geo.json, .gps.json ili .paw-gps.json datoteke koje sadrže lokacijske podatke. Karti se pristupa s web sučelja što znači da se može gledati s mobitela i računala. To je prednost ovog plugina da najlakše prikazuje APove na karti, jer inače korisnik mora poslati podatke na wigle.net ili ih samostalno skinuti i prikazati na karti. Prikaz webgpsmap stranice gdje se vide APovi koji su otkriveni na križanju Prolaza Venecije i Ulice Gradska Vrata u Novigradu:

Slika 10: Webgpsmap (Izvor: Osobna izrada)

4.5.14. Wigle

Wigle plugin šalje lokacijske podatke na wigle.net stranicu. Ta stranica je repozitorij za lokacije WiFi APova, bluetooth uređaja i odašiljača mobilnog signala. Kada se podaci pošalju na ovu stranicu onda korisnik može ići pogledati na karti lokacije APova koje je poslao. Detaljniji opis plugina nalazi se u poglavlju vezanom za vlastiti doprinos gdje je poboljšavan ovaj plugin. Korišten je cijelo vrijeme ovaj plugin, međutim do vlastitog poboljšanja nije slao nikakve podatke na wigle.net stranicu, jer je mogao slati samo .gps.json datoteke, a te datoteke nije bilo moguće dobiti jer za to treba imati vanjski gps prijemnik.

4.5.15. Wpa-sec

Wpa-sec je drugi od dva plugina preko kojih se može probiti šifra koju Pwnagotchi uhvati. Funkcionira jako slično OnlinHashCrack pluginu tako da se na API od stranice šalju .pcap datoteke koje se pokušaju dekriptirati na njihovoj stranici. Jedna prednost ovog plugina je što se u njegovoj konfiguraciji može namjestiti da se probijene šifre preuzmu u čitljivom obliku na Pwnagotchi. Preuzimanje se događa kada je Pwnagotchi u MANU načinu rada i kada postoji veza s internetom. Detaljnije funkcioniranje plugina objašnjeno je u poglavlju vezano za probijanje šifri.

4.5.16. Wpa-sec-list

Ovo je plugin koji je izrađen kao dio vlastitog doprinosa zajednici i služi za prikaz preuzetih šifri sa wpa-sec web stranice. Šifre se preuzimaju sa stranice jedino ako korisnik to namjesti u konfiguraciji inače neće biti ni šifri niti će ovaj plugin raditi. Wpa-sec-list ima svoju stranicu koja je formatirana kao tablica gdje su redovi iz preuzete liste šifri ljepše formatirani. Detaljnije funkcioniranje plugina objašnjeno je u poglavlju vezano za vlastiti doprinos Pwnagotchi zajednici.

4.6. Učenje Pwnagotchija

Kreator Pwnagotchija kada je radio s umjetnom inteligencijom koju je ugradio u Pwnagotchi želio je da tako poboljša Pwnagotchi tako da mu daje mogućnost izmjene svoje konfiguracije da se bolje prilagodi okolišu, ali i da za korisnika to bude jedan dio procesa korištenja Pwnagotchija. Što se umjetne inteligencije i RL algoritma tiče on uči samostalno samo tako što će biti izložen podražajima iz okoline. Od korisnika se očekuje da će Pwnagotchija upoznati s različitim okolinama i dati mu raznolike podatke tako i tako će Pwnagotchi brže učiti.

Ono što Pwnagotchi uči je namjestiti parametar potražnje prema podražajima iz okoliša, jer za određeni okoliš postoji kombinacija parametara koja mu dopušta da uhvati više WPA rukovanja. Parametri koje Pwnagotchi koristi su:

```
# list of channels to recon on, or empty for all channels
personality.channels = []
# minimum WiFi signal strength in dBm
personality.min_rssi = -200
# number of seconds for wifi.ap.ttl
personality.ap_ttl = 120
# number of seconds for wifi.sta.ttl
personality.sta_ttl = 300
# time in seconds to wait during channel recon
personality.recon_time = 30
# number of inactive epochs after which recon_time gets multiplied by
    recon_inactive_multiplier
personality.max_inactive_scale = 2
# if more than max_inactive_scale epochs are inactive, recon_time *=
    recon_inactive_multiplier
personality.recon_inactive_multiplier = 2
# time in seconds to wait during channel hopping if activity has been performed
personality.hop_recon_time = 10
# time in seconds to wait during channel hopping if no activity has been performed
personality.min_recon_time = 5
# maximum amount of deauths/associations per BSSID per session
personality.max_interactions = 3
# maximum amount of misses before considering the data stale and triggering a new
    recon
personality.max_misses_for_recon = 5
# number of active epochs that triggers the excited state
personality.excited_num_epochs = 10
# number of inactive epochs that triggers the bored state
personality.bored_num_epochs = 15
# number of inactive epochs that triggers the sad state
personality.sad_num_epochs = 25
```

Ovaj dio koda preuzet je iz config.toml datoteke i za svaki parametar postoji komentar na engleskom za njega. Ono što se može vidjeti iz parametara je to da se određuju WiFi kanali koji će se pretraživati, minimalna jačina signala od APa, koliko će trajati potraga na kanalu, koliko će se puta napasti AP i ostalo. Na ove parametre korisnik nema utjecaja, ali može ih pratiti preko log datoteke preko plugina za čitati log. Prilikom učenja ovog Pwnagotchija praćena je log datoteka da se promatra kako se mijenjaju parametri, ali teško je izvući podatke iz tog toka pošto se ne zna što sve točno Pwnagotchi osjeća u svojoj okolini. Pwnagotchijeva funkcija za određivanje nagrade je sljedeća:

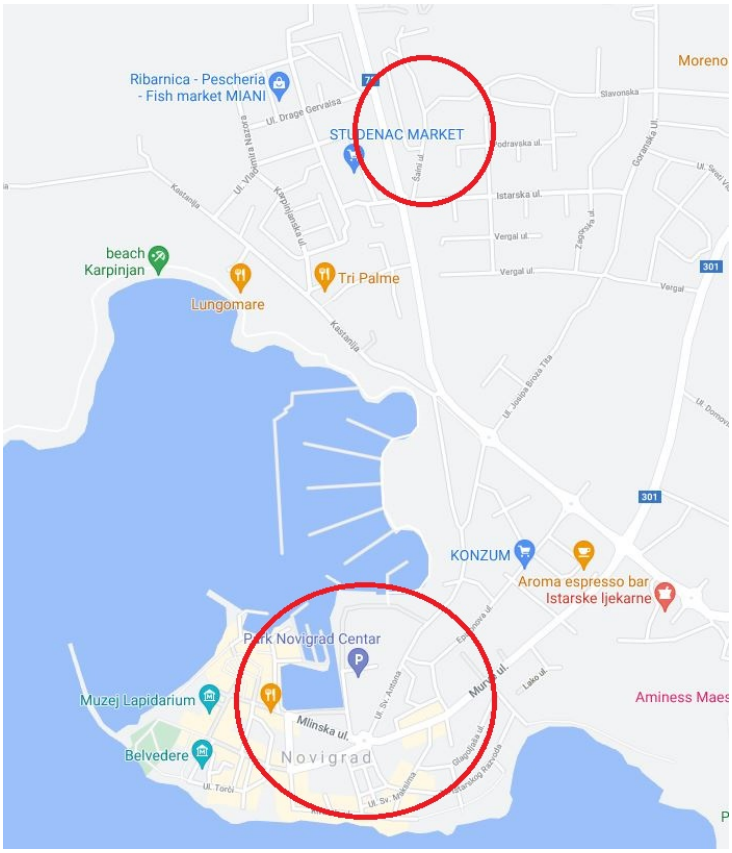
```
# state contains the information of the last epoch
# epoch_n is the number of the last epoch
tot_epochs = epoch_n + 1e-20 # 1e-20 is added to avoid a division by 0
tot_interactions = max(state['num_deauths'] + state['num_associations'], state['
    num_handshakes']) + 1e-20
tot_channels = wifi.NumChannels

# ideally, for each interaction we would have an handshake
h = state['num_handshakes'] / tot_interactions
# small positive rewards the more active epochs we have
a = .2 * (state['active_for_epochs'] / tot_epochs)
```

```
# make sure we keep hopping on the widest channel spectrum
c = .1 * (state['num_hops'] / tot_channels)
# small negative reward if we don't see aps for a while
b = -.3 * (state['blind_for_epochs'] / tot_epochs)
# small negative reward if we interact with things that are not in range anymore
m = -.3 * (state['missed_interactions'] / tot_interactions)
# small negative reward for inactive epochs
i = -.2 * (state['inactive_for_epochs'] / tot_epochs)

reward = h + a + c + b + i + m
```

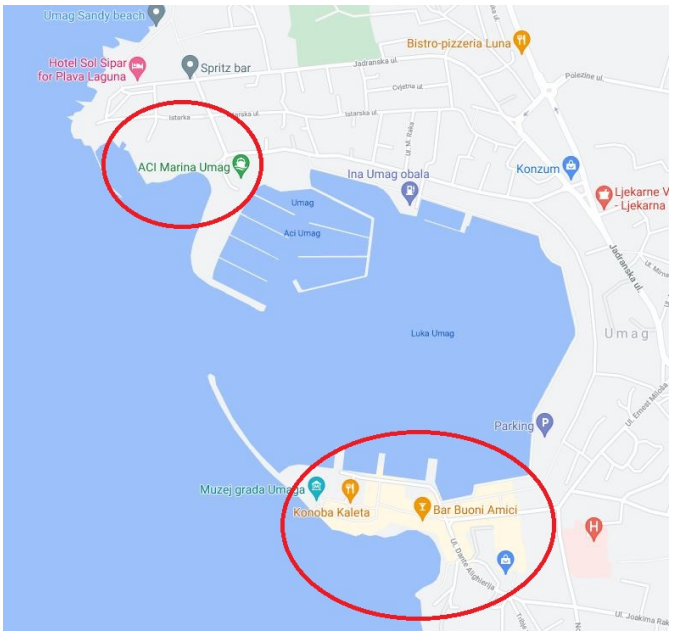
Vremenski periodi s kojima Pwnagotchi radi zovu se *Epochs*, prevedeno na hrvatski to bi bile epohe. Trajanje jedne epohe nije fiksno i mijenjat će se kako se mijenjaju parametri vezani za trajanja. Trajanje epohe je vrijeme potrebno da se izvrši glavna funkcija Pwnagotchija koja je zapravo jedna petlja koja se neprestano ponavlja u uključuje skupljanje svih APova koji se mogu osjetiti, slanje PMKID napada i napadanje odspajanjem uz dodatno vrijeme čekanja da se APovi reagiraju na napade.

