Metodologije i alati otvorenog koda za odziv i upravljanje sigurnosnim incidentima

Čoklica, Matija

Master's thesis / Diplomski rad

2018

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Organization and Informatics / Sveučilište u Zagrebu, Fakultet organizacije i informatike**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:211:201038

Rights / Prava: Attribution-NonCommercial-NoDerivs 3.0 Unported/Imenovanje-Nekomercijalno-Bez prerada 3.0

Download date / Datum preuzimanja: **2025-01-05**

SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
VARAŽDIN

Repository / Repozitorij:

Faculty of Organization and Informatics - Digital Repository

zir.nsk.hr

dabar
DIGITALNI AKADEMSKI ARHIVI I REPOZITORIJI

SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
VARAŽDIN

Matija Čoklica

METODOLOGIJE I ALATI OTVORENOG KODA ZA ODZIV I UPRAVLJANJE SIGURNOSNIM INCIDENTIMA

DIPLOMSKI RAD

Varaždin, 2018.

SVEUČILIŠTE U ZAGREBU

FAKULTET ORGANIZACIJE I INFORMATIKE

V A R A Ž D I N

Matija Čoklica

Matični broj: 45270/16–R

Studij: Informacijsko i programsko inženjerstvo

METODOLOGIJE I ALATI OTVORENOG KODA ZA ODZIV I UPRAVLJANJE SIGURNOSNIM INCIDENTIMA

DIPLOMSKI RAD

Mentor :

Doc. dr. sc. Tonimir Kišasondi

Varaždin, rujan 2018.

Matija Čoklica

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

Digitalno doba zahvatilo je gotovo svako poduzeće, neovisno o njihovoj veličini, zrelosti ili industriji. Podaci spremljeni unutar spremišta krajnjeg računala, poslužitelja, digitalnog oblaka, pa čak i mrežnih uređaja poput usmjernika, koji pripadaju poduzeću ili njihovim klijentima moraju imati sačuvan integritet podataka s kojima rukuju, a vrlo često i njihovu tajnost. Ne postoji „garancija“ koja će zaštiti takve podatke te spriječiti nastajanje štete (onu učinjenu namjerno i slučajno), ali zato postoji način na koji se šteta može kvalitetnije odstraniti ili ublažiti, pa čak i pravovremeno spriječiti. Alati, metodologije i politike te stručna skupina ljudi koji zajedno čine odziv i upravljanje sigurnosnim incidentima (eng. „Security incident handling and response“) upravo čine rješenje koje bavi navedenim slučajevima u poduzećima. Ovaj rad sastojati će se od nekoliko komponenata koje čine svojevrsnu hijerarhiju, a ona će biti opisana metodom odozdo (eng. „Bottom-up“). Naime, prva komponenta biti će pojmovi koji se koriste i „gradivne“ su jedinice unutar odziva i upravljanja sigurnosnim incidentima. Sigurnosni događaj, incident, upravljanje incidentom, IRT, SI, TI, SOC, CSIRT, SIEM... samo su neki od pojmova i kratica koji će se objasniti i opisati. Sljedeća komponenta biti će metodologije, okviri, vodiči i kuharice – koristiti će se prijašnje spomenuti pojmovi te će se objasniti njihova orkestracija uz metode, smjernice i recepte poznatih svjetskih organizacija koje se bave informacijskom sigurnosti. Treća komponenta biti će alati otvorenog programskog koda, koji omogućuju sigurnosnim timovima učinkovito upravljenje prijašnje spomenutim komponentama. Spomenuto će biti čak 15 takvih alata, od kojih će tri biti najdetaljnije analizirana i zapakirana u virtualno okruženje. To su TheHive (sa Cortex proširenjem), Koilde Fleet i MozDef.

Ključne riječi: informacijska sigurnost; sigurnosni incident; odziv; upravljanje; datoteka zapisa; metodologija; otvoreni programski kod; sigurnosni alati; forenzika;

Sadržaj

1. Uvod

Tema ovog diplomskog rada je učinkoviti pristup odzivu i upravljanju sigurnosnim incidentima te analiza tržišta alata za odziv incidenata, uz iznimku da su alati dostupni krajnjem korisniku u bilo koje doba za instalaciju, konfiguraciju ali i mijenjanje izvornog programskog koda u skladu sa licencom izvornog autora.

Učinkoviti pristup uključuje korištenje najboljih praksi, odnosno onih smjernica koje su navedene od strane svjetski poznatih i priznatih organizacija koje se unutar svojih djelatnosti bave i informacijskom sigurnošću. Budući da su odziv i upravljanje sigurnosnim incidentima već krajem prošlog stoljeća postala ozbiljna i složena domena sigurnosti, ne samo informacijske, već i računalne i mrežne, napravljeno je nekoliko metodologija, okvira i vodiča te kuharica koje primjenjuju stručni timovi za sigurnost u svojim poduzećima.

Iako stručnjaci na području sigurnosti mogu obavljati odzive i upravljanje sigurnosnim incidentima samo sa odgovarajućim metodologijama i smjernicama, naime, vrlo teško će prepoznati (potencijalni) incident bez pomoći alata za analizu datoteka zapisa ili forenzičkih alata u slučaju već zaraženih datoteka ili ostalih dijelova računala/mreže. Također, stručni tim morati će uložiti dodatni napor oko organizacije i načina podjele poslova/odgovornosti kod odziva i(li) upravljanja sigurnosnim incidentima, ako ne koriste platformu koja im spomenute probleme ne rješava ili barem olakšava. Na kraju, gotovo je nemoguće pronaći ključni podatak za daljnju analizu, bez postojanja središnje točke prikupljanja takvih podataka, odnosno prikupljanja datoteka zapisa. Stoga se u ovom radu spominju alati koji pokrivaju barem jedno područje gore navedenih problema. Njihovom integracijom odziv i upravljanje sigurnosnim incidentima čini dio svojevrsne obrane od malicioznih radnji unutar konteksta informatičke sigurnosti poduzeća.

Ova tema značajna je iz više razloga. Prvo, važnost ove teme u poslovnom svijetu raste sa porastom „Cyber“ kriminala. Poslovni podaci, bili oni u vlasništvu poduzeća u kojem su stacionirani ili partnera s kojim to poduzeće posluje, mogu imati određenu vrijednost za napadača (u svijetu „Cyber“ kriminala takva osoba se naziva „Black Hat“ hakerom). Napadač će iskoristiti ranjivosti kako bi došao do podataka. Ranjivosti se nalaze na osobnim računalima, poslužiteljima, mrežnoj infrastrukturi, algoritmima zaštite, pa čak i na samim zaposlenicima. Svijest o sigurnosti i mogućem načinu iskorištavanja može se podići kod ljudi preko raznih programa za podizanje svijesti, no takvo nešto ne funkcionira kod ostatka IT infrastrukture. Ujedno je ovo i drugi razlog zašto je tema ovog rada značajna – što je sve potrebno te na koji način obuhvatiti i zaštititi poduzeće od štete, kako bi se ona što više smanjila ili potpuno izbjegla. Ipak, na prvi pogled i samim sigurnosnim stručnjacima, implementacija zaštite može djelovati poput „znanstvene fantastike“, tj. analiza svake točke u mreži poduzeća, pripravne politike i procedure za konkretno poduzeće u slučaju incidenta te pronalazak i odabir alata koji bi omogućio integraciju, barem onu djelomičnu. Potrebno je dobro poznavanje poslovnih procesa, mrežne i računalne infrastrukture, ali i korake upravljanja i rješavanja pojedinih incidenata: krađa podataka, neovlašteni upad i mijenjanje podataka, zaraza malicioznim programskim kodom (za svaku vrstu koda posebno), uskraćivanje usluge... Ovo čini i treći razlog važnosti teme kao takve, jer kao što u naslovu piše „Metodologije i alati...“, posebna se pažnja pridodaje baš

ovima dvjema čimbenicima kod odziva i upravljanja sigurnosnim incidentima. Odabir krive metodologije ili smjernica može dovesti do neučinkovito napisanih politika i procedura upravljanja i odziva na razne događaje i(li) incidente, dok odabir krivog alata može drastično smanjiti učinkovitost stručnog tima. Iako se na tržištu najčešće uzimaju komercijalna rješenja, jer su robusnija i bolje podržana (najčešće se pruža i stalna servisna usluga), može se pronaći i besplatnih alata otvorenog programskog koda koji pružaju učinkovitu potporu (najčešće) odzivu na sigurnosne incidente. Mala napomena: budući da postoji velik broj alata koje pripada ili nudi rješenja za upravljanje i/ili odziv sigurnosnim incidentima, oni alati koji nisu održavani (ni jedna značajna promjena u protekle dvije godine) te oni alati koji su preopširni za opisati, neće biti navedeni unutar ovog diplomskog rada.

