ELK SKLAD ZA HEKERJE

Milan Gabor, Danijel Grah

Viris d.o.o.
e-pošta: milan@viris.si, danijel.grah@viris.si
URL: http://www.viris.si

Povzetek: *ELK sklad je sodobna rešitev zbirke orodij za obvladovanje dnevniških datotek. V prispevku bodo predstavljene posamezne komponente in predlagani načini uporabe ELK sklada na področju informacijske varnosti. Bolj natančno bo predstavljena rešitev za vizualizacijo prometa odjemalcev in dostopnostih točk v okoliških Wi-Fi omrežjih. Nov pristop bo pripomogel k učinkovitejši analizi prometa in varnosti Wi-Fi omrežij, saj vizualizacija vsebuje tudi časovno komponento in druge enostavnejše načine za vizualizacijo podatkov.*

1. UVOD

Z začetkom informacijske dobe upravljamo s čedalje večjo količino podatkov. Zaradi usmerjenosti v personalizacijo produktov, priljubljenosti družbenih omrežij in na splošno zaradi razmaha informacijsko-komunikacijske tehnologije, pojem masovni podatki (angl. big data) pridobiva na pomenu. Potreba po hranjenju, obdelavi in predstavitvi podatkov narašča s ciljem iz podatkov tvoriti informacije in iz informacij pridobiti znanje, ki ga lahko uporabimo pri poslovnih odločitvah. ELK sklad je ena izmed sodobnih rešitev, ki omogoča hranjenje, analizo in predstavitev velike količine podatkov. Prilagodljivost ELK sklada v smislu integracije in skalabilnosti ponuja tudi veliko možnosti na področju informacije varnosti. V prispevku je po predstavitvi posameznih komponent ELK sklada predstavljena rešitev uporabe ELK sklada za vizualizacijo brezžičnega prometa, ki nastane med odjemalci in dostopnimi točkami v okoliških Wi-Fi omrežjih.

2. ELK

ELK sklad tvorijo tehnologije Elasticsearch, Logstash in Kibana [1]. Elasticsearch je tehnologija, ki se uporablja za hranjenje, iskanje in analizo podatkov, Logstash je namenjen centraliziranju in obdelavi dnevniških datotek, Kibana pa se uporablja za vizualizacijo podatkov. Njihovo povezovanje in kombinacija omogoča obdelavo in hranjenje dnevniških datotek, ter vizualizacijo podatkov z namenom pridobivanja pomembnih informacij iz kopice podatkov. ELK sklad je pogosto uporabljena rešitev za spremljanje dogodkov na nivoju lokalnega omrežja v realnem času. Ustrezna vpeljava ELK sklada je primerljiva s komercialno dostopnimi SIEM (angl. Security information and event management) sistemi. Skrbniki lokalnih omrežij in sistemski administratorji lahko ELK sklad prilagodijo, tako da nudi ustrezne informacije za spremljanje in analizo dogodkov ter učinkovit in hiter odziv v primeru incidentov na nivoju omrežja. Ustrezna integracija tako doprinese k varnosti v lokalnih omrežjih združb, vendar uporabnost ELK sklada ni omejena zgolj na vizualizacijo dogodkov, ki nastanejo v lokalnih omrežjih. Rešitev se lahko aplicira v različnih situacijah in za različne namene.

2.1. Elasticsearch

Elasticsearch je porazdeljena podatkovna zbirka z možnostjo hranjenja, iskanja in analiziranja podatkov v realnem času. Večje korporacije, ki uporabljajo Elasticsearch so: Wikipedia, Guardian, Stack Overflow in GitHub [2]. Elasticsearch je implementiran v programskem jeziku Java nad Apache Lucene (knjižnica, ki se uporablja za »full-text-search«). Poleg funkcionalnosti »full-text-search«, Elasticsearch omogoča porazdeljeno hrambo dokumentov, kjer je vsako polje indeksirano in lahko iščemo po njem. Prav tako

omogoča porazdeljeno iskanje z možnostjo analiziranja v realnem času in veliko stopnjo fleksibilnosti pri prilaganju zmogljivosti glede na potrebe. Funkcionalnost Elasticsearch-a je dosegljiva preko REST (angl. Representational state transfer) API-ja, ki omogoča poizvedovanje in analizo s pomočjo poljubnega odjemalca (brskalnik, ukazna lupina, poljuben programski jezik …). V nadaljevanju je prikazanih nekaj primerov uporabe, kjer z odjemalcem curl v ukazni lupini najprej ustvarimo indeks, nato dodamo podatke in kasneje naredimo poizvedbo nad podatki. Primeri predpostavljajo lokalno dosegljivo in delujočo različico Elasticsearch-a, ki posluša na vratih 9200.