Korisnik ima mogućnost izmjene nekih parametara koji utječu na učenje. "Laziness" je parametar lijenosti koji je označen brojem što za Pwnagotchi znači da što je on veći to Pwnagotchi više vremenskih perioda provodi koristeći svoje znanje, a manje vremenskih perioda uči. Prema uputama sa službene stranice korisnik bi trebao povećati vrijednost tog parametra kako je Pwnagotchi iskusniji tako da se više usredotoči na iskorištavanje dobivenog znanja nego da pokušati skupiti još više. Manja vrijednosti parametra lijenosti označava veću učestalost ulaska Pwnagotchija u period učenja u kojem iskorištava širi spektar parametara kako bi uvidio moguća poboljšanja u ponašanju. Veća vrijednost parametra lijenosti označava veću učestalost perioda normalne operacije gdje Pwnagotchi iskorištava znanje koje je skupio tijekom učenja i ovaj period obilježen je konzervativnijim vrijednostima parametara operacije. Pwnagotchijevo početno stanje parametra lijenosti je 0.1 što mu osigurava najbrže učenje. Na službenoj stranici nije izričito navedeno u kojim razmacima bi trebalo povećavati parametra lijenosti, niti je na forumskim raspravama naveden preporučeni proces povećavanja tog parametra. Kako ne postoji službena napomena kako mijenjati parametar lijenosti odlučeno je bilo povećati ga za 0.2 nakon sva šetnje i procijeniti učinke. Ova nasumična procjena bila je efektivna jer je uočen veći broj skupljenih šifri i detektiranih APova. Tijekom korištenja vrijednosti parametra lijenosti povećana je sa 0.1 na finalnih 0.9. Moguće bi bilo ostaviti Pwnagotchija da duži period vremena radi na manjem parametru lijenosti kako bi više učio, međutim za to treba koristiti uvijek nova okruženja kako mu ne bi postalo dosadno. Koristeći nekoliko mjesta koja su bila posjećena više puta Pwnagotchi je bolje reagirao na manji parametra lijenosti. Kako je Pwnagotchi vođen u šetnje i oko područja gdje je već bio uviđeno je da je brzo počeo pokazivati lice da mu je dosadno, jer nije našao nove APove s kojima može raditi. To je bilo prije nego što je podignuta vrijednost lijenosti nakon čega je počeo osjećati nove APove na mjestima gdje je već bio, pretpostavlja se jer je mogao iskoristiti znanje koje je naučio. Još jedan važan faktor kod te lijenosti je što kada je ona mala Pwnagotchi konstantno ulazi u nove epohe učenja i brzo se zasiti s jednim područjem. Kada je bila povećana lijenost primijećeno je da Pwnagotchi "spava" duže tj. uzme duže periode čekanja gdje ne radi ništa, pretpostavlja se da to radi jer osjeti da stoji dugo na istom mjestu. Drugi parametar koji utječe na učenje je broj epoha po epizodi učenja. Početno

taj broj je bio 50, a smanjen je na 15. Ono što sam time dobio je koncentriranije epizode učenja jer se znalo dogoditi da je Pwnagotchi imao 20 zanimljivih epoha i onda 30 dosadnih do kraja epizode. Vlastitom promjenom učinilo se da mu epizode učenja budu zanimljivijim i također može brže učiti jer se saznanja iz epizode učenja primjenjuju tek na kraju epizode, tako da on sad uči 2 puta brže nego je učio na početku. Fizičke lokacije koje su obiđene sa Pwnagotchijem nalaze se u Umagu i Novigradu i pokazat će ih se na karti zaokružene s crvenim krugovima.



Slika 11: Područje učenja u Novigradu (Izvor: Google Maps)



Slika 12: Područje učenja u Umagu (Izvor: Google Maps)

Želja je bila prikazati područja na karti koja su obiđena sa Pwnagotchijem da se pokaže o kakvim se različitim područjima radilo. Gornje područje označeno s crvenim krugom u Novigradu je rezidencijalni dio grada gdje ljudi žive u kućama i ima apartmana gdje dolaze turisti. Šifre koje je Pwnagotchi ovdje pokupio bile su sve od kućnih APova koje ljudi koriste za osobne potrebe i ima odvojenih APa za turiste u apartmanu. Donji krugovi na obje karte su centri grada Novigrada i Umaga gdje je bilo više vrsta APova. Pwnagotchi je uhvatio APova koji se koriste za kućne potrebe, one koje koriste ugostiteljski objekti odvojeno za unutarnje potrebe, a i za goste lokala, te čak WiFi hotspotove mobitela. U centrima gradova naravno bila je najveća gustoća APova. Zadnji crveni krug na vrhu druge karte odnosi se na plažu i beach bar kao drugi oblik okoliša na kojem sam trenirao Pwnagotchija. To područje je imalo jako malo kućnih APova, nekoliko APova od beach bara i hotela tamo, te dosta mobilnih WiFi hotspota. Po pregledu probijenih šifri koje je Pwnagotchi uhvatio može se napraviti povezanost sa svrhom AP u vezi s jačinom njegove šifre. Primijećeno je da su najsigurnije šifre one koje ljudi sami postavljaju za svoje kućne APove i APove koje lokali koriste za unutarnju mrežu. To je tako očito jer neki ljudi vode računa o sigurnosti svoje mreže kod kuće, a sigurne šifre u unutarnjim mrežama lokala su takve, jer pretpostavlja se da su lokali unajmili stručnu osobu da im napravi mrežu. S druge strane najnesigurnije šifre su one od WiFi mobilnog hotspota gdje ljudi dosta često stavljaju "12345678" jer mobitel traži šifru od minimalno 8 znakova. To nije toliki sigurnosni rizik, jer se na mobitelu lako vidi koliko je uređaja spojeno i što udaljenost potrebna za spajanje nije velika. Ono što je isto očito, ali nije iznenađujuće, je da su APovi namijenjeni gostima lokala, gostima apartmana i neki kućni APovi zaštićeni jako lošim šiframa. Loša šifra na kućnim APovima pripisujem ljudima koji su stari ili nisu svjesni što je loša šifra pa koriste jednostavne šifre koje se lako pamte, ali nisu sigurne. U vezi gostiju lokala i apartmana, APovi koji su namijenjeni njima isto imaju lako pamtljive šifre kako bi konobar mogao lako reći gostu koja je šifra i da se gosti u apartmanu ne muče s upisivanjem šifre.

5. Pwnagotchijeva sposobnost probijanja šifri

Probijanje šifri tuđih WiFi AP-a nije primarna uloga Pwnagotchija iz dva razloga: imati tuđu šifru znači mogućnost upada u njihov sustav što se očito može smatrati napadom na nečiju mrežu, a kao drugi razlog je što Raspberry Pi Zero W nema dovoljne računalne performanse za izvoditi proces probijanja šifri u razumnom vremenu. Što se tiče hvatanja tuđih šifri, to nije legalno i moralno raditi, ali donekle se tolerira i ne može se otkriti da se napad dogodio ako ti podaci ne dovedu do daljnjih napada na nečiju mrežu. Druga strana te priče je da se osobi i široj javnosti može prikazati sigurnosni problemi s opremom, mrežnim protokolima ili programima, ako se na njih ili preko njihovih slabosti uspješno provedu napadi. Pwnagotchijev kreator pristaša je ove druge strane gdje se pokušava prikazati problem sa sadašnjim najkorištenijim sigurnosnim protokolom za zaštitu WiFija WPA/WPA2 protokolom. Sama činjenica da se toliko lako dođe do kriptiranih šifri mreže dovoljni je pokazatelj da je potrebna inovacija na području tog dijela sigurnosti WiFi-ja. U ovom poglavlju dublje će se ući u način kako se spremaju uhvaćene šifre na Pwnagotchi, kako se probijaju i zašto Pwnagotchi nema mogućnost samostalno probiti šifru, te što ga sprječava u tome.

5.1. Formati datoteka šifra

Pwnagotchi kada uhvati WPA rukovanje sprema taj podatak u .pcap obliku, gdje se stvara jedna .pcap datoteka za svaki AP za koji Pwnagotchi nešto uhvati. Ta jedna .pcap datoteka za AP sadrži sva uhvaćena rukovanja vezana za taj AP, međutim u tim rukovanjima ne mora se nužno nalaziti šifra u kriptiranom obliku [7]. Datoteke sa .pcap ekstenzijom koriste se za spremanje podataka koji su uhvaćeni presretanjem mrežnih paketa putem alata za presretanje mrežnih paketa. Ova vrsta datoteka se koristi za spremanje mrežnih paketa, jer se u programima za hvatanje mrežnog prometa vrši dodatna konverzija mrežnih podataka u podatke koji su čitljiviji ljudima [32].

Temeljni podatak od kojeg kreće probijanje WPA kriptirane šifre je *Pairwise Master Key* (kreće PMK). On se u svojem originalnom obliku neće nikad naći u .pcap datoteci, jer se on ne razmjenjuje u četverostrukom rukovanju, već se razmjenjuju ključevi nastali od PMK-a. Ti derivati PMK-a ne maskiraju PMK jer su nastali kroz napadaču poznatu proceduru i napadaču poznate podatke, te se lako dolazi do PMK-a. PMK je polazište probijanja šifre, jer je vremenski i računalnim kapacitetima najzahtjevniji za probiti, a nakon probijanja dobiva se šifra u čitljivom obliku. Enkripcija koja se koristi za stvaranje PMK-a iz šifre u čitljivom obliku je PBKDF2-SHA1 način stvaranja hash vrijednosti. SHA1 je algoritam kreiranja hash vrijednosti gdje koristi sol tj. neku vrijednost koja dolazi iz sustava koji radi hashiranje i pridružuje je originalnoj vrijednosti da se osigura stvaranje po sustav jedinstvene hash vrijednosti. To je prvi korak u PBKDF2-SHA1 algoritmu. Sljedeće korake lakše je prikazati preko programskog koda pa objasniti.

```
data1 = SHA1_Transform(sifra, sol);
data2 = SHA1_Transform(sifra, sol);
for (int i=0; i{
    data1 = SHA1_Transform(data1, data2);
```

```
    data2 = SHA1_Transform(data2, data1);
}
```

Stvaranje data1 kao glavne varijable i data2 kao pomoćne varijable navedeni je prvi korak u PBKDF2-SHA1 algoritmu, a drugi korak je izvođenje petlje 4096 iteracija. Petlja stvara novu vrstu data1 varijable svaku iteraciju koristeći data2 varijablu iz prošle iteracije, te promijeni data2 varijablu. Stvarajući različiti ključ u svakoj iteraciji služi za usporavanje napada grubom silom, jer za isprobati jednu pretpostavljenu šifru u čitljivom obliku treba izvrtjeti 4096 iteracija ove petlje umjesto da se samo jednom napravi SHA1 hash. Kada se pomoću PBKDF2-SHA1 kreira PMK, koristi se *Service Set Identifier* (kraće SSID) tj. ime WiFi mreže kao sol, šifra mreže za glavnu vrijednost i nastaje 256 bitni PMK [33] [34].

Programi koji probijaju WPA šifre većinom koriste .pcap datoteke kao ulazni podatak za započeti probijanje gdje ti programi dalje razgrađuju .pcap i traže podatke vezane za WPA šifru. Jedan nedostatak kod .pcap datoteka je što mogu narasti poprilično velike kad se sakupi puno mrežnog prometa, a možda većina tog prometa neće sadržavati podatke koji se mogu iskoristiti za probijanje šifre. Kreatori Hashcat-a, danas najboljeg programa za probijanje šifri osmislili su svoju vrstu datoteke, .hccapx koja se može stvoriti s alatom hcxpcapngtool kako bi sadržavala samo podatke relevantne za probijanje šifri i time smanjile veličinu datoteke. Prednost .hccapx datoteka je što se prilikom njihovog stvaranja smanjuje sadržaj samo na ono što je potrebno za probijanje šifre, a nedostatak je što takav format koristi mali broj alata za probijanje šifre pošto je ta vrsta datoteke kreirana od strane kreatora hashcat alata.