Autor ovog rada odabrao je baš ovu temu jer se trenutno bavi spomenutom domenom te je i sam sudjelovao u korištenju, mijenjanju te razvijanju alata koji potpomažu sigurnosnim stručnim timovima obaviti svoj posao što bolje mogu.

2. Metode i tehnike rada

Sve što je korišteno u diplomskom radu, usko je vezano uz autorovo istraživanje provedeno na stručnoj praksi te daljnji rad u poduzeću u kojemu je autor stručnu praksu obavio. Istraživačke aktivnosti provedene su na način pronalaska alata na platformi za razvoj softvera GitHub. Nakon toga, sljedeći korak je bio pronaći metodologije i vodiče vezane uz odziv i upravljanje sigurnosnim incidentima. Uz kratku analizu prikupljenih materijala, odabrane su ključne riječi koji su kritične za shvaćanje teme rada. One su posebno analiziranje u specijaliziranim enciklopedijama poput one na stranicama Techtarget. Od alata koju su testirani, unutar vlastitog virtualnog okruženja ili testnog okruženja u oblaku, to su:

- Forenzički orijentirane platforme: Kolide Fleet + Osquery, Doorman + Osquery, GRR Rapid Response
- Platforme sa ugrađenim SIEM-om: Zentral, Cyphon, MozDef
- Orkestracijske platforme za SOC (eng. „Security Operations Center“): TheHive + Cortex, SCOT, RT for Incident Response, Fast Incident Response, CrowdStrike Falcon Orchestrator
- Pomoćni alati za upravljanje i odziv sigurnosnim incidentima: threat_note, nightHawk Response, CIRTKit, MISP Threat Sharing

Odabrani alati za prezentaciju na obrani su Kolide Fleet + Osquery, MozDef i TheHive + Cortex.

3. Mala abeceda odziva i upravljanja sigurnosnim incidentima

Pojmovi koji se javljaju unutar područja odziva i upravljanja sigurnosnim incidentima (u nastavku skraćeno IH&R) mnogobrojni su i često slični jedni drugima. IH&R čini skup politika i vodiča, koji opisuju radnje ili korake rješavanja incidenta kao „temeljne jedinice“ IH&R. Incidenti nastaju iz događaja, a događaji se najčešće skupljaju unutar IDS/IPS sustava i SIEM platforme. CIRT i SOC timovi obavljaju IH&R, gdje SOC nadgleda SIEM preko različitih vizualizacija i generiranih alarma, dok CSIRT vrši odziv. Također, radnje koje CSIRT obavlja, vezane su uz vodiče koji opisuju postupak rješavanja specifične vrste incidenta, no vrlo često zahtijevaju mrežnu ili računalnu forenziku. Unutar ove cjeline, definirati će se upravo spomenuti pojmovi, razlike između sličnih pojmova, zatim će se objasniti zadaće stručnih timova, a na kraju ukratko objasniti SIEM te izvori informacija bez kojih IH&R ne bi bio mogući.

3.1. Sigurnosni događaj i sigurnosni incident

Sigurnosni događaj ili skraćeno događaj (eng. „event“) je bilo koja primijećena pojava u sustavu ili mreži [1][2][3]. Primjeri događaja su povezivanje korisnika sa uslugom za dijeljenje datoteka, poslužitelj koji dobiva zahtjev za web stranicom, vatroštit (eng. „firewall“) koji zaustavlja pokušaj povezivanja, itd. Događaj može ukazati na događanje ili već nastupili incident [1]. Primjeri potencijalnog incidenta su zapisi unutar „system boot“ datoteke zapisa, rušenje sustava, degradacija mrežnih performansi, 500 novih e-mailova unutar 5 minuta, promijenjena zaštitna suma izvršne datoteke [2]. Sigurnosni incident ili skraćeno incident, je prijetnja ili povreda sigurnosnih politika, pravila prihvatljivog korištenja ili standardne sigurnosne prakse [3]. Primjeri incidenata su [1]:

- Napadač upravlja mrežom zaraženih čvorova kako bi poslao veliku količinu podataka, dovoljnu da sruši poslužitelj.
- Korisnici su prevareni otvaranjem e-mail privitka zaraženog malicioznim sadržajem koji inficira korisnikovo računalo i komunicira sa napadačevim poslužiteljem.
- Napadač izvrši povredu podataka tako da „ukrade“ podatke i za njih traži određenu svotu novaca.
- Korisnik slučajno otkrije tajne informacije kroz uslugu podjele datoteka.

Incident se također može objasniti kao svaki događaj koji krši elemente tajnosti, integriteta i dostupnosti informacijskog sustava i njegovih podataka [2][4]. Potencijalni incidenti (ali ne i nužno incidenti) mogu biti [2]:

- napadi malicioznim programskim kodom
- skeniranje portova i mreže

- neovlašteni pristup
- neovlašteno korištenje usluga
- onemogućivanje korištenja usluga (eng. „DOS“)
- zloupotreba
- špijunaža
- podvala
- virus
- upad i povreda podataka

Je li događaj incident, to će odrediti stručna osoba, no općenito se događaji razlikuju od incidenata po razini ozbiljnosti i količini rizika za zahvaćenu organizaciju [5]. Upravljanje incidentima (eng. „Incident Management“) je proces razvoja i održavanja sposobnosti upravljanja incidentima unutar organizacije, tako da se incident što prije sanira te se oporavak postigne unutar određenog vremena [2]. Također, upravljanje incidentima je pojam koji opisuje aktivnosti prepoznavanja, analize i ispravljanja opasnosti kako bi se spriječila ponovna pojava istog ili sličnog incidenta [6]. Upravljanje računalnim sigurnosnim incidentima (eng. „Computer Security Incident Management“) je poseban oblik upravljanja incidentima, čija je primarna svrha razvoj razumljivog i predvidljivog odgovora na štetne događaje [7]. Ova vrsta upravljanja incidentom uključuje praćenje i otkrivanje sigurnosnih događaja na računalu i/ili njegovoj mreži te izvršavanje odgovarajućih akcija ili odgovora na te događaje [2][7]. Česta praksa analitičara koji se bave upravljanjem incidenata je pisanje bilješki. Neke literature navode kako upravljanje incidentima počinje vlastitim zapisima [8], koji dobro dođu kod formalne dokumentacije, ali i kod uobičajenih promatranja i razmjene znanja sa ostalima koji sudjeluju u upravljanju i/ili odzivu na isti incident.

Jedan od alata spomenutih unutar ovog diplomskog rada, unutar cjeline „Pomoćni alati za upravljanje i odziv sigurnosnim incidentima“, daje primjer kako pisanje bilješki može biti strukturirano te tako povećati učinkovitost upravljanja incidentom, posebice na području evidentiranja (potencijalnih) indikatora kompromisa.