Kreiramo indeks:

curl -XPUT 'http://localhost:9200/ots/'

Pogledamo stanje indeksov:

curl 'http://localhost:9200/_cat/indices?v'

```
health status index               pri rep docs.count docs.deleted store.size pri.store.size
yellow open   logstash-2015.04.20   5   1    1284599            0    444.6mb        444.6mb
yellow open   .kibana               1   1          8           19     58.7kb         58.7kb
yellow open   ots                   5   1          0            0       575b           575b
```

Slika 1: Stanje indeksov

Dodamo podatke:

curl -XPOST "http://localhost:9200/ots/prispevki" -d '{ "naslov" : "ELK sklad za hekerje", "avtor": "Milan Gabor, Danijel Grah" }'

Naredimo poizvedbo:

curl -XGET 'http://localhost:9200/ots/_search?pretty'

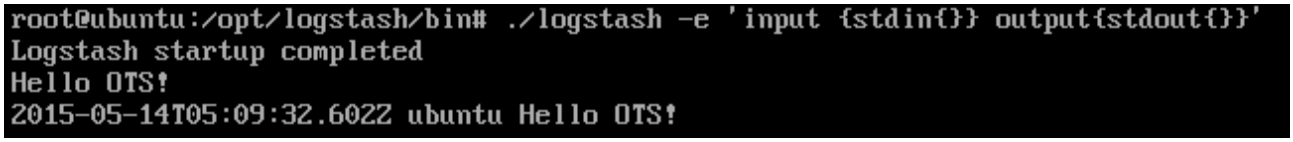

```
{
  "took" : 1,
  "timed_out" : false,
  "_shards" : {
    "total" : 5,
    "successful" : 5,
    "failed" : 0
  },
  "hits" : {
    "total" : 1,
    "max_score" : 1.0,
    "hits" : [ {
      "_index" : "ots",
      "_type" : "prispevki",
      "_id" : "AU1G4Ur0r_IA71mL89MT",
      "_score" : 1.0,
      "_source":{ "naslov" : "ELK sklad za hekerje", "avtor": "Milan Gabor, Danijel Grah" }
    } ]
  }
}
```


Slika 2: Rezultati poizvedbe

Funkcionalnost in uporabnost Elasticsearch-a, kot so na primer točkovanje iskalnih zadetkov, kompleksne poizvedbe (Query DSL) in porazdeljeno delovanje presega okvire tega prispevka. Več informacij o tehnologiji Elasticsearch je zapisanih v viru [3].

2.2. Logstash

Logstash je orodje za zbiranje, razčlenitev in obdelavo vhodnih podatkov iz različnih virov. Sestavljajo ga vhodni vmesniki, filtri, kodeki in izhodni vmesniki. Temelji na programskem jeziku Java in za delovanje potrebuje zgolj virtualno napravo Java. Primer preproste uporabe Logstash-a prikazuje Slika 3, kjer se vhodni niz »Hello OTS!« iz standardnega vhoda ukazne lupine preslika v izhodni niz »Hello OTS!« na standardnem izhodu ukazne lupine.



Slika 3: Osnovna uporaba orodja Logstash

Logstash omogoča širok nabor vira vhodnih in ponora izhodnih podatkov. V vmesnem koraku se podatki obdelajo s pomočjo filtrov. Podatke na vhodu oziroma preden izstopijo lahko obdelamo tudi s pomočjo kodekov. Običajno posamezne razdelke definiramo v nastavitveni datoteki. Primer nastavitvene datoteke je prikazan spodaj.