5.2. Napadi probijanja šifri

Postoje tri napada s kojima se mogu probiti WPA šifre: napad grubom silom (engl. *Brute-force attack*), napad rječnikom (engl. *Dictionary attack*) i neki programi nude poboljšanu verziju napada rječnikom, napad temeljen na pravilima (engl. *Rule-based attack*). Redom kojim su navedeni napadi povećava im se efektivnost i brzina probijanja šifre, zato je logično da napad temeljen na pravilima koriste web stranice na koje Pwnagotchi može poslati uhvaćena WPA rukovanja. Programi koji se mogu instalirati lokalno nude izbor koji će se napad koristiti za probijanje šifre. Ono što je jedinstveno svim napadima je da se za probijanje generira šifra u čitljivom obliku koja bi mogla biti jednaka traženoj i provodi se nad njom isti proces kriptiranja kao što se proveo nad traženom šifrom. Na kraju kriptiranja pogođene šifre, ako hash vrijednost ispadne identična uhvaćenoj hash vrijednosti tražena šifra u čitljivom obliku jednaka je pogođenoj šifre u čitljivom obliku. Razlika između napada je način na koji se dolazi do pogođene šifre.

5.2.1. Brute-force attack

Brute-force attack najjednostavniji od tri napada. Iz imena daje se naslutiti da taj napad koristi sirovu silu što bi u računalnom smislu bila njegova sposobnost izvođenja izračuna. Ova vrsta napada generira pogođenu šifru u čitljivom obliku tako što za sve znakove koje se mogu nalaziti u šifri ili samo određeni set znakova generira sve moguće kombinacije danih znakova.

Pošto postoji konačni izbor znakova koji se mogu nalaziti u šifri i šifra ima svoju maksimalnu dužinu znači da se ovim napadom u određenoj količini vremena može probiti bilo koja šifra [35]. Svemu tome prethodi znanje o kojem se protokolu kriptiranja radi, ali velika većina današnjih protokola za kriptiranje je javno, jer se time isto provjerava njihova snaga.

Kod ovog napada postoji više faktora s dvije strane koji definiraju količinu vremena potrebno da se probije šifra. Sa strane šifre vrijeme probijanja se produljuje s povećanjem dužine šifre i s povećanjem različitih vrsta znakova koje se nalaze u šifri. Sa strane računala koje radi probijanje vrijeme probijanja skraćuje se s povećanjem broja izračuna u sekundi. Pošto ovaj napad generira svaku moguću šifru od izbora mogućih znakova postoji formula za izračun broja mogućih šifri glasi ovako: broj mogucih znakova$^{\text{broj znakova sifre}}$ [36]. Na primjer, za šifru koja je duljine 8 znakova, a znamo da su svi ti znakovi brojevi to nam daje $10^8 = 100.000.000$ mogućih kombinacija za šifru, ali šifra duljine isto 8 znakova gdje znamo da su svi znakovi dolaze iz engleske abecede i samo su mala slova to nam daje $26^8 = 208.827.064.576$ mogućih kombinacija za šifru, a kad uključimo velika i mala slova engleske abecede i posebne znakove koji se mogu koristiti u WiFi šifri to nam daje $94^8 = 6.095.689.385.410.816$ mogućih kombinacija. Računalu koje može izgenerirati 1,000,000,000 kriptiranih šifri u sekundi trebat će najduže 83 i pol dana da probije šifru zadnjeg navedenog formata [37].

Važno je napomenuti da nije svejedno koji je algoritam kriptiranja korišten, jer dok na primjer SHA1 algoritam ima jedan korak kriptiranja, navedeni PBKDF2-SHA1 koji se koristi za WPA šifre ima 4096 ponavljanja koraka za izgenerirati jednu kriptiranu šifru, što znači da brute-force napad mora povrh isprobavanja svake mogućnosti šifre, za svaku pojedinu šifru ponoviti potrebna 4096 koraka kriptiranja, tim načinom PBKDF2-SHA1 produljuje trajanje mogućeg brute-force napada. Web stranice na koje Pwnagotchi može poslati uhvaćene šifre zato koriste napade temeljene na pravilima, čak iako Onlinehashcrack stranica nudi brute-force napad uz naplatu, njihova snaga procesiranja omogućava im da probiju šifru duljine 8 znakova od brojeva, malih i velikih slova engleske abecede za 315 dana uz cijenu od 12$ po satu obrade [38]. Iz danih primjera očito je zašto se brute-force napad najrjeđe koristi, ali i zašto je potrebno kreirati duže šifre i koristeći velika i mala slova, brojeve, te posebne znakove.

5.2.2. Dictionary attack

Napad rječnikom ili dictionary attack je evolucija brute-force napada koja ubrzava proces probijanja šifre i/ili smanjuje potrebne računalne zahtjeve. Dictionary attack je evolucija brute-force napada, jer i dalje način traženja šifre je pogađanje šifre, ali ovaj put ne isprobavaju se sve moguće kombinacije šifri za dani izbor znakova. Napad rječnikom temelji se na činjenici da se može žrtvovati mogućnost probijanja svih šifri jako sporo, da bi se dobilo na brzini probijanja za neke šifre. Ljudi koji sami sebi za svoju uporabu stvaraju šifru, naprave je da često sadrži neki pojam kojeg neće zaboraviti i time neće zaboraviti svoju šifru. Time se dolazi do zaključka da je moguće stvoriti veliki rječnik koji će sadržavati jedan dio mogućih šifri u čitljivom obliku koje će onda program kriptirati i testirati prema traženoj vrijednosti. Žrtvuje se mogućnost probijanja nekih šifri, ali se dobiva na manjem vremenu potrebnom da se prođe kroz rječnik mogućih pojmova, jer je iz fonda mogućih šifri izbačen veliki dio kombinacija slova i brojki koji iako mogu

biti validna šifra ljudi neće koristiti, jer je poprilično nepamtljiva. Prednosti ove vrste napad je naravno brzina kojom se prolazi kroz rječnik tj. umjesto isprobavanja 6 bilijuna kombinacija, rječnici imaju od milijun do par stotina milijuna mogućih šifri. Nedostatak ove vrste napada je naravno mogućnost da se tražena šifra ne nalazi u rječniku što znači da nećemo moći probiti šifru [39].

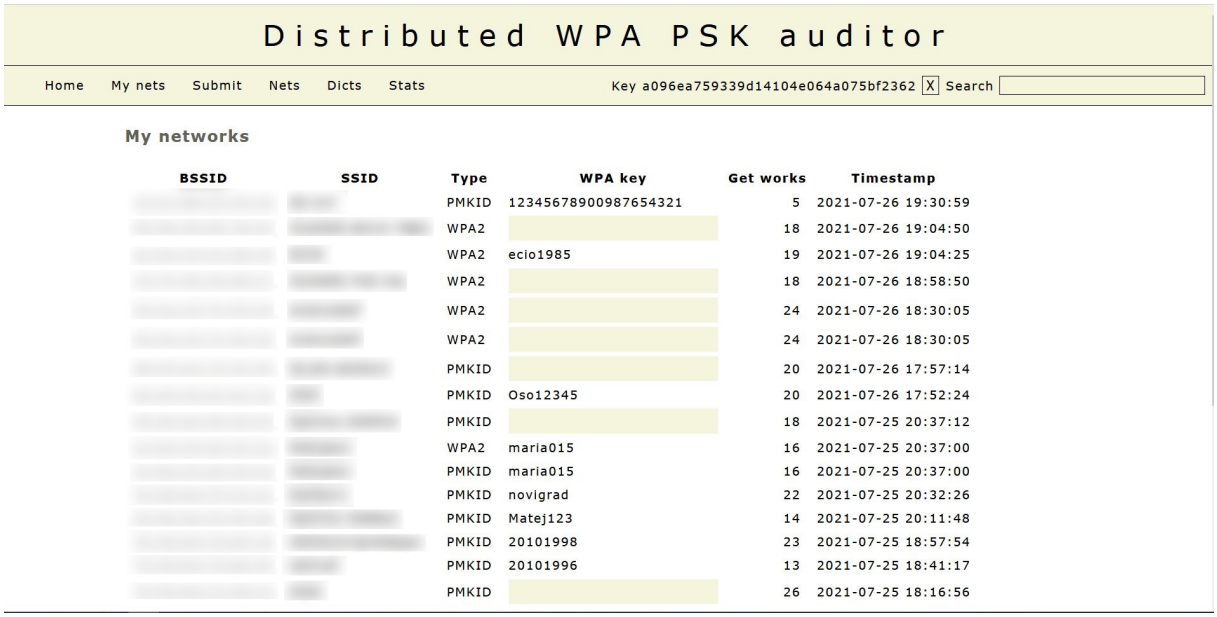
5.2.3. Rule-based attack

Napad temeljen na pravilima je zadnji korak u evoluciji brute-force napad što znači da se i dalje kriptiraju pogođene šifre i uspoređuju s traženom vrijednosti. Novina kod ove metode je što je znatno kompliciranija od prošlih verzija napada, ali zato i uspješnija. Temelji se na činjenici da je napad rječnikom prekrut oblik napada gdje nije efikasno pisati svaku verziju šifre koja bi recimo mogla sadržavati jedan pojam i par brojeva, ali bi pojam mogao imati prvo ili zadnje veliko slovo, a brojke bi mogle biti u svakakvim kombinacijama. Za napad rječnikom sve te kombinacije bi trebale prethodno biti zapisane u rječniku inače šifra ne bi mogla biti probijena. Rule-based napad daje mogućnost da se programatski kreiraju mogući kandidati za šifru iz postojećeg rječnika, ali da se prema pravilima svaka riječ modificira na određeni način [40]. Time se ovaj napad smješta između dictionary napada i brute-force napada sa širinom mogućih šifri tako da nije preširok u besmislene kombinacije i nije preuzak u strogo definirane riječi, a zbog složenosti je sporiji od dictionary napada i brži od brute-force napada. Pravila omogućavaju veliki raspon modifikacije zapisa u rječniku tako da dopuštaju: pretvaranje slova u velike i male, dodavanje znakova na bilo koju poziciju, brisanje znakova, prepisivanje znakova, zamjena mjesta znakovima, izvrtanje riječi itd. Ovaj način obavljanja napada daje veliku fleksibilnost, jer moguće uz poznavanje nekakvih čestih praksi ili uz korištenje psihologije doći do dobrih rezultata s dobro promišljenim pravilima.