3.2. Upravljanje incidentom, odziv te plan odziva na incidente

Sigurnosni stručnjaci podijeljeni su po pitanju definiranja komponenata upravljanja incidentom. Neke literature spominju kako upravljanje incidentima uključuje odziv na incidente (eng. „Incident Response“) [2], dok opet druge literature navode postojanje posebnog pojma upravljanja incidentom (eng. „Incident Handling“), koji je dio „Incident Managementa“. Ipak, najčešće se radi „u paru“, gdje jedna osoba analizira incident i komunicira sa potrebnim ljudima, a druga vrši forenziku i oporavak nakon incidenta. Prva osoba naziva se „Incident Handler“

i zadužena je za logistiku, komunikacije, koordinaciju i planiranje potrebno za rješavanje incidenta na miran i učinkovit način [9]. Od takve osobe očekuju se kvalitetne komunikacijske i upraviteljske sposobnosti. Druga osoba, „Incident Responder“, bavi se odzivom na incident, a to je organizirani pristup rješavanju posljedica povrede sigurnosti, „Cyber“ napada ili općenito – sigurnosnog incidenta [10]. Cilj odziva na incident je kvalitetno analizirati (za tehničke strane) i zatvoriti/suzbiti incident te tim putem smanjiti daljnju štetu i povećati šanse/smanjiti vrijeme za oporavak. Takva osoba mora biti stručnjak za računalne mreže, analitičar datoteka zapisa i računalni/mrežni forenzičar [9].

Općenito, ciljevi aktivnosti upravljanja i odziva na incident mogu se sažeti u sljedeće korake [2]:

- pronađi incidente brže nego prije
- dijagnosticiraj preciznije
- upravljaj incidentima propisno
- suzbij i minimiziraj štetu
- vrati usluge pogođene incidentom na stanje prije incidenta
- pronađi uzrok nastanka incidenta
- implementiraj poboljšanja kako se incident ne bi ponovno dogodio

Kako bi cijeli „Incident Management“ bio učinkovit, potrebno je sagraditi inicijalni plan odziva na incidente (eng. „Incident Response Plan“, skraćeno IRP), koji bi trebao sadržavati korake detekcije, načina odziva i ograničavanja negativnog učinka incidenta na organizaciju. Plan odziva na incidente mora biti jedinstven jer je u relaciji sa misijom, ciljevima, veličinom, strukturom i funkcijama organizacije [1]. Postoji devet koraka koji čine kvalitetni plan odziva na incidente [11]:

- Analizirati poslovnu okolinu, ali i IT infrastrukturu kojom je potpomognuta.
- Pronaći kritične točke u poslovanju i saznati uloge/prava zaposlenika koji su u doticaju sa IT infrastrukturom.
- Stvoriti učinkoviti tim stručnjaka koji će pratiti sigurnosne događaje te upravljati i rješavati sigurnosne incidente.
- Razviti potrebne KPI (eng. „Key Performance Indicators“) koji će biti prikazani skupom razine ozbiljnosti, značaja za poslovanje te potrebnog napora za otklanjanje i SLA (eng. „Service-Level Agreement“).
- Izraditi vodiče za rješavanje raznih neželjenih (generičkih) scenarija koji se mogu dogoditi tokom napada na organizaciju. Vodiči moraju imati razvijen način ili korake otkrivanja, rješavanja i oporavka od incidenta.

- Testirati vodiče i po potrebi prilagoditi organizaciji.
- Osmisliti i integrirati rješenja u slučaju poslovne štete nastale tokom incidenta (virtualna okruženja, rezervne pohrane najvažnijih podataka, brzi kontakti za oporavak. . .).
- Uključiti i ostale odjele u razvoj plana, ne samo SOC i/ili CSIRT.
- Uspostaviti proces procjene incidenta te primjene naučenog. Po potrebi ponovo prilagoditi ovaj plan, tako da se u slučaju ponovnog incidenta on može učinkovitije riješiti, pa čak i izbjeći.

3.3. Što je svijest o informatičkoj sigurnosti i zašto je uopće bitna

Sukladno s prvim korakom, određeni ljudi sa određenim odgovornostima, a samim time i pravima (što najviše zanima napadače), postaju meta, a vrlo često i žrtve. Zato je potrebno održavati edukaciju o svijesti za informatičku sigurnost (eng. „Security Awareness“) unutar organizacije. To je edukacija zaposlenika o sigurnosnim politikama unutar organizacije, zatim ispravno korištenje IT usluga potrebnih za njihovo poslovanje te načini i vektori napada preko kojih zaposlenici mogu postati žrtve i sudionici u mogućem incidentu. Svaka organizacija ima vlastiti stil treninga i edukacije koji je kompatibilan sa njezinom organizacijskom kulturom [12]. To mogu biti klasične edukacije u učionicama, zatim „online“ edukacija i radionice, vizualizacije i posteri o sigurnosti postavljeni na predviđena mjesta, ali i „phishing“ kampanje – jer čovjeku se napadom najbolje dokaže koliko je slab/jak. Naravno, takve „phishing“ kampanje moraju biti odobrene od nadležnih tijela organizacije te ne smiju napraviti pravu štetu u poslovanju. Neke literature čak navode kako bi zaposlenici jedni druge morali preispitivati, pa čak i ponekad pomoći sigurnosnim timovima u njihovim aktivnostima [2], kako bi se bolje upoznali s prijetnjama. Predmete koje bi ovakva edukacija trebala pokrivati su [12]:

- „Phishing“ napadi: zaposlenici trebaju znati prepoznati klasične „spam“ poruke, ali i specifične tzv. „Spear Phishing“ napade.
- Fizička zaštita: potrebno je naučiti zaposlenike koliko je važno zaključavati svoje ladice, kako postupati sa gostima u uredu, zašto je opasno na zaslonu računala imati papirić sa korisničkim podacima, itd.
- Računalna zaštita: naučiti zaposlenike koje su moguće posljedice nezaključavanja računala te uparivanja nepoznatog USB uređaja sa računalom na kojem je aktivan povlašteni korisnik (npt. domenski korisnik).
- Opasnosti na mreži: Naučiti koje opasnosti vrebaju u npr. otvorenoj bežičnoj mreži i zašto nije najpametnija ideja spojiti se na takvu mrežu.
- Opasnosti od korištenja loših lozinka: Potrebno je objasniti zaposlenicima zašto se toliko forsiraju kompleksne lozinke, zatim mijenjanje lozinke nakon određenog vremena, itd. Dobra praksa je upoznati zaposlenike sa alatima za upravljanje lozinkama.

- Osnove o malicioznom kodu: Potrebno je naučiti ljude kakve sve vrste malicioznog programskog koda postoje te jasni čimbenici njihova postojanja, bez da se zaposlenike uči tehnikama forenzike.

3.4. Uloge sigurnosnih stručnjaka u organizaciji

Spomenute aktivnosti, poput upravljanja i odziva na sigurnosne incidente, implementacije plana za odziv na incidente i edukacija zaposlenika, zadaća je sigurnosnog tima ili timova koji postoje u organizaciji kao zasebni odjel ili kao treća strana („outsourcing“ usluge informatičke sigurnosti). Takav odjel naziva se operacijski centar za sigurnost (eng. „Security Operations Center“, skraćeno SOC). Sastoji se od tima ljudi koji su stručnjaci informacijske sigurnosti te su primarno zaduženi za kontinuirano praćenje događaja i analizu položaja sigurnosti organizacije [13]. Cilj ovog tima je pronaći, analizirati i odgovoriti na sigurnosne incidente koristeći kombinaciju tehnologije i ljudskog iskustva. Idealno, aktivnosti odziva na sigurnosne incidente obavlja tim za odzive na računalni sigurnosni incident (eng. „Computer Security Incident Response Team“, skraćeno CSIRT) [5]. CSIRT je tim koji može postojati u organizaciji kao posebni odjel ili se stvara „ad-hoc“ prema potrebi sa zaposlenicima iz SOC odjela. Također, postoji stalni CSIRT na razini države, koji se bavi odzivima incidenata na razini cijele države. Razne maliciozne kampanje koje ciljaju ministarstvo, više poduzeća, javni sektor, itd. posao su za državni CSIRT. Općenito, CSIRT dobiva izvještaje od sigurnosnim povredama i prodorima, zatim vrši analize nad izvještajima, koristeći tehnologije i znanja koja uključuju forenziku, reverzno inženjerstvo, itd., a na kraju analize, svoje odgovore šalju dalje na obradu [14].