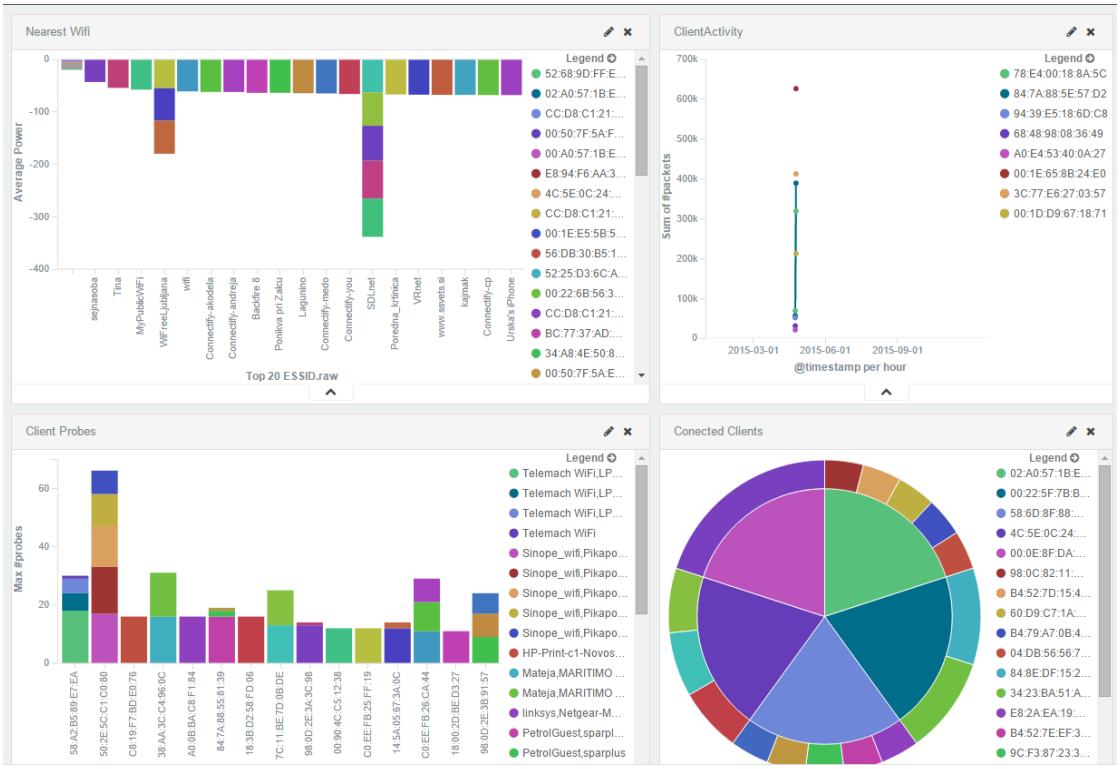
```
input {
  file {
    path => "/tmp /access_log"
    start_position => "beginning"
  }
}
filter {
  if [path] =~ "access" {
    mutate { replace => { "type" => "apache_access" } }
    grok {
      match => { "message" => "%{COMBINEDAPACHELOG}" }
    }
  }
  date {
    match => [ "timestamp" , "dd/MMM/yyyy:HH:mm:ss Z" ]
  }
}
output {
  elasticsearch {
    host => localhost
  }
  stdout { codec => rubydebug }
}
```

Kot razvidno ima nastavitvena datoteka značilno strukturo, ki je input, filter in output. V delu input specificiramo datoteko, ki jo želimo obdelati. Za filter uporabimo vtičnike mutate, grok in date, ki vhodne podatke obdelajo. V output pa definiramo ponor podatkov, ki je v tem primeru standardni izhod in Elastic-search. Strukturiranost, širok nabor vtičnikov in predvsem njihova uporabnost omogoča obdelavo in

normalizacijo praktično poljubnih vhodnih podatkov. Več primerov uporabe in obsežen vir informacij o Logstash-u se nahaja v [4].