Liste s pravilima moguće je stvoriti sam ili iskoristiti postojeće liste pravila. Najpoznatija lista je lista sa 64 najboljih pravila koja se u jednom istraživanju pokazala kao jako dobrom i nju koriste web stranice na koje Pwnagotchi može poslati WPA šifre [41]. U tom istraživanju pokazano je kako složenost liste pravila povećava, u odnosu na rječnik koji se koristi, broj mogućih kandidata šifre koji se moraju isprobati. Velikom listom pravila se time produljuje vrijeme trajanja probijanja, a možda ne toliko povećava mogućnost probijanja šifre, čak iako vrijedi pravilo da sa složenijom listom postoji veća šansa da će se šifra probiti. Za listu pravila također vrijedi da može biti toliko složena da ako joj je dana relativno jednostavna šifra, trebat će mnogo vremena da se pronađe zbog nepotrebnih kombinacija koje stvori lista pravila. Lista najboljih 64 pravila je prema testiranju dobra jer ne stvara preveliki broj kandidata, ima zadovoljavajući broj probijenih šifri, a najvažnija činjenica za web stranice je što ima mali prosječni broj pokušaja dok dođe do tražene šifre. Ovo je trenutno najbolji način probijanja WPA šifri, jer je moguće efikasno prilagoditi širinu mogućih kandidata i moguće vrijeme trajanja probijanja šifre, gdje ako osoba ima nekakva saznanja o mogućem obliku šifre može prilagoditi potreba za tim tipom.

5.3. Programi za probijanje šifri

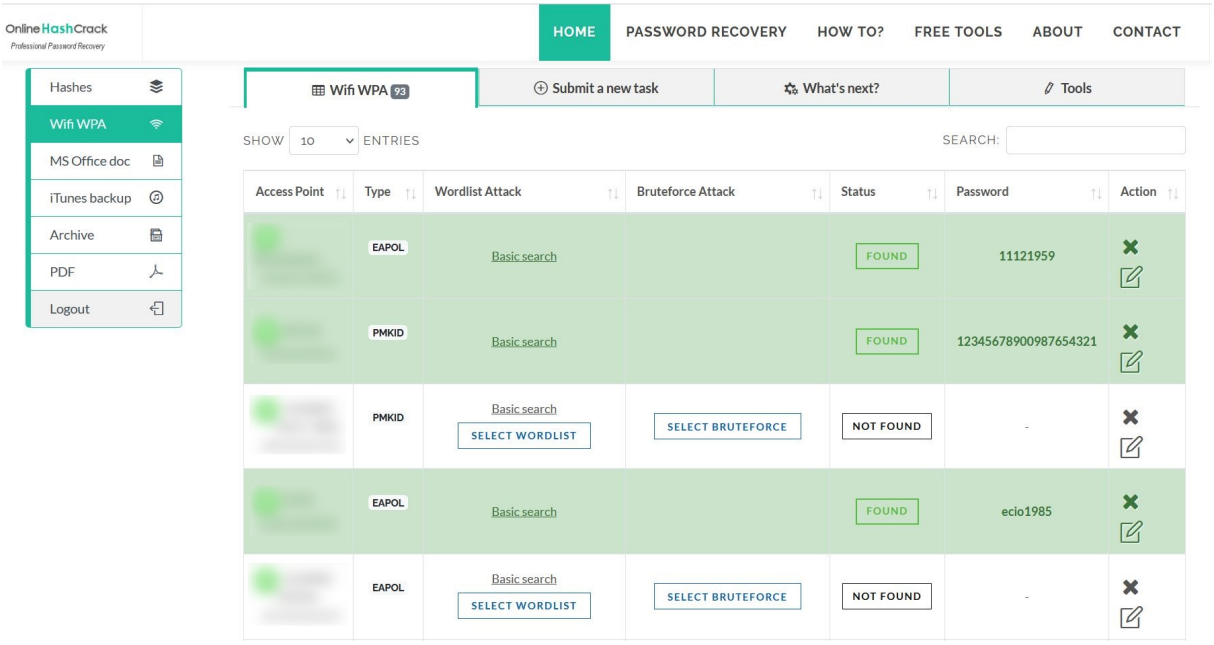
Postoji mnoštvo programa za probijanje šifri općenito, ne samo WPA šifri. Hashcat i John The Ripper su najpoznatiji i trenutno najbolji programi takve vrste. Oni mogu provoditi sve već navedene vrste napada nad WPA šiframa koje im se daju, a mogu se i dodatno konfigurirati ako se specijalizirano radi probijanje šifri uz korištenje procesora grafičke kartice koji je višestruko bolji za taj zadatak od običnog procesora. Te programe se za korištenje sa Pwnagotchijem mora instalirati na posebno računalo na koje će se sa Pwnagotchija prebaciti uhvaćene šifre. Pwnagotchi kao sustav nema ugrađen niti jedan od navedenih programa za probijanje šifra, a ni druge moguće programe. Razlog za to je što je pretpostavljena hardverska podloga Raspberry Pi Zero W koji nema ni približno dovoljno računalnog kapaciteta za pokušati probiti šifre koje uhvati. Kako bi se olakšalo probijanje uhvaćenih šifri ako korisnik Pwnagotchija to želi, kreator Pwnagotchija stvorio je plugine koji nakon aktivacije šalju uhvaćene šifre na web stranice wpa-sec.stanev.org i na onlinehashcrack.com. Te web stranice omogućavaju ljudima koji žele probiti WPA šire da to naprave bez instalacije programa poput hashcat-a i John The Ripper-a lokalno i uče koristiti te programe.



Slika 13: Wpa-sec web stranica (Izvor: Wpa-sec, 2021)

Nemaju obje stranice iste mogućnosti, wpa-sec je stranica usmjerena samo na WPA šifre gdje se mogu predati samo .pcap datoteke i koja radi na principu distribuirane obrade. Distribuirana obrada u ovom slučaju znači da ljudi koji žele pomoći ovoj stranici mogu skinuti Python skriptu koja dohvaća jednu po jednu šifru na računalo i pokuša je probiti, te ako uspije vrati rezultat na wpa-sec stranicu. Prije nego se šifre mogu poslati sa Pwnagotchija mora se kreirati jedinstveni ključ korisnika koji se unosi u Pwnagotchi kako bi se identificirao pošiljatelj šifri. Taj ključ služi za pregled poslanih i probijenih šifri na njihovoj stranici.

OnlineHashCrack web stranica koristi računanje u oblaku za probijanje šifra više vrsti datoteka: WPA šifru, šifru bilo koje MS Office datoteke, šifru PDF datoteke, šifru Zip/Rar datoteke ili šifru iTunes backup-a. Za šifre poslane sa Pwnagotchija koristi se osnovni paket usluga



Slika 14: OnlineHashCrack web stranica (Izvor: OnlineHashCrack, 2021)

koje nude, a to je da se može pristupiti listi poslanih i probijenih šifri, a šifre se probijaju jednostavnim napadom temeljenim na pravilima uz korištenje najboljih 64 pravila. Stranica nudi bolje liste pravila i rječnike za napadati šifre, ali taj dio usluge se naplaćuje. Za započeti koristiti ovu opciju na Pwnagotchiju potrebno je registrirati se na OnlineHashCrack stranici s e-mail adresom koja je onda jedinstveni identifikator korisnika i mora se unijeti u Pwnagotchi prije nego se može početi koristiti.

5.4. Analiza uhvaćenih WPA šifri

Kroz korištenje Pwnagotchija uhvaćeno je ukupno 93 WPA šifre i podijeljene su prema načinu na koji su uhvaćene: PMKID napad i uhvaćeno klasično rukovanje. Od ukupno 93 šifre, 59 ih je došlo preko PMKID napada, a 34 ih je od klasičnih rukovanja. Prema tome podjela po postocima glasi 63% PMKID šifri i 37% običnih šifri. Uz znanje na koji način Pwnagotchi radi i kako se može doći do šifre, postoci prema izvoru šifre nisu iznenađujući, jer za PMKID nije potrebno imati spojeni uređaj na AP za uhvatiti šifru, a za obično rukovanje je to potrebno. To ide u prilog načinu korištenja Pwnagotchija gdje se šetalo ili vozilo kroz neko područje i nije bilo vremena čekati da se uređaj spoji na AP za uhvatiti rukovanje. Doduše protiv ovog argumenta ide činjenica da nisu svi AP-i ranjivi na PMKID napad i kod takvih AP-a jedini način za pribaviti šifru je uhvatiti obično rukovanje. Kada se gleda postotak probijenih šifri podijeljenih po izvoru šifre, 26% običnih rukovanja i 37% PMKID šifri je probijeno, međutim izvor šifre nije uopće faktor u mogućnosti ili težini probijanja šifre.

Za probijanje uhvaćenih šifri korištene su samo navedene dvije web stranice za koje je aktivirano da Pwnagotchi automatski šalje uhvaćene šifre. Lokalno nije bilo pokušavano probijati šifre, jer zahtijeva korištenje Linux operativnog sustava i relativno snažnog računala, a te uvjeti nisu postojali, doduše ne bi to ostvarilo značajnu ako ikakvu razliku u broju probijenih

šifri. Pri početku korištenja Pwnagotchija i web stranica kao odabranog medija za probijanje šifri, nije bilo nade da će se toliki broj šifri probiti, ali onda nije ni bilo poznato da postoji napad temeljen na pravilima koji je uvelike pomogao da se probiju šifre. Wpa-sec stranica očito ima bolje rječnike i/ili liste pravila, jer je bila malo uspješnija u probijanju šifri.

Primjeri probijenih šifri: "12345678900987654321", "ecio1985", "Oso12345", "novigrad", "20101998", "LaRiva123", "nemaneta", "09121977"

Iz probijenih šifri može otkriti nešto o samim šiframa, ali i o stranicama koje ih probijaju. Šifre koje su probijene imaju neke karakteristike koje ih čine jako slabim: sadržavaju samo brojke, sadržavaju samo riječi iz engleskog jezika, sadržavaju samo osobna imena bilo kojeg kulturnog područja, na imena ili riječi dodani su brojevi na početak ili kraj, te su sve šifre minimalne dužine (8 znakova). Kao što je navedeno ranije, šifre koje sadrže samo brojke nemaju dovoljno široki raspon znakova i apsolutno nisu sigurne jer se jako lako i brzo probiju. Koristiti riječi engleskog rječnika nije poželjno jer ne postoji mnogo specijaliziranih po jezicima, a jedan veliki od svih jezika je nezgrapan za koristiti u probijanju. Ako se na takve riječi dodaju brojke na početka ili kraj to ne povećava puno složenost šifre, jer postoje pravila koja su dizajnirana za takve šifre. Osobna imena isto nije pametno koristiti, jer pretpostavljam da je lako agregirati listu osobnih imena koja očito već postoji, a ne postoji toliko različitih osobnih imena da postaje nezgrapno koristiti takav rječnik. Još jedna glavna slabost je koristiti šifre minimalne duljine što isto može biti jasno zašto nije dobro iz formule o mogućem broju kandidata za šifru. To su stvari koje vidim i znam.

Drugi dio analize je pretpostavka o stvarima koje ne znam, a to su šifre koje nisu probijene i rječnici, te pravila koje koriste stranice. Počevši od web stranica za probijanje, očito je da nisu napravljene da probiju jače šifre, jer wpa-sec ne želi previše opteretiti donatore koji distribuirano obrađuju, a OnlineHashCrack želi naplatiti bolju uslugu. Domet ovih stranica jesu kraće šifre bez posebnih znakova i gdje su riječi engleskog rječnika ili osobna imena. Za šifre koje nisu probijene pretpostavlja se bi da su oblikovane na jedan od sljedećih načina: sadrže specifično hrvatsku riječ uz dodatne brojke, sadrže englesku ili hrvatsku riječ s brojkama unutar riječi ili sadrže riječ s brojkama i posebnim znakovima. Za ovaj zadnji oblik šifre dala bi najveća vjerojatnost da je najsigurniji, jer ostala dva oblika koji su navedeni moguće je da nisu bili probijeni jer web stranice nemaju takvu mogućnost probijanja, no to ne znači da netko tko to lokalno radi neće imati.