3.5. „Intelligence“

Pierluigi Paganini, jedan od svijetski poznatih CISO-a, navodi kako se tipična SOC infrastruktura sastoji od vatroštita, sustava pronalaska i zaustavljanja upada, ispitnih agenata te sustava za upravljanje sigurnosnim informacijama i događajima [13]. Tehnologija koja se koristi, mora prikupljati informacije putem tokova podataka, telemetrije, snimanja (mrežnih) paketa, datoteka zapisa i ostalih prihvaćenih metoda kako bi se svaka aktivnost, potrebna za analizu, mogla učinkovito pregledati i korelirati, a po potrebi i dalje analizirati. Svaka informacija koja je važna za zaštitu organizacije od unutrašnjih i vanjskih prijetnji, zatim prijašnje spomenute tehnologije i politike poput plana za odziv na incidente, zajednički se nazivaju „Security Intelligence“ [15] (hrvatski prijevod ovog pojma bio bi dosta „nespretan“). Oba tima, SOC i CSIRT, rade s „Security Intelligence“ - najčešće razlike su u tome što alati CISRT-a moraju biti više forenzički usmjereni, dok će alati za SOC biti analitički i vizualizacijski alati. Također, CSIRT će raditi samo nad incidentom, dok će SOC obraditi svaki sumnjivi događaj, odnosno potencijalni incident i incident nakon CSIRT-ove analize i odgovora.

Jedan dio „Security Intelligence“-a koji nije spomenut, a može igrati veliku ulogu u ranom otkrivanju ili zaštiti od pojave incidenta, je „Threat Intelligence“ – skup analiziranih, pročišćenih i organiziranih informacija o potencijalnim ili trenutnim rizicima i napadima koji prijete

organizaciji [16]. Neke literature navode važnost elementa „Threat Intelligence“-a u svakodnevnim sigurnosnim, ali i operativnim IT funkcijama i poslovima, do razine sastavljanja posebnog tima ljudi unutar organizacije koji se bave isključivo „Threat Intelligence“-om [8]. Glavna namjena „Threat Intelligence“-a je pomoći organizacijama u razumijevanju i savladavanju rizika svih vanjskih prijetnji, kao što su „zero-day“ ranjivosti, razne napadačke grupe (popularno nazvane „Advanced Persistent Threats“ ili skraćeno APT) te maliciozne kampanje. Jedan od načina korištenja „Threat Intelligence“-a kojeg organizacije često prihvaćaju je putem „Threat Intelligence Feed“-a, skraćeno TIF. To je kontinuiran tok podataka koji su povezani sa potencijalnim ili trenutnim prijetnjama usmjerenim prema organizacijskoj sigurnosti [17]. Taj tok podataka prikazan je kroz određeni format datoteke, a sam tok može biti generiran za unutrašnje upotrebe te je dostupan samo za jednu ili ograničen broj organizacija. Druga vrsta takvog toka podataka generirana je za javne upotrebe (gdje se organizacije mogu pretplatiti na TIF), izložena besplatno ili sa određenim modelom plaćanja svim klijentskim organizacijama. U praksi, relevantniji i precizniji podaci su najčešće komercijalizirani TIF, no potrebno je uzeti u obzir kako ne postoje sustavi detekcije sa 100-postotnim učinkom nalaženja samo „True Positive“ indikatora. Najčešće vrste podataka koje se mogu podijeliti preko TIF-a su maliciozne IP adrese, domene, potpisi („Hashevi“) i user-agenti. Naravno, ove podatke SOC ili CSIRT timovi ne bi smjeli ručno analizirati, jer bilo bi veoma neučinkovito i mukotrpno uzimati sumnjive podatke iz npr. SIEM platforme, a zatim ih uspoređivati sa svakim TIF-om kojeg organizacija koristi. Kako bi se automatizirao i uskladio rad platforme za praćenje i uspoređivanje podataka događaja s podacima iz TIF-ova, postoji „Threat Intelligence Platform“. Takva platforma mora imati sljedeće funkcionalnosti [18]:

- Agregacija „Threat Intelligence“-a s više izvora, uz mogućnosti ponude odgovarajućeg TIF-a.
- Normiranje, obogaćivanje i automatska procjena rizika pristiglih podataka.
- Integracija s postojećim sigurnosnim rješenjima, kako bi se postigla veća analitička učinkovitost.
- Analiza i dijeljenje vlastitog „Threat Intelligence“-a.

Unutar cjeline „Pomoćni alati za upravljanje i odziv sigurnosnim incidentima“ nalazi se kratak opis jedne od „Threat Intelligence Platform“-i otvorenog programskog koda koja je popularna među sigurnosnom zajednicom.

3.6. Središnja točka praćenja te početna točka pronalaska i analize incidenta

Uz toliko alata koji čine sigurnosnu komponentu IT infrastrukture organizacije, postavlja se pitanje, kako je uopće moguće pratiti sigurnosne događaja u stvarnom vremenu, ali i kako odabrati te prepoznati događaj kao incident. Potrebna je jedna središnja točka koja će primati događaje sa raznih izvora te pravovremeno reagirati prema pravilima koja su navedena unutar

plana za odziv incidentima ili eksplicitno od strane SOC tima. Prije spomenuti sustavi pronalaska i zaustavljanja upada (eng. „Intrusion Detection and Prevention Systems“, skraćeno IDPS) mogu poslužiti ovoj svrsi. Ova vrsta alata uzima na analizu događaje sa raznih izvora unutar IT infrastrukture te ih prema pravilima detektira kao sumnjive, ali i odgovara na maliciozne događaje prema namještenim pravilima i dozvoljenim akcijama [19]. Uz pomoć alarma, IDPS može opskrbiti CSIRT tim indikatorima potencijalno malicioznih aktivnosti kako bi odziv na incident bio kvalitetnije odrađen [2]. Nedostatak IDPS-a je činjenica da su spomenute funkcionalnosti ograničene na računalne (eng. „Host“) i mrežne aktivnosti [20], gdje je jedan IDPS specijaliziran samo za jednu od tih aktivnost (gdje se opet dolazi do problema pronalaska jedne središnje točke praćenja događaja).

Sustav za upravljanje sigurnosnim informacijama i događajima (eng. „Security Information and Event Management“, skraćeno SIEM) nema spomenuti nedostatak. Ovaj alat je upravo i napravljen u tu svrhu – agregacija važnih događaja sa više izvora (bez obzira na njegovu aktivnost i domenu), identifikacija anomalija te poduzimanje određenih mjera poput slanja alarma SOC ili CSIRT odjelu [21]. Za razliku od IDPS, sigurnosni timovi kod implementacije SIEM alata imaju veću slobodu upravljanja datotekama zapisa [2], poput njihova obogaćivanja dodatnim podacima ili ignoriranju na razini ulaska podataka u sustav. Zatim tu je i podrška za „Threat Intelligence Platform“-e i razne istraživačke agente. Također SIEM nudi razinu korelacije događaja koje niti jedan dio „Security Intelligence“-a ne može ponuditi [22]. Na tržištu postoje i razni SIEM alati s funkcionalnostima poput automatske forenzike, pa čak i strojnog učenja [21]. No, sa tako moćnim oružjem, dolaze velike odgovornosti – kupnja SIEM-a te njegovo „bacanje“ u pogon po samoj kupnji napraviti će više štete nego koristi [22]. Također, kao jedna o kompleksnijih tehnologija, SIEM može stvarati poteškoće manjim organizacijama koji nemaju dovoljno resursa poput analitičara i stručnjaka za upravljanje SIEM-om [20], a tu su i strojna ograničenja – ovisno o frekvenciji pristiglih događaja i pamćenju onih starijih, potrebno je izdvojiti infrastrukture s diskovnim prostorima od nekoliko desetaka terabajta!

SOC, a povremeno i CSIRT, vrlo često ima potrebu za alatom koji će biti agregator svih sigurnosnih događaja, spremište za brzu pretragu i razne vizualizacije uz koje će analitičari moći reagirati na sumnjive događaje i razne anomalije te na vrijeme izvijestiti potrebite o incidentu. Ako se implementacija SIEM-a ne isplati zbog raznih značajki organizacije (poput veličine), potrebno je pronaći alternativu. Iako je SIEM kompleksan i tema je koja prekoračuje opseg ovog diplomskog rada, pronađeno je i opisano nekoliko alata otvorenog programskog koda koji dolaze sa SIEM-om kao jednom od svojih dodatnih komponenata.