2.3. Kibana

Na Kibano lahko gledamo kot na uporabniški vmesnik, ki omogoča iskanje, pregled in analizo podatkov, ki so shranjeni v Elasticsearch-u. Kibana omogoča pregled vhodnih podatkov, enostavno in zahtevnejše iskanje po podatkih, ustvarjanje vizualizacij in sestavljanje slednjih v pregledno ploščo (angl. dashboard). Vsa ta vizualizacija je mogoča brez predznanja programiranja ali drugih potrebnih znanj. Vgrajena podpora agregaciji in grupiranju podatkov poenostavi ustvarjanje kompleksnejših iskalnih zahtev in posledično pohitri postopek kreiranja vizualizacij. Slika 4 prikazuje primer pregledne plošče, ki temelji na prometu odjemalcev in dostopnih točk v okoliških Wi-Fi omrežjih.



Slika 4: Pregledna plošča Kibana

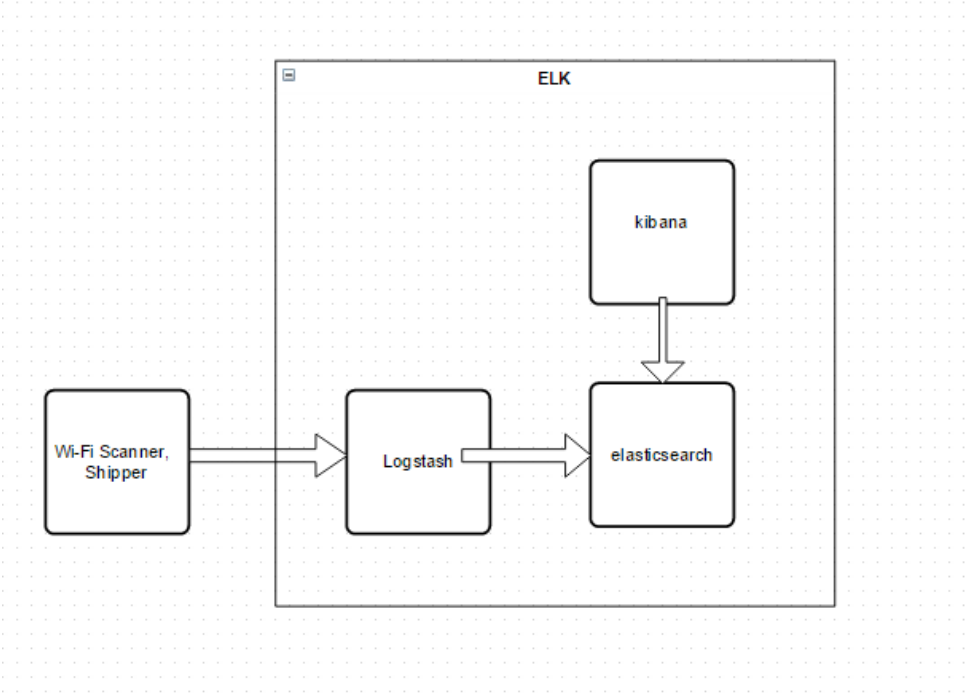
3. ELK SKLAD ZA HEKERJE

Zaradi odprtosti in prilagodljivost ELK sklada zanimanje za integracijo in uporabo naraščа na področju informacijske varnosti. Vpeljava SIEM sistema za spremljanje in nadzor dogajanja v notranjem omrežju je le ena od možnih integracij ELK sklada. Rešitev lahko tudi uporabimo in prilagodimo za spremljanje dogajanja na CTF (angl. Capture The Flag) tekmovanjih, ki se odvijajo v sklopu različnih konferenc na področju informacijske varnosti. V nadaljevanju opisujemo rešitev, ki se lahko uporabi za vizualizacijo zajema Wi-Fi prometa v okolici z namenom učinkovitejše varnostne analize.