6. Vlastiti doprinos

Tema ovog poglavlja je predstavljanje vlastitog doprinosa Pwnagotchi zajednici u obliku poboljšanja i novih mogućnosti sustava koji su samostalno kreirani. Ovaj diplomski rad predviđa da se kroz izradu rada napravi nekakav doprinos Pwnagotchi zajednici implementacijom novih mogućnosti, poboljšavanje postojećih ili izradom nekakve analize sustava. Tijekom korištenja Pwnagotchija uočeno je nekoliko dijelova koji se mogu poboljšati i dodati. Jedan način na koji se zajednica može uključiti u poboljšavanje Pwnagotchija je tako što napravi prijevod za njegove poruke na svom jeziku. Napravljen je tako prijevod Pwnagotchijevih poruka na hrvatski jezik. Druga stvar koja je uočena je da se s pluginom za wpa-sec stranicu mogu skinuti probijene šifre na Pwnagotchi, ali nema načina da se prikažu na ekranu Pwnagotchija ili preko web sučelja. Tu je uočena mogućnost da se napravi plugin koji će nuditi tablični prikaz šifri koje su skinute na Pwnagotchi i kojima će se pristupati na isti način kako se pristupa ostalim pluginima na web sučelju. Treće poboljšanje koje je napravljeno vezano je za slanje podataka o AP-ovima na stranicu Wigle.net. Ta se stranica specijalizira za prikazivanje WiFi APova, transmitera mobilne mreže i bluetooth uređaja na karti svijeta, a podatke šalju dobrovoljci koji svojim uređajima detektiraju i bilježe te točke. Plugin za slanje podataka na Wigle.net koji je došao sa Pwnagotchijem podržava samo jedan oblik datoteke sa gps podacima, pa je bio proširen da podržava još druga dva oblika datoteke sa gps podacima.

6.1. Prijevod na hrvatski jezik

Ideju za ovo poboljšanje došla je čitanja službene stranice od Pwnagotchija, točnije sekcije koja objašnjava kako se može dati vlastiti doprinos Pwnagotchi zajednici. Na stranici pišu upute kako se radi novi prijevod i kako nitko još nije napravio hrvatski prijevod, odlučeno je da će se to napraviti u sklopu ovo projekta. Procedura nije teška već uključuje izvršavanje dvije skripte i nešto promjena u stvorenoj datoteci. Prije svega toga trebalo je napraviti fork GitHub repozitorija od evilsocketa na vlastiti GitHub profil, te klonirati taj repozitorij na vlastito računalo. Nakon toga potrebno je koristiti skriptu za generiranje novog direktorija i datoteke prijevoda [44]. Potrebno je pozicionirati se u pwnagotchi direktorij i izvršiti sljedeći dio koda u cmd prozoru:

```
./scripts/language.sh add "hr"
```

Ta skripta stvori direktoriji za dani jezik i datoteku ekstenzije .po koja sadrži podatke o kreatoru, te identifikatore svih poruka i njihove pripadajuće prijevode koje treba ispuniti. Za primjer par redova datoteke .po:

```
msgid "I'm living the life!"
msgstr "To se zove život!"

msgid "I pwn therefore I am."
msgstr "Pwnam dakle postojim."

msgid "I'm having so much fun!"
msgstr "Super se zabavljam!"
```

Prema .po datoteci koja se ispunilo s hrvatskim prijevodima stvara se drugom skriptom .mo datoteka koja je binarnog oblika i koja će se zapravo koristiti u Pwnagotchijevom kodu. Datoteka u kojoj se nalaze prijevodi ostaje zato da se mogu u čitljivom obliku pratiti promijene na GitHub repozitoriju. Skripta koja se koristi za generiranje .mo oblika:

```
./scripts/language.sh compile "hr"
```

Nakon toga potrebno je napraviti pull request na originalnom repozitoriju Pwnagotcija i čekati da autori potvrde promijene. Vlastiti prijevod nisam isprobao na ovom Pwnagotchiju i razloga što je za to potrebno kreirati iz nule sliku iz koje se instalira Pwnagotchi i prebrisati staru verziju. Staru verziju nije bilo moguće prebrisati, jer je Pwnagotchi bio usred procesa učenja i izgubilo bi se ono što je dosad Pwnagotchi naučio. Smatra se da nije potrebno isprobavati ovakvu promjenu, jer je dovoljno jednostavna, točnije jedini utjecaj koji je imala je upisivanje prijevoda u datoteku. U trenutku pisanja nije pregledan pull request i time nije inkorporiran u kod Pwnagotchija.

6.2. Tablica probijenih šifri

Ovo poboljšanje odlučeno je da će se napraviti kad je uočena mogućnost skidanja probijene šifre sa wpa-sec web stranice i tijekom listanja GitHub stranicu Pwnagotchija gdje je jedna osoba predložila da se napravi tablični prikaz svih uhvaćenih rukovanja njihovim podacima, vremenima i eventualno probijenih šiframa. Prijedlog s GitHub-a nije isti kao vlastita finalna ideja, jer .pcap datoteke nemaju vrijeme kad su uhvaćena rukovanja. Onda je uočeno da datoteka koja se skine sa wpa-sec stranice stoji neiskorištena pa je odlučeno uključiti je u funkcije Pwnagtochija.

Ovakva funkcionalnost zahtijeva kreiranje vlastitog plugina. Plugin za Pwngotchi sustav označava neku dodatnu funkcionalnost koju je opcionalna i može je se uključiti i isključiti kada korisnik to hoće. Plugini se mogu uključiti ili isključiti kroz web sučelje Pwnagotchija ili direktnim modificiranjem datoteke konfiguracije. Kako je Pwnagotchi zamišljen tako da zajednica može doprinositi razvoju sustava onda je kreator odvojio plugine na dva GitHub repozitorija i kako ću ja samostalno napraviti ovaj plugin tada on spada u plugine koje je napravila zajednica na repozitorij *pwnagotchi-plugins-contrib*. Za započeti izradu plugina trebam na svome računalu klonirati navedeni repozitorij za plugine i kreirati .py datoteku s kodom poboljšanja, te kreirati popratnu datoteku konfiguracije u .toml formatu. Datoteka konfiguracije sadrži polja koja korisnik može mijenjati i utjecat će na funkcioniranje plugina, a sve datoteke konfiguracije sadrže polje koje govori pluginu da se aktivira ili deaktivira. Vlastiti plugin imena *wpa-sec-list* nema dodatne parametre u konfiguraciji već samo parametar za aktivaciju. Datoteka konfiguracije u .toml formatu za vlastiti plugin sadrži sljedeće podatke:

```
main.plugins.wpa-sec-list.enabled = false
```

Ovaj plugin u suštini čini dvije stvari: učitava podatke iz datoteke i od njih stvara listu svojstava, te kreira i oblikuje tablicu na kojoj će se to prikazati. Za plugin važno je kad se mora njihova funkcija aktivirati, a to znaju prema callback pozivima koje se propagiraju u određenim trenucima tijekom normalne operacije Pwnagotchija. Callback za aktivaciju mog plugina je *onwebhook* koji jedan od mnogih callbacka, a on se propagira kada se ide na web adresu

http://.local:8080/plugins/wpa-sec-list, točnije propagira se da je to webhook samo za wpa-sec-list plugin. Kada se aktivira, plugin otvara datoteku *wpa-sec.cracked.potfile* koja se nalazi u direktoriju gdje se spremaju sva rukovanja tj. u direktoriju /root/handshakes. Datoteka je formatirana tako da je jedan red vezan za jedan AP i pripadajuću šifru, a sadrži podatke: SSID APa, BSSID tj. MAC adresu APa, MAC adresu klijenta, te šifru. Plugin otvara datoteku i stvara listu rječnika koji sadrži ime polja i njegovu vrijednost. Kada se stvori lista s podacima stvara se prikaz stranice prema HTMLu kojemu je proslijeđena lista s podacima. Kod za obradu podataka iz datoteke:

```
def on_webhook(self, path, request):
        if not self.ready:
            return "Plugin not ready"

        if path == "/" or not path:
            try:
                passwords = []
                with open(self.config['bettercap']['handshakes'] + "/wpa-sec.cracked
                    .potfile") as file_in:
                    for line in file_in:
                        fields = line.split(":")
                        password = {
                            "ssid": fields[2],
                            "bssid": fields[0],
                            "clientStation": fields[1],
                            "password": fields[3]
                        }
                        passwords.append(password)
                return render_template_string(TEMPLATE,
                                              title="Passwords list",
                                              passwords=passwords)
            except Exception as e:
                logging.error("[wpa-sec-list] error while loading passwords: %s" % e
                    )
                logging.debug(e, exc_info=True)
```

Drugi dio plugina je HTML kod tablice i JavaScript kod koji čine predložak kojeg Python kod popuni s podacima i prikaže. Tablica je obična HTML tablica s četiri stupca, svaki za jednu vrstu podataka i na vrhu stranice je tekstualno polje za pretragu prema SSIDu APa. U JavaScript dijelu predloška nalazi se funkcija prema vrijednosti SSID polja retka u tablici "filtrira" prikazane redove tj. postavlja CSS vrijednost *display* na nevidljivo za one redove koji ne odgovaraju traženom. Tablica je formatirana preko CSSa tako da postoje dvije vrste prikaza, jedan za široke ekrane, a jedan za male ekrane kao što su oni od mobitela. U mobitel načinu gledanja tablica je postavljena horizontalno, a inače je postavljena vertikalno. Predložak za tablicu:

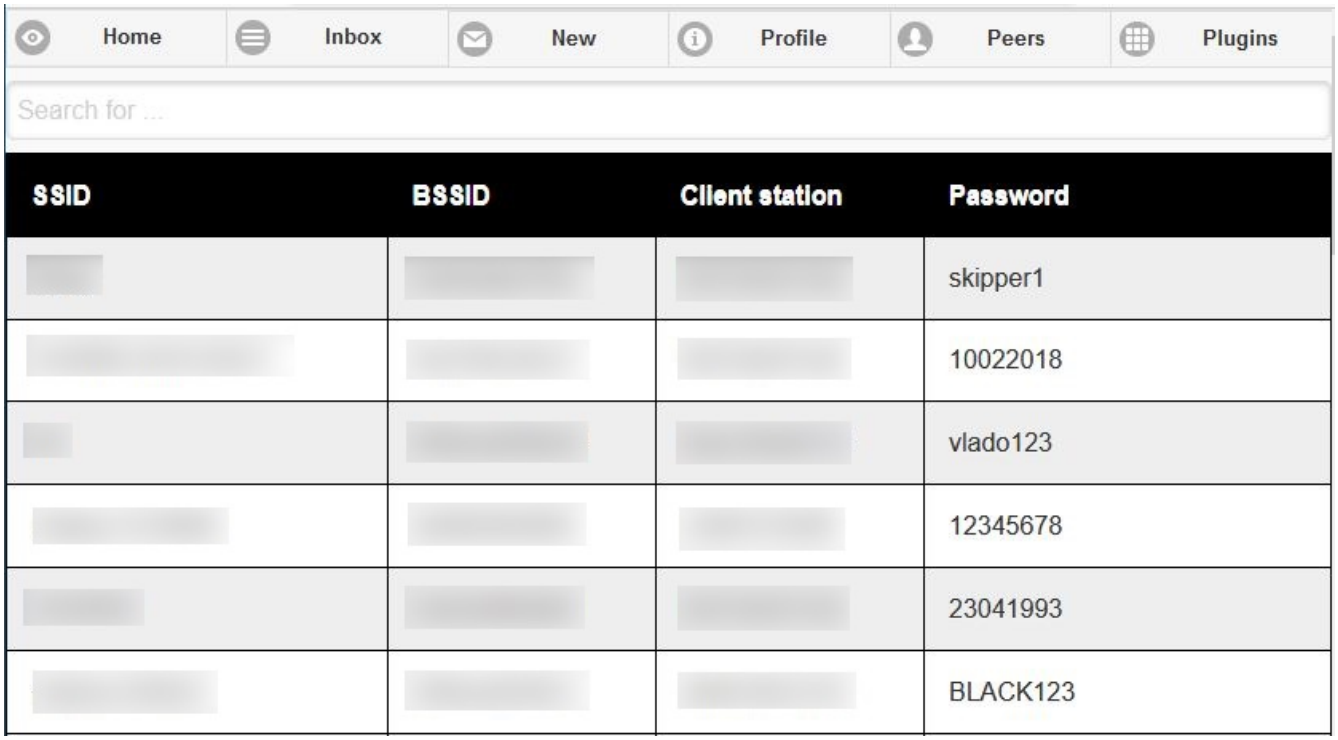
```
{% extends "base.html" %}
{% set active_page = "passwordsList" %}

{% block title %}
    {{ title }}
{% endblock %}
```

```
{% block meta %}
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, user-scalable=0" />
{% endblock %}

{% block styles %}
{{ super() }}
    <style>

        #searchText {
            width: 100%;
        }

        table {
            table-layout: auto;
            width: 100%;
        }

        table, th, td {
            border: 1px solid black;
            border-collapse: collapse;
        }

        th, td {
            padding: 15px;
            text-align: left;
        }

        table tr:nth-child(even) {
            background-color: #eee;
        }

        table tr:nth-child(odd) {
            background-color: #fff;
        }

        table th {
            background-color: black;
            color: white;
        }

        @media screen and (max-width:700px) {
            table, tr, td {
                padding:0;
                border:1px solid black;
            }

            table {
                border:none;
            }

            tr:first-child, thead, th {
```

```
                display:none;
                border:none;
            }

            tr {
                float: left;
                width: 100%;
                margin-bottom: 2em;
            }

            table tr:nth-child(odd) {
                background-color: #eee;
            }

            td {
                float: left;
                width: 100%;
                padding:1em;
            }

            td::before {
                content:attr(data-label);
                word-wrap: break-word;
                background-color: black;
                color: white;
                border-right:2px solid black;
                width: 20%;
                float:left;
                padding:1em;
                font-weight: bold;
                margin:-1em 1em -1em -1em;
            }
        }
    </style>
{% endblock %}
{% block script %}
    var searchInput = document.getElementById("searchText");
    searchInput.onkeyup = function() {
        var filter, table, tr, td, i, txtValue;
        filter = searchInput.value.toUpperCase();
        table = document.getElementById("tableOptions");
        if (table) {
            tr = table.getElementsByTagName("tr");

            for (i = 0; i < tr.length; i++) {
                td = tr[i].getElementsByTagName("td")[0];
                if (td) {
                    txtValue = td.textContent || td.innerText;
                    if (txtValue.toUpperCase().indexOf(filter) > -1) {
                        tr[i].style.display = "";
                    }else{
                        tr[i].style.display = "none";
                    }
```

```
                }
            }
        }
    }

{% endblock %}

{% block content %}
    <input type="text" id="searchText" placeholder="Search for ..." title="Type in a
        filter">
    <table id="tableOptions">
        <tr>
            <th>SSID</th>
            <th>BSSID</th>
            <th>Client station</th>
            <th>Password</th>
        </tr>
        {% for p in passwords %}
            <tr>
                <td data-label="SSID">{{p["ssid"]}}</td>
                <td data-label="BSSID">{{p["bssid"]}}</td>
                <td data-label="Client station">{{p["clientStation"]}}</td>
                <td data-label="Password">{{p["password"]}}</td>
            </tr>
        {% endfor %}
    </table>
{% endblock %}
```

Izgled plugina kada se otvori na računalu tj. u širokom prikazu:



Slika 15: Prikaz wpa-sec-list plugina (Izvor: Osobna izrada)

Kao što se može vidjeti iz slike uspješno sam napravio plugin i radi kako je zamišljeno. Za plugine s drugog repozitorija nije potrebno stvarati novu sliku Pwnagotchija, nego se taj repozitorij klonira direkt na Pwnagotchi i podesi se u glavnoj konfiguracijskoj datoteci lokacija ostalih plugina. Tako sam uspio na svome Pwnagotchiju isprobati svoj plugin. Za kraj napravljen je pull request na repozitoriju *pwnagotchi-plugins-contrib* kojeg nadgledaju isti ljudi kao i temeljni Pwnagotchi repozitorij. U trenutku pisanja nije pregledan i odobren pull request. Zbog sličnog iskustva s pull requestom za prijevod na hrvatski misli ses da ljudi koji rade na ovom projektu uključujući i kreatora nisu jako ažurni s pregledavanjem pull requestova dobrim dijelom zato, jer je ovo neprofitni projekt što oni rade iz hobija.

6.3. Poboljšanja Wigle plugina

Ideja za poboljšanje ovog plugina došla je čitajući prijedloge za poboljšanja na GitHub stranici Pwnagotchija. Tamo je osoba predložila točno ovo, unaprjeđivanje wigle plugina da može uzeti podatke iz ostalih formata za prikaz pozicije na karti koje Pwnagotchi podržava. U originalnom obliku wigle plugin podržava uzimanje podataka samo iz datoteka s ekstenzijom .gps.json. Ta se datoteka dobije ako se koristi *gps* plugin koji u momentu hvatanja rukovanja zapiše poziciju Pwnagotchija. Ograničavajuća činjenica je što za gps plugin treba imati na Pwnagotchi spojen kompatibilan samostalni GPS lokator. Druge dvije datoteke za pozicioniranje su datoteke s ekstenzijom .geo.json i .paw-gps.json.

Datoteka .geo.json stvara se kad je aktivan plugin *net-pos* koji koristi pozicije drugih WiFi APova, antena za mobilni signal i bluetooth uređaja za čiju se poziciju zna da se prema jačini signala od tih znanih pozicija otprilike utvrdi lokacija Pwnagotchija. Kada se Pwnagotchi spoji na internet onda se koristi *Mozilla Location Service* da se izračuna otprilike pozicija prema zabilježenih jačinama signala. Primjer zapisa u .geo.json datoteci:

```
{"location": {"lat": 45.8293, "lng": 15.9793}, "accuracy": 25000.0, "fallback": "ipf
    ", "ts": 1627227487}
```

Datoteke .paw-gps.json stvaraju se kada se koristi *paw-gps* plugin koji dolazi originalno sa Pwnagotchijem. Taj plugin koristi mobilnu Android aplikaciju po imenu *PAW Server for Android* aplikacija koja omogućava podizanje poslužitelja na vlastitom mobitelu ili na drugom Android uređaju. Ta aplikacija sadrži mnoge alate, ali nas zanima mogućnost da locira mobitel ako postoji veza prema internetu. Za doći do gps podataka potrebno je kreirati .xhtml datoteku na mobitelu u direktoriju gdje se nalazi aplikacija koja sadržava kod koji će od aplikacije uzeti podatke i vratiti ih u željenom obliku. Plugin *paw-gps* radi GET poziv prema mobitelu prema kreiranoj datoteci koja će onda vratiti željene podatke ako ih PAW poslužitelj može dati. Još jedan zahtjev tu je da Pwnagotchi mora biti spojen preko bluetootha na mobitel inače nije moguće pristupiti datoteci na mobitelu. Plugin dohvaća podatke o lokaciji kada se uhvati rukovanje i sprema ih na datoteku s ekstenzijom .paw-gps.json. Primjer zapisa u .paw-gps.josn datoteci:

```
{"Updated": "2019-11-14T12:26:35.09589078+01:00", "Latitude": 0.0000, "Longitude":
    0.0000, "NumSatellites": 0, "Altitude": 0}
```

Wigle plugin funkcionira tako da nakon aktivacije čeka *on internet available* poruku koja se propagira nakon što Pwnagotchi ostvari konekciju s internetom. Kada se ostvari konekcija s internetom wigle plugin prvo skuplja konfiguracijske podatke i skuplja imena datoteka sa gps podacima koja se ne ignoriraju ili već jesu prijavljena. Dio koda koji to radi:

```
config = agent.config()
display = agent.view()
reported = self.report.data_field_or('reported', default=list())
handshake_dir = config['bettercap']['handshakes']
all_files = os.listdir(handshake_dir)
all_gps_files = [os.path.join(handshake_dir, filename)
                 for filename in all_files
                 if filename.endswith('.gps.json') or filename.endswith('.paw-gps.
                     json') or filename.endswith('.geo.json')]

all_gps_files = remove_whitelisted(all_gps_files, self.options['whitelist'])
new_gps_files = set(all_gps_files) - set(reported) - set(self.skip)
```

Vlastita modifikacija ovdje je unutar if petlje, dodan je dio koda koji dopušta uključivanje datoteka s nastavcima .paw-gps.json i .geo.json. Sljedeći korak u obradi je provjera ako postoje nove datoteke koje se ne ignoriraju ili jesu već prijavljene i ako ima novih datoteka ide se u njihovu obradu. Za svaku datoteku sa gps podacima traži se pripadajuća datoteka s podacima o rukovanju, ako takve datoteke nema prelazi se na sljedeću gps datoteku. Dio koda koji to radi:

```
if new_gps_files:
    logging.info("WIGLE:␣Internet␣connectivity␣detected.␣Uploading␣new␣handshakes␣to
        ␣wigle.net")
    csv_entries = list()
    no_err_entries = list()
    for gps_file in new_gps_files:
        if gps_file.endswith('.gps.json'):
            pcap_filename = gps_file.replace('.gps.json', '.pcap')
        if gps_file.endswith('.paw-gps.json'):
            pcap_filename = gps_file.replace('.paw-gps.json', '.pcap')
        if gps_file.endswith('.geo.json'):
            pcap_filename = gps_file.replace('.geo.json', '.pcap')
        if not os.path.exists(pcap_filename):
            logging.debug("WIGLE:␣Can't␣find␣pcap␣for␣%s", gps_file)
            self.skip.append(gps_file)
            continue
```