3.7. Nema odziva bez forenzike

Budući da se prošlo poglavlje više odnosilo na SOC nego na CSIRT, jer SIEM je alat iz kojeg će analitičar „izvaditi“ događaje, dodatno ih analizirati uz pomoć svojeg znanja, iskustva i drugih pomoćnih alata te na kraju odlučiti hoće li ih proslijediti CSIRT-u ili ne, potrebno je sada spomenuti „glavnu radnju“ stručnjaka iz CSIRT odjela – forenziku. Općenito, forenzika je pojam

korištenja znanosti i tehnologije u potraži za dokazima i utvrđivanju činjenica u kaznenim ili građanskim pravnim sudovima [23]. Unutar domene IT-a, forenzika je podijeljena na mrežnu, računalnu, animacijsku te autorsku. CSIRT se bavi mrežnom i računalnom forenzikom. Mrežna forenzika predstavlja snimanje i analizu mrežnih događaja kako bi se otkrio izvor sigurnosnih napada ili općenito incidenta [24]. Raširene su dvije vrste mrežne forenzike, jedna koristi više diskovnog prostora dok druga koristi više procesorske snage. Ono što se često gleda i korelira između mrežnih paketa su sumnjive IP adrese, domene, user-agenti i sadržaji paketa. Ako je potrebno i ako CSIRT tim ima za to osposobljen tim, moguće je odraditi lov na napadača (eng. „Threat Hunting“) te pokušati onesposobiti njegovu infrastrukturu poput C2 poslužitelja. Računalna forenzika obuhvaća tehnike istraživanja i analiza za prikupljanje i očuvanje dokaza iz napadnutog računalnog uređaja na način prikladan za prezentaciju. Cilj je pronaći lanac događaja koji su postepeno radili štetu na uređaju te rekonstruirati napad (u praksi se ovo često naziva stvaranje PoC-a) i pokušati saznati tko stoji iza njega, koje su ranjivosti iskorištene, koji maliciozni programski kod je korišten, itd. [25]

Neovisno o vrsti, od forenzičkog softvera se očekuju funkcionalnosti poput veoma snažnih pretraga po uzorku, poput pretrage cijelog diskovnog prostora samo kako bi našli sve zapisane e-mail adrese [8]. Budući da postoji velik broj istraživačkih radova i unapređenja u području forenzike, pa i sve sofisticiranijih napada i malicioznih programskih kodova (eng. „Malware“), stručnjaci iz CSIRT odjela moraju kontinuirano evaluirati svoje alate te tehnike i procedure, kako bi znali jesu li i dalje pogodni za izvršavanje forenzike [3]. Tokom praktičnog dijela ovog diplomskog rada, biti će opisani razni alati otvorenog programskog koda koji pomažu CSIRT-u kod izvršavanja forenzike.

4. VERIS - okvir za sigurnosne incidente

Rječnik za evidenciju događaja i dijeljenje incidenata (eng. „Vocabulary for Event Recording and Incident Sharing“, skraćeno VERIS) okvir je koji predstavlja standardizirani oblik prikupljanja, klasifikacije, analize i dijeljenja informacija o sigurnosnim incidentima na strukturiran, ponovljiv, anoniman i siguran način [26]. Ovaj okvir je kao projekt otvorenog koda pokrenula organizacija Verizon 2010. godine, a projekt nastavlja zajednica ljudi koji stvaraju nove ili ažuriraju postojeće kategorije, tzv. enumeracije incidenata. VERIS se može protumačiti kao skup metrika koje pružaju zajednički jezik za opisivanje sigurnosnih incidenata ili prijetnji [27], kojeg bi svaki analitičar mogao jednostavno interpretirati unutar svoje organizacije, povećavajući učinkovitost sigurnosnih timova te „security intelligence“-a, ali i pridonijeti kvalitetnijem upravljanju rizika. Iz pozicije sigurnosnog stručnjaka koji opisuje sigurnosni incident, VERIS okvir nudi definiranje općih informacija (eng. „Incident Tracking“), demografije žrtava (eng. „Victim Demographics“), informacija o pronalasku i odzivu (eng. „Discovery and Response“), procjene štete (eng. „Impact Assessment“), indikatora kompromisa (eng. „Indicators“) te informacija prema modelu prijetnje nazvanom A4 (eng. „Actors-Actions-Assets-Attributes“). Gleda li se VERIS s tehničkog aspekta, ne radi se o nikakvom obliku programskog koda poput biblioteke, proširenja ili gotovog softvera. VERIS se sastoji od JSON datoteke koja predstavlja shemu (postoje određene razine hijerarhije), skripti za uvoz/izvor sheme u nekoliko digitalnih formata te mapiranje informacija u/iz VERIS okvira, zatim dokumentacije koja predstavlja vodič za implementaciju okvira, edukaciju i gotove primjere, a tu je i baza podataka o javno prikupljenim incidentima (većinom oko 2012. i 2013. godine) pisanim u JSON formatu prateći VERIS shemu [28].

Budući da VERIS opisuje sigurnosne incidente, uz hijerarhijski koncept, potrebno je objasniti koji su minimalni uvjeti te općeniti elementi opisa JSON datoteke pisane uz pomoć VERIS okvira (u nastavku, takve će se datoteke nazivati VERIS formatom). Koristi se „Top-down“ pristup – elementi, koji su prikazani tekstom bez razmaka, razdvojeni su točkom gdje element koji se nalazi s lijeve strane točke pripada višoj razini hijerarhije nego element sa desne strane točke. U ovom poglavlju biti će navedeni samo oni elementi prve (najveće) razine hijerarhije. Minimalni uvjeti za opis incidenta su elementi koji određuju je li događaj stvarno incident (eng. „security_incident“) te kako je događaj otkriven (eng. „discovery_method“) [28]. Poželjno je definirati vrijeme nastanka incidenta (eng. „timeline“) i sažetak koji u jednoj rečenici opisuje incident (eng. „summary“). Svaki adekvatni opis incidenta mora imati A4 model prijetnji, tj. elemente tko je kriv za incident (eng. „actor“), koje akcije je krivac poduzeo (eng. „action“), koja/kakva je imovina zahvaćena incidentom (eng. „asset“) te kako/što je zahvaćeno/oštećeno kod te imovine (eng. „attribute“). Često se smiju (javno) definirati i podaci o žrtvi (eng. „victim“) i šteti koja je počinjena nastalim incidentom (eng. „impact“). Također, tu su i metapodaci poput jedinstvenog identifikacijskog niza znakova incidenta (eng. „incident,_id“) te inačice VERIS sheme koja se koristi u trenutnoj datoteci za opis incidenta (eng. „schema_version“). Želi li korisnik VERIS okvira dodati vlastite elemente, može ih smjestiti pod „plus“ element koji također pripada najvećoj razini hijerarhije.

4.1. Objašnjene VERIS sheme

VERIS dokumentacija olakšava shvaćanje i korištenje sheme te na strukturiran način opisuje hijerarhiju unutar okvira. Za svaku varijablu VERIS okvira (varijable predstavljaju „listove“ u hijerarhijskom stablu) definirani su:

- „Question text“: na koji problem ili pitanje odgovara varijabla.
- „User notes“: napomene vezane uz varijablu, omogućuje bolje razumijevanje varijable ili elementa razine iznad.
- „Question type“: opisuje tip podataka varijable (običan tekst, broj, enumeracija, itd.)
- „Variable name“: puno ime varijable.
- „Enumerations“: iz koje enumeracije dolazi varijabla (kod nekih varijabli ovo je izostavljeno).
- „Purpose“: objašnjenje zašto je potrebno imati ovu varijablu unutar VERIS-a.
- „Developer notes“: napomene vezane isključivo za razvojne inženjere VERIS okvira.
- „Miscellaneous“: razna upozorenja, ograničenja i zanimljivosti za korisnike VERIS okvira.