3.1. Vizualizacija in analiza Wi-Fi prometa in omrežij

Za testiranje varnostni Wi-Fi omrežij strokovnjaki informacijske varnostni pogosto uporabljajo nabor orodij paketa Aircrack-ng [5]. Aircrarck-ng omogoča spremljanje in analizo prometa med vozlišči (dostopnimi točkami in odjemalci) kot tudi izvajanje aktivnih napadov z namenom ugotavljanje stopnje varnostni

posameznih Wi-Fi omrežij. Pri pasivnem spremljanju prometa se uporabljajo orodja airmon-ng in airodump-ng. S pomočjo airmon-ng postavimo kartico v stanje nadzora, kjer je možno spremljati promet z airodump-ng, ki omogoča zajem prometa paketov, ki so zanimivi za analizo. Rezultati so shranjeni v datotekah različnih formatov. Kot primer lahko navedemo datoteko s končnico .pcap, ki vsebuje celotni promet v času izvajanja zajema in datoteko s končnico .CSV, ki vsebuje trenutno stanje zajema (stanje prometa v določenem trenutku). Izhodne podatke orodja airodump-ng lahko primerjamo z dnevniškimi datotekami. Za vizualizacijo podatkov pa je potrebnih nekaj dodatnih orodij in pristopov. Omeniti velja orodje Airodump-NG Scan Visualizer [6], ki omogoča nalaganje CSV datotek, tabelarični prikaz vrednosti, sortiranje in grafični prikaz. Kot omejitve bi lahko izpostavili naravo CSV datoteke, ki vsebuje podatke o zajemu prometa v nekem trenutku. Tako grafični predstavitvi podatkov manjka časovna komponenta. Naš prispevek na področju vizualizacije Wi-Fi prometa je povezava ELK sklada z orodjem airodump-ng. Arhitekturno postavitev rešitve za vizualizacijo prikazuje Slika 5.



Slika 5: Arhitekturna postavitev rešitve za vizualizacijo

3.1.1. Prilagajanje komponente Wi-Fi Scanner

V vlogi komponente Wi-Fi Scanner smo uporabili prilagojeno različico orodja airodump-ng. Spremembe, ki so bile dodane v orodje vključujejo zapis podatkov zajema v obliki JSON in spremembe omejitve maksimalnega števila povpraševanj na odjemalca (angl. probes).

Odjemalci pogosto (odvisno od izvedbe nastavitev in stanja) pošiljajo povpraševanja s katerimi ugotavljajo ali je omrežje v bližini. Naprava, kot je na primer telefon, ima lahko shranjenih veliko Wi-Fi omrežij s katerimi je bila v preteklosti že povezana. Imena Wi-Fi omrežij so pogosto govoreče šifre, kot na primer ime lokala, restavracije, ime in priimek osebe, naziv podjetja … Zato lahko večja omejitev glede števila povpraševanj na odjemalca koristi v fazi zbiranja informacij pri izvedbi napada socialnega inženirstva. Del izvorne kode datoteke airodump-ng.h z omejitvijo v obliki konstante NB_PRB prikazuje Slika 6.

```
#ifndef _AIRODUMP_NG_H_
#define _AIRODUMP_NG_H_

#include "eapol.h"

/* some constants */

#define REFRESH_RATE 100000  /* default delay in us between updates */
#define DEFAULT_HOPFREQ 250  /* default delay in ms between channel hopping */
#define DEFAULT_CWIDTH  20 /* 20 MHz channels by default */

#define NB_PWR  5       /* size of signal power ring buffer */
#define NB_PRB 10       /* size of probed ESSID ring buffer */

#define MAX_CARDS 8 /* maximum number of cards to capture from */

#define STD_OPN     0x0001
#define STD_WEP     0x0002
#define STD_WPA     0x0004
#define STD_WPA2    0x0008

#define STD_FIELD   (STD_OPN | STD_WEP | STD_WPA | STD_WPA2)
```