Nakon što je stavljeno da se uzimaju u obzir datoteke s novim nastavcima potrebno je ovdje bilo ugraditi još dvije if petlje koje se odnose na nove nastavke tako da se ispravno mogu naći .pcap datoteke. Nakon toga izvlače se podaci iz datoteka sa gps podacima i provjerava se ako postoje podaci o geografskoj dužini i širini u tim podacima, ako ne postoje onda se preskače ova i prelazi na obradu sljedeće gps datoteke. Kod koji obavlja ovaj postupak:

```
try:
    gps_data = _extract_gps_data(gps_file)
except OSError as os_err:
    logging.debug("WIGLE:␣%s", os_err)
```

```
    self.skip.append(gps_file)
    continue
except json.JSONDecodeError as json_err:
    logging.debug("WIGLE:␣%s", json_err)
    self.skip.append(gps_file)
    continue
if gps_data['Latitude'] == 0 and gps_data['Longitude'] == 0:
    logging.debug("WIGLE:␣Not␣enough␣gps-information␣for␣%s.␣Trying␣again␣next␣time.
        ", gps_file)
    self.skip.append(gps_file)
    continue
```

Kod funkcije *_extract_gps_data(gps_file)*:

```
def _extract_gps_data(path):
    """
    Extract data from gps-file

    return json-obj
    """

    try:
        if path.endswith('.geo.json'):
            with open(path, 'r') as json_file:
                tempJson = json.load(json_file)
                d = datetime.utcfromtimestamp(int(tempJson["ts"]))
                return {"Latitude": tempJson["location"]["lat"], "Longitude":
                    tempJson["location"]["lng"], "Altitude": 10, "Updated": d.
                    strftime('%Y-%m-%dT%H:%M:%S.%f')}
        else:
            with open(path, 'r') as json_file:
                return json.load(json_file)
    except OSError as os_err:
        raise os_err
    except json.JSONDecodeError as json_err:
        raise json_err
```

Modifikacije koje su bile potrebne u ovoj funkciji je if petlja za gps datoteke s ekstenzijom .geo.json. If petlja je potrebna jer .geo.json datoteka nije formatirana isto kao i .paw-gps.json ili .gps.json. U .geo.json datoteci formatiranje je drugačije tako da se podaci o geografskoj širini i dužini ne nalaze pod identifikatorom "Latitude" i "Longitude" nego pod identifikatorima "lat" i "long" unutar dodatnog objekta. Također datum i vrijeme zapisa podatka nije u UTC obliku nego u UNIX timestamp obliku, te taj podatak treba konvertirati u pravilni obliku. Ova konverzija podataka potrebna je jer je lakše u ovoj momentu obrade izvući potrebne podatke iz datoteke i ispravno ih formatirati da kasnije u obradi ne treba prilagođavati funkciju obrade podataka. Za kasniju obradu podataka potreban je jedan objekt sa gps podacima sljedećeg oblika:

```
{"Latitude": 0.0000, "Longitude": 0.0000, "Altitude": 0, "Updated": "2019-11-14T12
    :26:35.09589078"}
```

Sljedeći korak u obradi je izvlačenje podataka iz .pcap datoteke gdje su zapisani podaci o APu koji se prijavljuje i pakiranje gps podataka s podacima o APu u .csv oblik, te zapisivanje tog csv

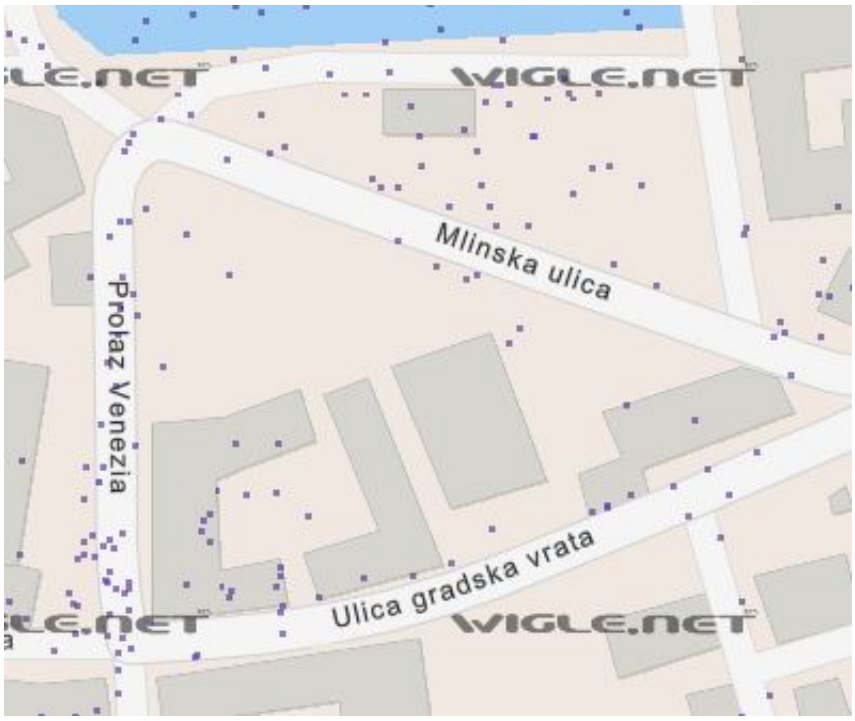
retka u listu redaka koji će se kasnije slati. Dio koda koji to radi:

```
try:
    pcap_data = extract_from_pcap(pcap_filename, [WifiInfo.BSSID,
                                                  WifiInfo.ESSID,
                                                  WifiInfo.ENCRYPTION,
                                                  WifiInfo.CHANNEL,
                                                  WifiInfo.RSSI])
except FieldNotFoundError:
    logging.debug("WIGLE:␣Could␣not␣extract␣all␣information.␣Skip␣%s", gps_file)
    self.skip.append(gps_file)
    continue
except Scapy_Exception as sc_e:
    logging.debug("WIGLE:␣%s", sc_e)
    self.skip.append(gps_file)
    continue
new_entry = _transform_wigle_entry(gps_data, pcap_data, self.__version__)
csv_entries.append(new_entry)
no_err_entries.append(gps_file)
```

Zadnji dio obrade je provjera ako ima uspješno formatiranih redaka za slanje na wigle stranicu i ako postoje onda se prolazi kroz listu redaka koji se pakiraju za slanje. Ako sve dobro prođe javi se poruka o uspješno slanju podataka i poslani podaci se zapisuju u listu poslanih podataka. Kod koji je zaslužan za to:

```
if csv_entries:
    display.on_uploading('wigle.net')

    try:
        _send_to_wigle(csv_entries, self.options['api_key'], donate=self.options['
            donate'])
        reported += no_err_entries
        self.report.update(data={'reported': reported})
        logging.info("WIGLE:␣Successfully␣uploaded␣%d␣files", len(no_err_entries))
    except requests.exceptions.RequestException as re_e:
        self.skip += no_err_entries
        logging.debug("WIGLE:␣Got␣an␣exception␣while␣uploading␣%s", re_e)
    except OSError as os_e:
        self.skip += no_err_entries
        logging.debug("WIGLE:␣Got␣the␣following␣error:␣%s", os_e)

    display.on_normal()
```

Prije nego se mogu početi slati podaci na wigle web stranicu potrebno je registrirati se na njihovu stranicu kako bi se dobio API ključ korisnika. Taj ključ služi za identifikaciju prema poslužitelju tako da ne može svatko slati podatke na wigle stranicu, ali tako se identificira i tko je poslao podatke na poslužitelj. Prema tom ključu onda mogu filtrirati samo one APove koje sam ja otkrio. API ključ potrebno je ubaciti u konfiguracijsku datoteku Pwnagotchija kako bi plugin mogao ispravno raditi.



Slika 16: Wigle karta svih APova u dijelu Novigrada (Izvor: Wigle, 2021)

Slika 17: Bliži prikaz wigle karte s imenima APova (Izvor: Wigle, 2021)

Na slikama gore koje prikazuju kartu dijela Novigrada sa wigle web stranice APovi označeni su ljubičastim kvadratićima i na toj slici prikazani su svi APovi koji su poslani na wigle stranicu. Na drugoj slici vidi se prikaz imena APova i taj prikaz dobije se kada se više približi slika na web stranici, a svi APovi koji se vide na toj približenoj slici su oni koji su s ovim Pwnagotchijem otkriveni. Sve podatke koji su poslani na wigle i koji se vide u drugoj slici dobiveni su iz datoteka vrste .geo.json kroz poboljšanja koja su napravljena na wigle pluginu. Pull request prema originalnom Pwnagotchi repozitoriju je napravljen, ali u trenutku pisanja nije potvrđen od strane voditelja repozitorija.

7. Zaključak

Tema ovog diplomskog rada je Pwnagotchi sustav koji se sastoji od više programa povezanih u jednu cjelinu koja radi na hardverskoj podlozi Raspberry Pi malih računala. Rad uključuje instalaciju, korištenje i davanje doprinosa Pwnagotchi zajednici putem razvoja dodatnih funkcionalnosti ili na neki drugi način. U radu je objašnjena hardverska i softverska podloga Pwnagotchija, proces instalacije, konfiguracije i korištenja Pwnagotchija, te proces razvoja i testiranje novih funkcija za Pwnagotchi.

Kroz iskustvo korištenja Pwnagotchija, ali i instalacije i konfiguracije osvrnuo bih se na nekoliko stvari koje su primijećena i koje su utjecale na korištenje Pwnagotchija tijekom izrade ovog rada. Instalacija i konfiguracija Pwnagotchija je vrlo jednostavna, jedini preduvjet je imati pravu opremu i uz pratnju uputa jednostavno se uspije dobiti funkcionalni Pwnagotchi, a za bilo kakve probleme koji se mogu pojaviti ima dovoljno podrške od kreatora i od zajednice. Postoje doduše neki problemi s određenim pluginima ili verzijama Pwnagotchija, ali oni su poznati i zapisani pa nije teško naći rješenje ili kako zaobići taj problem. Korištenje Pwnagotchija je također vrlo lako, točnije korisnik nema utjecaja u njegovoj osnovnoj funkciji, a to je skupljanje WPA rukovanja. To Pwnagotchi radi dobro i od korisnika se samo očekuje da radi male izmjene u postavkama za lijenost kako Pwnagotchi postaje iskusniji. To međutim dovodi korisnika da relativno brzo dođe do točke gdje nema puno interakcije sa svojim Pwngotchijem. Plugini koji su razvijeni tu donekle pomažu i nude dodatne opcije, a i činjenica da je za osobu koja ima iskustva s programiranjem i Pythonom može napraviti svoj plugin sa svojom funkcionalnošću. Kod izrade plugina dolazi se do više činjenice kojih korisnik mora biti svjestan. Prvo je da je Pwnagotchi zamišljen da bude implementiran na Raspberry Pi Zero W računalu koje ima dosta slabe hardverske performanse, višestruko slabije nego današnji prosječni mobitel, ako se koristi RPi 3 ili 4 to se donekle popravlja, ali Pwnagotchi nije najbolje prilagođen za ta računala. Sljedeća činjenica je da je jako teško raditi debug Python skripte koja se koristi na Pwnagotchiju gdje sam ja koristi log datoteku i postavljao ručno mjesta ispisa u kodu. Zadnja činjenica je da u trenutku pisanja ovog rad podrška za Pwnagtochi opada, pretpostavlja se da nema više toliko interesa za projekt od strane kreatora i zajednice, što znači da bilo kakvi pull requesti prema Pwnagtochi repozitorijima čekaju dugo za potvrdu ili komentare i to značajno usporuje razvoj. Jedna od mogućnosti koju se provjeravalo kroz rad je mogućnost probijanja šifri sa Pwnagtochijem, što je brzo otkriveno da je nemoguće isključivo zbog niskih performansi hardvera. Postoje plugini koji su doskočili tom problemu i učinkovitiji su bili nego prema početnim procjenama po broju probijenih šifri. Tome je pridonijela činjenica da je oko trećina šifri koje su bile uhvaćene prejednostavna i ne prati osnovne formate kako se formiraju sigurne šifre.