Potrebno je napomenuti kako VERIS dokumentacija sadrži opis varijabli kroz cjeline, ali i kroz enumeracijsku listu (eng. „schema enumerations“), no može se primijetiti kako naziv (putanja) varijable nije uvijek identičan unutar oba dijela dokumentacije. Također potrebno je spomenuti kako VERIS dokumentacija navodi slobodu korištenja VERIS okvira, tj. VERIS ne tjera organizaciju ka potpunoj implementaciji okvira. Ponekad postoje razlozi gdje nije preporučeno koristiti VERIS format, jer potrebno se prisjetiti da jedan od razlog postojanja VERIS-a je potreba za normiranim dijeljenjem incidenata sa širom zajednicom. Želi li se koristiti VERIS okvir „as-is“, uz funkcionalnosti dijeljenja, preporučeno je detalje incidenta koji moraju ostati tajni spremiti unutar „plus“ elementa, a zatim sve „plus“ elemente (sa njihovim varijablama) izostaviti kod dijeljenja s javnošću.

4.2. Opće informacije o incidentu

Opće informacije o incidentu postoje kako bi se normirao postupak identifikacije, pohrane i dohvaćanja incidenata kroz vrijeme. Postoji sedam varijabli koje pripadaju ovoj cjelini:

- „incident_id“: varijabla za jedinstvenu identifikaciju incidenta za potrebe pohrane i praćenje kroz vrijeme.
- „source_id“: varijabla koja predstavlja poveznicu sa entitetom koji upravlja incidentom ili pruža usluge odziva.

- „security_incident“: odgovara na pitanje „je li stvarno ovo incident?“. Moguće vrijednosti su „Confirmed“, „Suspected“, „False positive“ i „Near miss“, gdje je ova zadnja vrijednost varijable interpretirana kao „evidentirane akcije nisu oštetile imovinu“.
- „summary“: opis incidenta unutar jedne rečenice.
- „related_incidents“: ako je incident povezan s drugim incidentom, potrebno je napisati „incident_id“ tog incidenta ili opis relacije.
- „confidence“: kolika je točnost/preciznost informacije o danom incidentu. Vrijednosti mogu biti „High“, „Medium“, „Low“ ili „None“.
- „notes“: ostale važne napomene vezane uz navedene varijable ove cjeline.

4.3. Demografije žrtava

Demografije žrtava cjelina su koja opisuje (ali ne identificira) organizaciju zahvaćenu incidentom. Postoji osam varijabli koje pripadaju ovoj cjelini:

- „victim.victim_id“: identifikator organizacije koju je zahvatio incident. Vrlo često ova varijabla neće biti popunjena (tajni podatak ili nema potrebe za dijeljenjem incidenata sa širom zajednicom).
- „victim.industry“: NAICS identifikator za poslovnu domenu organizacije (glavna djelatnost organizacije).
- „victim.country“: ISO 3166 identifikator države.
- „victim.state“: regija unutar države ili bilo koji podatak koji pobliže opisuje geografski položaj.
- „victim.employee_count“: približan broj zaposlenih unutar organizacije. VERIS ovu varijablu definira kao enumeraciju, a vrijednosti se nalaze unutar njihove sheme.
- „victim.revenue“: približni godišnji prihod. Sastoji se od brojčanog iznosa i valute (VERIS predstavlja valutu kao enumeraciju).
- „victim.locations_affected“: broj lokacija zahvaćen incidentom.
- „victim.notes“: ostale važne napomene vezane uz navedene varijable ove cjeline.

4.4. A4 model

VERIS omogućuje modeliranje prijetnji koje naziva A4 modelom, a sastoji se od krivaca za incident, njihovih akcija, imovina koje su zahvatili te na koji način (ili što) je ta imovina zahvaćena.

4.4.1. Krivci

Unutar incidenta mogu postojati jedan ili više krivaca (eng. „actors“) koji su skrivili incident. Njihove akcije mogu biti zlonamjerne, namjerne ili slučajne te uzročne ili doprinosne. VERIS definira tri glavne kategorije krivaca: vanjski (eng. „external“), unutarnji (eng. „internal“) i partner (eng. „partner“). Varijable koje se prema VERIS okviru mogu definirati su:

- „actor.X.motive“: koji su motivi doveli krivca da izazove incident. Na mjestu X može biti bilo koja od tri vrste krivaca. VERIS ovu varijablu definira kao enumeraciju.
- „actor.X.variety“: vrsta krivca. Na mjestu X mogu biti vanjski ili unutarnji krivac. VERIS ovu varijablu definira kao enumeraciju.
- „actor.X.origin“: zemljopisno porijeklo krivca (država). Na mjestu X mogu biti vanjski ili partner krivac. VERIS ovu varijablu definira kao enumeraciju.
- „actor.partner.industry“: NAICS identifikator za poslovnu domenu krivca.
- „actor.X.notes“: ostale važne napomene vezane uz krivca. Na mjestu X može biti bilo koja od tri vrste krivaca.

4.4.2. Akcije

Svaki incident ima barem jednu akciju koju je krivac izvršio, ne bi li izazvao incident. VERIS definira sedam primarnih kategorija (elementi unutar hijerarhije ispod „action“) za akcije krivca: maliciozni kod (eng. „Malware“), hakiranje (eng. „Hacking“), socijalni inženjering (eng. „Social“), zloupotreba (eng. „Misuse“), fizički napad (eng. „Physical“), greške (eng. „Error“) i prostorne nepogode (eng. „Environmental“). Varijable koje se prema VERIS okviru mogu definirati su:

- „action.X.variety“: vrsta konkretne akcije. Na mjestu X može biti bilo koja akcija. VERIS ovu varijablu definira kao enumeraciju.
- „action.X.vector“: na koji način ili zbog čega se dogodila ova akcija. Na mjestu X može biti bilo koja akcija osim prostorne nepogode. VERIS ovu varijablu definira kao enumeraciju.
- „action.X.cve“: identifikator ranjivosti (CVE) korištene kod akcije. Na mjestu X može biti maliciozni kod ili hakiranje.
- „action.malware.name“: naziv malicioznog koda dobiven npr. od strane antivirusne kuće koja je taj maliciozni kod otkrila.
- „action.social.target“: odgovara na pitanje „tko je meta napada socijalnim inženjeringom?“. VERIS ovu varijablu definira kao enumeraciju.
- „action.physical.location“: odgovara na pitanje „gdje su se dogodili fizički napadi?“

- „action.X.notes“: ostale važne napomene vezane uz krivca. Na mjestu X može biti bilo koja akcija.

4.4.3. Imovina

Svaki incident mora zahvatiti barem jednu imovinu unutar organizacije, koja je pretrpjela prijetnju ili štetu temeljem akcije krivca. Varijable koje se prema VERIS okviru mogu definirati su:

- „asset.variety“: količina (broj) i vrsta konkretne imovine. VERIS ovu varijablu (samo vrstu) definira kao enumeraciju.
- „asset.ownership“: odgovara na pitanje „tko je vlasnik imovine zahvaćene incidentom?“. VERIS ovu varijablu definira kao enumeraciju.
- „asset.management“: odgovara na pitanje „tko upravlja zahvaćenom imovinom?“. VERIS ovu varijablu definira kao enumeraciju.
- „asset.hosting“: odgovara na pitanje „tko ili što je poslužitelj imovine zahvaćene incidentom?“. VERIS ovu varijablu definira kao enumeraciju.
- „asset.accessibility“: odgovara na pitanje „kome je imovina zahvaćena incidentom (bila) dostupna?“. VERIS ovu varijablu definira kao enumeraciju.
- „asset.cloud“: ako je imovina smještenja unutar oblaka, unutar ove varijable potrebno je definirati dodatne faktore koji su potpomogli utjecaj ili širenje incidenta. VERIS ovu varijablu definira kao enumeraciju.
- „asset.notes“: ostale važne napomene vezane uz imovinu.