Slika 6: Del izvorne kode datoteke airodump-ng.h

Dodatna sprememba, ki je bila vgrajena orodju airodump-ng, je izpis podatkov v datoteko v obliki JSON. Pri tem se beleži celotna zgodovina zajema, torej stanje dostopnih točk in odjemalcev v nekem trenutku skozi čas. Uporabljen je bil del izvorne kode, ki se uporablja za izpis v datoteko v obliki CSV. Tako lahko gledamo na JSON datoteko kot na množico CSV datotek porazdeljeno skozi čas pri čemer ena CSV datoteka obsega stanje zajema v nekem trenutku. Del izvorne kode, ki smo jo dodali, prikazuje Slika 7.

```
int dump_write_json( void )
{
    int i, n, probes_written;
    struct tm *ltime;
    struct AP_info *ap_cur;
    struct ST_info *st_cur;
    char * temp;
    printf("DUMP JSON\n");
    if (! G.record_data || !G.output_format_json)
        return 0;

    fseek( G.f_json, 0, SEEK_END );


    ap_cur = G.ap_1st;

    while( ap_cur != NULL )
    {
        if( memcmp( ap_cur->bssid, BROADCAST, 6 ) == 0 )
        {
            ap_cur = ap_cur->next;
            continue;
        }

        if(ap_cur->security != 0 && G.f_encrypt != 0 && ((ap_cur->security & G.f_encrypt) == 0))
        {
            ap_cur = ap_cur->next;
            continue;
        }

        if(is_filtered_essid(ap_cur->essid))
        {
            ap_cur = ap_cur->next;
            continue;
        }

        fprintf( G.f_json, "{\"BSSID\":\"%02X:%02X:%02X:%02X:%02X:%02X\", ",
                 ap_cur->bssid[0], ap_cur->bssid[1],
                 ap_cur->bssid[2], ap_cur->bssid[3],
                 ap_cur->bssid[4], ap_cur->bssid[5] );

        ltime = localtime( &ap_cur->tinit );
        fprintf( G.f_json, "\"FirstTimeSeen\":\"%04d-%02d-%02d %02d:%02d:%02d\", ",
                 1900 + ltime->tm_year, 1 + ltime->tm_mon,
                 ltime->tm_mday, ltime->tm_hour,
                 ltime->tm_min,  ltime->tm_sec );

        ltime = localtime( &ap_cur->tlast );
```

Slika 7: Del dodane izvorne kode datoteki airodump-ng.c

3.1.2. Izbira komponente za prenos dnevniških datotek

Komponenta Wi-Fi Scanner zajema promet v okolici in zapisuje podatke o zajemu v obliki JSON v datoteko. Naloga te komponente je spremljanje stanja datoteke s podatki JSON in pošiljanje teh podatkov do Logstash-a, ki je lahko tudi na oddaljeni lokaciji. Odvisno od razmejitve med ELK skladom in komponento ter stopnjo varnosti, ki jo želimo pri pošiljanju podatkov, lahko izbiramo različna orodja, ki so sposobna dostavljati podatke do Logstash-a.

Varna rešitev pošiljanje podatkov je z orodjem Logstash Forwarder, ki omogoča vzpostavitev varne komunikacije s pomočjo protokola Lumberjack. Pri tem je potrebno pridobiti ali generirati digitalno potrdilo skupaj z zasebnim ključem, ter poskrbeti za distribucijo digitalnega potrdila na vse instance komponente (v primeru, da jih je več). Pri generiranju digitalnega potrdila imamo dve možnosti. Ali vežemo potrdilo na IP naslov od strežnika ali pa vežemo potrdilo na veljavno domensko ime strežnika. V prvem primeru lahko nastanejo težave, ko se spremeni IP naslov Logstash-a, v drugem pa je potrebno poskrbeti za razpoznavanje DNS imena strežnika.

Podatke pa lahko tudi pošiljamo neposredno brez avtentikacije in šifriranja povezave. Obstaja več orodij za posredovanje dnevniških datotek med katerimi lahko izpostavimo Beaver [7], ki temelji na programskem

jeziku Python in orodje NXLog [8], ki je dosegljivo za različne platforme. Obe orodji nudita vse potrebne funkcionalnost za spremljanje dogodkov in posredovanje le teh na strežnik z Logstash-em.

3.1.3. Nastavitve ELK sklada

Namestitev ELK sklada je glede na številčnost virov in navodil na spletu trivialno opravilo. Pomembni del nastavitev je način komunikacije med posrednikom in prejemnikom dnevniških datotek (komponenti Shipper in Logstash). V nadaljevanju navajamo primer konfiguracij orodja Beaver, ki skrbi za posredovanje dnevniških datotek (komponenta Shipper) in prejemnika ter obdelovalca datotek Logstash.