Smatram da je Pwnagtochi odličan projekt za osobu koju zanima WiFi sigurnost, a još više je zanima rad s Raspberry računalima. U sklopu ovog rada nisu rađene izmjene na RPi pločici, ali korisnik koji bi htio može lemiti i dodavati stvari na nju. Osobu koju zanima programiranje u Pythonu isto može uživati u ovom projektu, jer ima još prostora za razvoj funkcionalnosti. Iz ovog projekta može se naučiti nešto o RLu i AIu, ali je teško pratiti putem log datoteke kako Pwnagotchi uči. Moje mišljenje je da je ovo zanimljiv projekt za pokušati implementirati i pokušati poboljšati.

Popis literature

[1] V. Beal. (2021.). „What is WiFi," adresa: https://www.webopedia.com/definitions/wifi/ (pogledano 6. 7. 2021.).

[2] NCTA. (2018.). „Wi-Fi: How Broadband Households Experience the Internet," adresa: https://www.ncta.com/whats-new/wi-fi-how-broadband-households-experience-the-internet (pogledano 6. 7. 2021.).

[3] D. Meraj. (2020.). „Ethernet vs WiFi," adresa: https://www.tmcnet.com/topics/articles/2020/01/27/444323-ethernet-vs-wifi.htm (pogledano 6. 7. 2021.).

[4] M. D. Aime, G. Calandriello i A. Lioy, „Dependability in Wireless Networks: Can We Rely on WiFi?" *IEEE Security & Privacy*, sv. 5, br. 1, str. 23–29, 2007.

[5] M. Mwikali. (2020.). „Capturing and Cracking WPA Handshake using Aircrack-ng," adresa: https://shehackske.medium.com/capturing-and-cracking-wpa-handshake-using-aircrack-ng-d9496f30c7c3 (pogledano 6. 7. 2021.).

[6] evilsocket. (2019.). „Weaponizing and Gamifying AI for WiFi Hacking: Presenting Pwnagotchi 1.0.0," adresa: https://www.evilsocket.net/2019/10/19/Weaponizing-and-Gamifying-AI-for-WiFi-Hacking-Presenting-Pwnagotchi-1-0-0/ (pogledano 6. 7. 2021.).

[7] ——, (2019.). „Pwnagotchi: Deep Reinforcement Learning for WiFi pwning: Introduction," adresa: https://pwnagotchi.ai/intro/ (pogledano 6. 7. 2021.).

[8] ——, (2019.). „Pwnagotchi: Deep Reinforcement Learning for WiFi pwning: Installation," adresa: https://pwnagotchi.ai/installation/ (pogledano 11. 7. 2021.).

[9] R. Pi. (2021.). „Raspberry Pi Foundation - About Us," adresa: https://www.raspberrypi.org/about/ (pogledano 11. 7. 2021.).

[10] R. Cellan-Jones. (2011.). „A 15 pound computer to inspire young programmers," adresa: https://www.bbc.co.uk/blogs/thereporters/rorycellanjones/2011/05/a_15_computer_to_inspire_young.html (pogledano 11. 7. 2021.).

[11] I. H. (2017.). „Što je to SoC," adresa: https://pcchip.hr/helpdesk/sto-je-to-soc/ (pogledano 13. 7. 2021.).

[12] M. Richardson i S. Wallace, *Getting Started with Raspberry Pi*. Maker Media, 2013., str. 2–8.

[13] R. Pi. (2021.). „Raspberry Pi Zero W," adresa: https://www.raspberrypi.org/products/raspberry-pi-zero-w/ (pogledano 13. 7. 2021.).

[14] R. Barnes. (2016.). „Raspberry Pi 3: Specs, benchmarks & testing," adresa: https://magpi.raspberrypi.org/articles/raspberry-pi-3-specs-benchmarks (pogledano 13. 7. 2021.).

[15] R. Pi. (2021.). „Raspberry Pi 4 Tech Specs," adresa: https://www.raspberrypi.org/products/raspberry-pi-4-model-b/specifications/ (pogledano 13. 7. 2021.).

[16] eLinux.org. (2019.). „RPiconfig," adresa: https://elinux.org/RPiconfig (pogledano 13. 7. 2021.).

[17] PixelLytical.com. (2021.). „Raspberry Pi Zero With Pre-Soldered Header WH (Wireless)," adresa: http://pixellytical.com/product/raspberry-pi-zero-with-pre-soldered-header-wh-wireless/ (pogledano 13. 7. 2021.).

[18] J. S. Domingo. (2017.). „Raspberry Pi Zero W Review," adresa: https://www.pcmag.com/reviews/raspberry-pi-zero-w (pogledano 13. 7. 2021.).

[19] Alzashop.com. (2021.). „Raspberry Pi 4 Model B - 2GB RAM," adresa: https://www.alzashop.com/raspberry-pi-4-model-b-2gb-ram-d5655285.htm (pogledano 15. 7. 2021.).

[20] evilsocket. (2021.). „Bettercap: Introduction," adresa: https://www.bettercap.org/intro/ (pogledano 18. 7. 2021.).

[21] ——, (2021.). „Bettercap: Modules/WiFi," adresa: https://www.bettercap.org/modules/wifi/ (pogledano 18. 7. 2021.).

[22] ——, (2021.). „Bettercap: Modules/Bluetooth LE," adresa: https://www.bettercap.org/modules/ble/ (pogledano 18. 7. 2021.).

[23] ——, (2021.). „Bettercap: Modules/HID on 2.4GHz," adresa: https://www.bettercap.org/modules/hid/ (pogledano 18. 7. 2021.).

[24] ——, (2021.). „Bettercap: Modules/IPV4/IPV6," adresa: https://www.bettercap.org/modules/ethernet/ (pogledano 18. 7. 2021.).

[25] d3ad R1nger. (2021.). „Deauthentication Attack using Kali Linux," adresa: https://sudorealm.com/blog/deauthentication-attack-using-kali-linux (pogledano 18. 7. 2021.).

[26] N. Haitham Ameen, A. Shahidan M. i M. Haydar Imad, „An Automated Approach to Detect Deauthentication and Disassociation Dos Attacks on Wireless 802.11 Networks," *International Journal of Computer Science*, sv. 12, br. 4, str. 23–29, 2015.

[27] Wifi-professionals. (2019.). „4-Way Handshake," adresa: https://www.wifi-professionals.com/2019/01/4-way-handshake (pogledano 18. 7. 2021.).

[28] evilsocket. (2019.). „Pwning WPA/WPA2 Networks With Bettercap and the PMKID Client-Less Attack," adresa: https://www.evilsocket.net/2019/02/13/Pwning-WiFi-networks-with-bettercap-and-the-PMKID-client-less-attack/ (pogledano 18. 7. 2021.).

[29] I. C. Education. (2020.). „Artificial Intelligence (AI)," adresa: https://www.ibm.com/cloud/learn/what-is-artificial-intelligence (pogledano 4. 8. 2021.).

[30] R. Gilman. (2018.). „Intuitive RL: Intro to Advantage-Actor-Critic (A2C),” adresa: https://hackernoon.com/intuitive-rl-intro-to-advantage-actor-critic-a2c-4ff545978752 (pogledano 4. 8. 2021.).

[31] T. Simonini. (2018.). „An intro to Advantage Actor Critic methods: let's play Sonic the Hedgehog!” Adresa: https://www.freecodecamp.org/news/an-intro-to-advantage-actor-critic-methods-lets-play-sonic-the-hedgehog-86d6240171d/ (pogledano 4. 8. 2021.).

[32] T. Keary. (2021.). „PCAP: Packet Capture, what it is & what you need to know,” adresa: https://www.comparitech.com/net-admin/pcap-guide/ (pogledano 24. 7. 2021.).

[33] A. Ku. (2011.). „Wi-Fi Security: Cracking WPA With CPUs, GPUs, And The Cloud,” adresa: https://www.tomshardware.com/reviews/wireless-security-hack,2981-5.html (pogledano 24. 7. 2021.).

[34] W. J. Buchanan. (2011.). „WPA-2 Hash Cracking,” adresa: https://asecuritysite.com/encryption/ssid_hm (pogledano 24. 7. 2021.).

[35] CloudFlare. (2021.). „What is a brute force attack?” Adresa: https://www.cloudflare.com/learning/bots/brute-force-attack/ (pogledano 25. 7. 2021.).

[36] hashcat. (2021.). „Brute-Force Attack,” adresa: https://hashcat.net/wiki/doku.php?id=brute_force_attack (pogledano 25. 7. 2021.).

[37] Lockdown.co.uk. (2009.). „Password Recovery Speeds,” adresa: https://web.archive.org/web/20180412051235/http://www.lockdown.co.uk/?pg=combi&s=articles (pogledano 25. 7. 2021.).

[38] OnlineHashCrack. (2021.). „About / Pricing,” adresa: https://www.onlinehashcrack.com/about-pricing.php (pogledano 25. 7. 2021.).

[39] B. Vigliarolo. (2018.). „Brute force and dictionary attacks: A cheat sheet,” adresa: https://www.techrepublic.com/article/brute-force-and-dictionary-attacks-a-cheat-sheet/ (pogledano 25. 7. 2021.).

[40] hashcat. (2021.). „Rule-based Attack,” adresa: https://hashcat.net/wiki/doku.php?id=rule_based_attack (pogledano 25. 7. 2021.).

[41] NotSoSecure. (2017.). „One Rule to Rule Them All,” adresa: https://notsosecure.com/one-rule-to-rule-them-all/ (pogledano 25. 7. 2021.).

[42] Wpa-sec. (2021.). „My nets,” adresa: https://wpa-sec.stanev.org/?my_nets (pogledano 27. 7. 2021.).

[43] OnlineHashCrack. (2021.). „My dashboard,” adresa: https://www.onlinehashcrack.com/wpa (pogledano 27. 7. 2021.).

[44] evilsocket. (2019.). „Pwnagotchi: Deep Reinforcement Learning for WiFi pwning: Contributing,” adresa: https://pwnagotchi.ai/contributing/ (pogledano 30. 7. 2021.).

[45] Wigle. (2021.). „Map,” adresa: https://wigle.net/map (pogledano 2. 8. 2021.).

Popis slika

Popis kratica

RPi	Raspberry Pi
RPi0W	Raspberry Pi Zero W
WiFi	Wireless Fidelity
WPA	WiFi Protected Acccess
IEEE	Institute of Electrical and Electronics Engineers
A2C	Actor-Advantage-Critic
SoC	System on a chip
GPIO	General purpose input-output
OS	Operacijski sustav
AP	Access point
DoS	Denial of Service
MAC	Media access control address
HID	Human interface device
BLE	Bluetooth low energy
PMK	Pairwise master key
RSN	Remote Security Network
SSID	Service Set Idetifier
PBKDF	Password-Based Key Derivation Function
SHA	Secure Hash Algorithm
AI	Artificial Intelligence
DL	Deep learning
RL	Reinforcement learning