4.4.4. Svojstva

Sigurnosna svojstva imovine, koja su zapravo napadnuta tokom incidenta, VERIS dijeli u tri, odnosno u šest kategorija: povjerljivost/posjedovanje (eng. „confidentiality/possession“), integritet/autentičnost (eng. „integrity/authenticity“) i dostupnost/korisnost (eng. „accessibility/usefulness“). Povjerljivost je svojstvo koje se odnosi na ograničen pristup imovini. Gubitak povjerljivosti može značiti neovlašteni pristup imovini i pregled informacija koje su morale ostati tajne. Posjedovanje je svojstvo koje se odnosi na vlasnika imovine koji s pravom ima kontrolu nad imovinom te to može i dokazati. Gubitak posjedovanja znači isključivo krađu ili nemogućnost kontrole nad vlastitom imovinom. Integritet je svojstvo koje garantira da podaci nisu mijenjani od trenutka nastanka ili ovlaštenog mijenjanja. Gubitak integriteta znači bilo koji oblik povrede podataka koji uključuje mjenjanje vrijednosti ili oblika podataka. Autentičnost imovine je svojstvo kojim se imovina identificira istinitom i u skladu s namjerom. Gubitak autentičnosti znači neispravnost imovine. Dostupnost imovine je svojstvo koje imovinu označuje slobodnom i spremnom za korištenje. Gubitak dostupnosti stvara imovinu nedostupnom za korištenje. Korisnost imovine kao njezino svojstvo pokazuje koliko je

ta imovina prikladna za svrhu koju obavlja ili bi morala obavljati. Gubitak korisnosti čini imovinu smetnjom i/ili bespotrebnim troškom. Varijable koje se prema VERIS okviru mogu definirati su:

- „attribute.confidentiality.data_disclosure“: odgovara na pitanje „jesu li tajni podaci postali javno dostupni?“. VERIS ovu varijablu definira kao enumeraciju.
- „attribute.confidentiality.data.variety“: vrsta i količina (broj) podataka koji su neovlašteno otkriveni ili kompromitirani. VERIS ovu varijablu (samo vrstu) definira kao enumeraciju.
- „attribute.X.variety“: vrsta X svojstva koju je izgubila imovina. Ova varijabla analogna je varijabli „attribute.confidentiality.data.variety“, s time da je dostupna za ostala dva svojstva imovine. VERIS ovu varijablu definira kao enumeraciju.
- „attribute.confidentiality.state“: odgovara na pitanje „u vrijeme otkrića incidenta, je li imovina bila pohranjena, u prijenosu ili unutar procesa obrade?“. VERIS ovu varijablu definira kao enumeraciju.
- „attribute.availability.duration“: trajanje gubitka dostupnosti/korisnosti, sastoji se od broja i jedinice vremena, koju VERIS definira kao enumeraciju.
- „attribute.X.notes“: ostale važne napomene vezane uz svojstva imovine. Na mjestu X može biti bilo koja vrsta svojstva.

4.5. Informacije o pronalasku i odzivu

Informacije o pronalasku i odzivu sljedeća su cjelina unutar VERIS okvira koja definira vremenski period incidenta, kako je incident otkriven, koji je postupak sanacije te što je naučeno iz pruženog odziva incidentu. Ova cjelina pomaže organizaciji da se učinkovitije pripremi za slične incidente u budućnosti. Varijable koje se prema VERIS okviru mogu definirati su:

- „discovery_method“: odgovara na pitanje „kako je incident otkriven?“. VERIS ovu varijablu definira kao enumeraciju.
- „control_failure“: varijabla koja sadrži uzroke incidenta, tj. kvarove ili slabosti koje su doprinjele nastanku incidenta.
- „corrective_action“: sve akcije koje pridonose sprečavanju i/ili otkrivanju sličnih incidenata u budućnosti.
- „targeted“: odgovara na pitanje „je li se dogodio ciljani ili oportunistički napad?“. VERIS ovu varijablu definira kao enumeraciju.

4.6. Vremenski period

Što se tiče vremenskog perioda, odnosno vremenske crte, VERIS preporučuje evidentiranje sljedećih vremenskih trenutaka: početak maliciozne akcije nad žrtvom, prvo saznanje o

gubitku svojstava imovine, prvi događaj koji predstavlja povredu podataka, vrijeme kada je događaj prepoznat kao incident i vrijeme kada je incident saniran. Ipak, kako je navedeno u samoj dokumentaciji [28], VERIS nerijetko mijenja način praćenja vremenskog perioda. Varijable za vremenski period koje se prema VERIS okviru mogu definirati su:

- „timeline.incident.year“: godina kada se dogodio incident (tip podatka je broj).
- „timeline.incident.month“: mjesec kada se dogodio incident (tip podatka je broj).
- „timeline.incident.day“: dan kada se dogodio incident (tip podatka je broj).
- „timeline.incident.time“: vrijeme kada se dogodio incident (tip podatka je tekst, tj. „sat:minute:sekunde zona“).
- „timeline.compromise“: odgovara na „prvo saznanje o gubitku svojstava imovine“. Vrijednost varijable sastoji se od teksta koji predstavlja vrijeme i vremenske jedinice koju je VERIS enumerirao.
- „timeline.exfiltration“: odgovara na „prvi događaj koji predstavlja povredu podataka“. Vrijednost varijable sastoji se od teksta koji predstavlja vrijeme i vremenske jedinice koju je VERIS enumerirao.
- „timeline.discovery“: odgovara na „vrijeme kada je događaj prepoznat kao incident“. Vrijednost varijable sastoji se od teksta koji predstavlja vrijeme i vremenske jedinice koju je VERIS enumerirao.
- „timeline.containment“: odgovara na „vrijeme kada je incident saniran“. Vrijednost varijable sastoji se od teksta koji predstavlja vrijeme i vremenske jedinice koju je VERIS enumerirao.

4.7. Procjena štete

Procjenu štete teško je izmjeriti zbog širokog raspona (ne)materijalnih troškova, no čak i približan podatak važan je dio unutar evidentiranja incidenta. Tri su perspektive koje VERIS nudi svojim korisnicima u procjeni: vrsta gubitka povodom štete, jačina štete i utjecaj štete na organizaciju. Varijable koje se prema VERIS okviru mogu definirati su:

- „impact.loss.variety“: vrsta gubitka koja je proizašla iz ovog incidenta. VERIS ovu varijablu definira kao enumeraciju.
- „impact.loss.amount“: procjena gubitaka za vrstu gubitka. Tip vrijednosti varijable je broj.
- „impact.loss.min_amount“: donja granica za procjenu gubitka jedne vrste gubitka. Tip vrijednosti varijable je broj.
- „impact.loss.max_amount“: gornja granica za procjenu gubitka jedne vrste gubitka. Tip vrijednosti varijable je broj.

- „impact.loss.rating“: broj koji odgovara nekoj od vrijednosti iz vlastite ili normirane skale za procjenu štete jedne vrste gubitka.
- „impact.overall_amount“: procjena gubitka za cijelu organizaciju povodom incidenta. Tip vrijednosti varijable je broj.
- „impact.overall_min_amount“: donja granica za procjenu gubitka unutar cijele organizacije. Tip vrijednosti varijable je broj.
- „impact.overall_max_amount“: gornja granica za procjenu gubitka unutar cijele organizacije. Tip vrijednosti varijable je broj.
- „impact.iso_currency_code“: valuta za procjenjeni gubitak. VERIS ovu varijablu definira kao enumeraciju.
- „impact.overall_rating“: broj koji odgovara nekoj od vrijednosti iz vlastite ili normirane skale za procjenu štete unutar cijele organizacije.
- „impact.notes“: ostale važne napomene vezane uz procjenu štete.

U slučaju postojanja više gubitaka, tada se dodaje još jedna razina između „impact.loss“ elementa i varijabli, a to je broj koji označuje redni broj gubitka. Tako na primjer, postoje li dva gubitka unutar incidenta, vrsta i procjena gubitka za prvi definiraju se kao „impact.loss.1.variety“ i „impact.loss.1.amount“, a za drugi „impact.loss.2.variety“ i „impact.loss.2.amount“.