Nastavitvena datoteka za orodje Beaver

```
# /etc/beaver/conf

[beaver]
tcp_host: IP
tcp_port: 5001
format: raw

logstash_version: 1
[./data/out-01.json]
```

Nastavitvena datoteka za Logstash:

```
input {
  tcp {
    host => '127.0.0.1'
    port => '5001'
    type => 'wifi'
  }
}
filter {
 if [type]== "wifi" {
            json {
                        source => message
            }
 }
}
output {
  elasticsearch { host => localhost }
  stdout { codec => rubydebug }
}
```

3.1.4. Rezultati

Pri navajanju rezultatov predpostavljamo, da sta komponenti Wi-Fi Scanner in komponenta za prenos dnevniških datotek nameščeni na napravi Raspberry Pi [9] z operacijskim sistemom Kali Linux in tako fizično ločeni od ELK sklada. Naprava Raspberry Pi ima dva omrežna vmesnika. Prvi omrežni vmesnik je uporabljen za vzpostavitev VPN tunela do ELK sklada, drugi omrežni vmesnik se uporablja za zajem okoliškega Wi-Fi prometa. Pred začetkom izvajanja je bila na napravi Raspberry PI prevedena in zgrajena

modificirana različica orodja Airodump-ng. Zagon orodja airodump-ng, ki zapisuje rezultate zajema kot podatke v obliki JSON, prikazuje spodnji ukaz. Pri tem je predpostavljeno, da je omrežni vmesnik wlan0mon postavljen v stanje, ki omogoča zajem prometa (angl. monitoring mode).

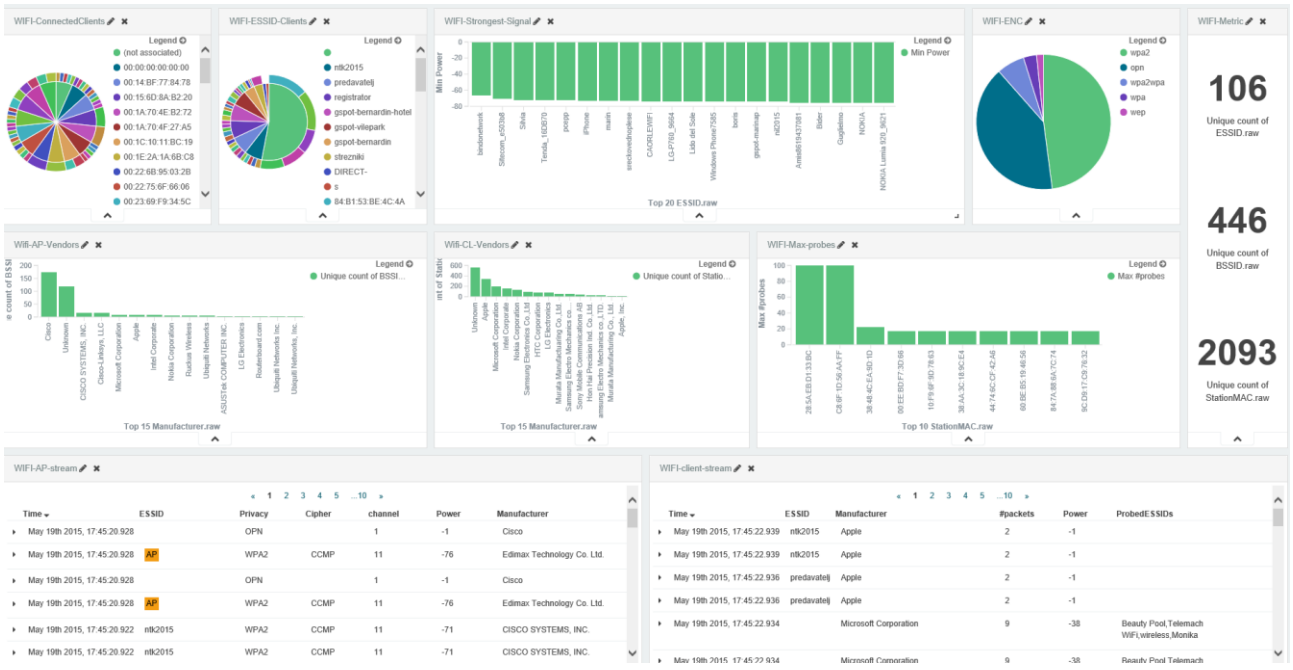
./airodump-ng wlan0mon –output-format json –w ./data/out

Del datoteke s podatki JSON, ki jih zapisuje orodje airodump-ng so prikazani spodaj.

{"BSSID":"BC:67:1C:A6:D0:03", "FirstTimeSeen":"2015-04-24 20:29:39", "LastTimeSeen":"2015-04-24 20:30:40", "channel": 1, "max_speed":" 11","Privacy":"WPA2", "Cipher":"CCMP ", "Authentication":"MGT", "Power":-60, "#beacons": 194,"#IV": 0, "LANIP":" 0. 0. 0. 0", "ID-length": 2, "ESSID":"AS", "wlan_type":"AP","timestamp":"1429900240"}

{"StationMAC":"10:08:B1:09:E2:BF", "FirstTimeSeen":"2015-04-24 20:30:15", "LastTimeSeen":"2015-04-24 20:30:15", "Power":-56, "#packets":1, "BSSID":"(not associated)" ,"ProbedESSIDs":"","#probes":0,"ESSID":"","wlan_type":"CL","timestamp":"1429900240"}

Po zagonu orodja Beaver, Logstash Forwarder ali NXLog se podatki iz datoteke dostavljajo preko VPN tunela na strežnik z Logstash-em. Logstash podatke preoblikuje in posreduje do Elasticsearch-a. Rezultati so vidni v Kibani. Primer rezultatov je prikazan na Slika 8.



Slika 8: Primer pregledne plošče Wi-Fi

S pomočjo Kibane na učinkovit način pridobimo informacije o:

- jakosti signala dostopnih točk skozi čas,
- povpraševanjih (angl. probes) odjemalcev,
- gibanju odjemalcev (se približujejo, oddaljujejo),
- najbolj aktivnih odjemalcih,
- zasedenosti posameznih kanalov v okolici,
- itd…

4. ZAKLJUČEK

ELK sklad je sodobna rešitev za obvladovanje dnevniških datotek iz različnih virov. Zaradi skalabilnosti, prilagodljivosti in odprtosti ELK sklada se ponujajo različne priložnosti pri integraciji na področju informacijske varnostni. V prispevku se je omenjalo možne rešitve prilagajanja ELK sklada za potrebe informacijske varnosti. Bolj podrobno je bil predstavljen način vizualizacije prometa odjemalcev in dostopnih točk v okoliških Wi-Fi omrežjih. Nov pristop k vizualizacijo omogoča prilagodljivost pri predstavitvi pomembnih podatkov in tako pripore k hitrejšemu pridobivanju informacij pri opravljanju analize Wi-Fi prometa in varnosti.

5. LITERATURA

[1] https://www.elastic.co/webinars/introduction-elk-stack, (An Introduction to the ELK stack), obiskano 20.5.2015

[2] https://www.elastic.co/guide/en/elasticsearch/guide/current/getting-started.html, (Elasticsearch: The Definitive Guide » Getting Started), obiskano 20.5.2015

[3] https://www.elastic.co/products/elasticsearch, (Elasticsearch | Search & Analyze Data in Real Time), obiskano 20.5.2015

[4] https://www.elastic.co/guide/en/logstash/current/getting-started-with-logstash.html, (Getting Started with Logstash), obiskano 20.5.2015

[5] http://www.aircrack-ng.org/, (Aircrack-ng), obiskano 20.5.2015

[6] http://hackoftheday.securitytube.net/2015/03/airodump-ng-scan-visualizer-ver-01.html, (Airodump-NG Scan Visualizer ver 1.0), obiskano 20.5.2015

[7] http://beaver.readthedocs.org/en/latest/, (beaver), obiskano 20.5.2015

[8] http://nxlog.org/, (nxlog), obiskano 20.5.2015

[9] https://www.raspberrypi.org/, (Raspberry Pi - Teach, Learn, and Make with Raspberry Pi), obiskano 20.5.2015

[10] https://www.kali.org/, (Kali Linux | Penetration Testing and Ethical Hacking Linux Distribution), obiskano 20.5.2015