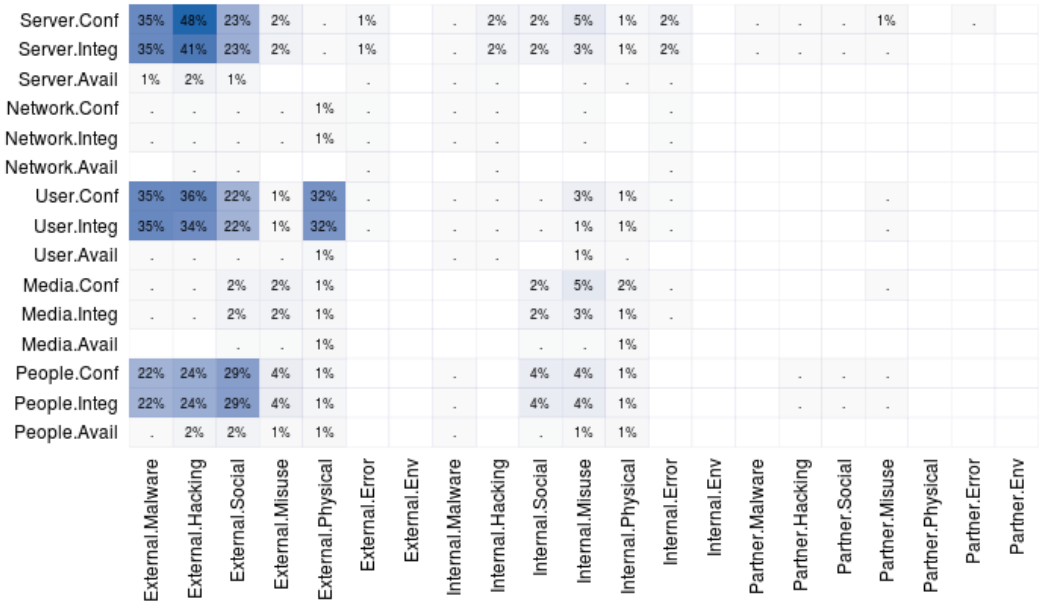
4.8. Indikatori kompromisa

Budući da se VERIS okvir primarno usredotočuje na strateške informacije te one koje se tiču upravljanja rizikom, dio „security intelliegence“-a poput indikatora kompromisa nije uključen u VERIS shemu. Unutar dokumentacije, za ovu svrhu preporuča se koristiti primjerenija shema poput STIX (eng. „Structured Threat Information Expression“). Postoje dvije varijable koje definira VERIS, a to su „ioc.indicator“ koji predstavlja naziv ili samu vrijednost indikatora i „ioc.comment“ koji opisuje indikator.

4.9. A4 mreža (grid)

Koristeći VERIS format u poslovanju, njegova struktura koja svojevrsno prati najbolje prakse olakšava mjerenje i odziv na incidente te modeliranje rizika u svojoj organizaciji. Jedna od cjelina VERIS okvira, nazvana A4, može se iskoristiti za praćenje trendova prijetnji i identificiranja izvora najčešćih ranjivosti unutar organizacije. Također, zbog postojanja otvorene baze evidentiranih incidenata u VERIS formatu, koja je nazvana DBIR, moguće je raditi usporedbe i pratiti trendove prijetnji unutar šire zajednice, kako bi se na vrijeme pripremili planovi i odradile procedure s ciljem da se incidenti pokušaju izbjeći. „A4 Grid“ svojevrsni je okvir unutar VERIS-a, predstavljen u obliku mreže sa x i y osi. Na x osi nalaze se

sve moguće kombinacije parova „actors-actions“, odnosno krivci i njihove radnje koje su prouzročile incident. Na y osi nalaze se sve moguće kombinacije parova „assets-attributes“, odnosno imovina i njihova svojstva koja su napadnuta. Unutar jednog incidenta može se definirati između jedne i 315 kombinacija, odnosno presjeka parova sa x i y osi. Parovi nisu nasumično raspoređeni po mreži, već stvaraju određenu logičnu cjelinu (ovo se tiče samo „actors“ i „assets“). Upravo zbog toga, moguće je nakon svakog incidenta izvaditi podatke o A4, razvrstati ih po „A4 Grid“-u na način da se odgovarajući presjeci inkrementiraju, a nakon određenog vremena napraviti mapu kritičnih točaka (eng. „Heat-map graph“) i analizirati gdje se nalazi najviše propusta, odnosno zapitati se zašto se baš ta imovina i njezino svojstvo napadaju određenom radnjom. Jedan od primjera mape kritičnih točaka, temeljen na „A4 Grid“-u nakon nekoliko incidenata, nalazi se na slici 1. Naravno, nisu ni ostale vrste vizualizacije isključene kod prikaza kombinacija parova sa x i y osi. Bitno je da sigurnosni stručnjaci mogu uz pomoć strukturiranih podataka donjeti bolje odluke (eng. „Data-driven decisions“).



Slika 1: Primjer „A4 Grid“ mape kritičnih točaka [28]

4.10. Primjeri upotebe VERIS okvira za opis incidenata

4.10.1. Incident #1

Zaposlenici poduzeća „Digidi DigiDi LLC“, odjeli prodaje i službe za korisnike, dobili su e-mail 23.8.2018. sa MS Word dokumentom kao predloškom kojega su preuzeli i otvorili, jer je u sadržaju poruke izgledao važno za njihovo poslovanje. Dokument je zapravo uz sebe imao maliciozni kod nazvan Felixroot, koji nakon što je pokrenut, preuzima drugu razinu malicioznog koda i u tajnosti se izvršava na računalu. Daljnom istragom te odzivom na incident, pronađeno je kako maliciozni kod šalje na IP adresu 205.93.40.123 sustavske

podatke o žrtvinom računalu i podatke web preglednika. Nije utvrđeno kolika je cjelokupna šteta, no čišćenje diskovnog prostora računala koštalo je 4400 kn.

```
{
    "incident_id":  1,
    "schema_version": "1.3.1",
    "security_incident": "Confirmed",
    "summary": "Employees donwloaded malicious mail attachment which
       turns out to be a backdoor with C2 communication ability.",
    "confidence": "High",
    "victim": [
        {
            "country": "HR",
            "employee_count": "11 to 100",
            "industry": 54121,
            "revenue": "1000000 HRK",
            "state": "ZG",
            "victim_id": "Digidi Digidi LLC"
        }
    ],
    "actor": {
        "external": {
            "motive": [
                "Espionage"
            ],
            "variety": [
                "Unknown"
            ],
            "notes": "Chances that Lazarus Group APT done this are
               quite high, because of its connection with Felixroot."
        }
    },
    "action":{
        "malware": {
            "variety": [
                "Backdoor",
                "C2",
                "Downloader"
            ],
            "vector": [
                "Email attachment"
            ],
            "name": "Felixroot",
            "cve": [
```

```
                "CVE-2017-0199",
                "CVE-2017-11882"
            ]
        },
        "social": {
            "variety": [
                "Phishing"
            ],
            "vector": [
                "Email"
            ],
            "target": [
                "Finance",
                "Helpdesk"
            ]
        }
    },
    "asset": {
        "variety": [
            "U - Desktop",
            "U - Laptop"
        ],
        "ownership": [
            "Employee"
        ],
        "accessibility": [
            "Internal"
        ],
        "management": [
            "Internal"
        ]
    },
    "attribute": {
        "confidentiality": {
            "data": {
                "variety": [
                    "System"
                ]
            }
        }
    },
    "discovery_method": "Int - unknown",
    "corrective_action": "Security Awareness",
```

```
    "targeted": "Targeted",
    "timeline": {
        "incident": {
            "year": "2018",
            "month": "08",
            "day": "23"
        }
    },
    "impact": {
        "overall_min_amount": "4400 HRK",
        "overall_rating": "3",
        "notes": "Investigation about overall loss is this in process
           ."
    },
    "ioc": [
        {
            "indicator": "205.93.40.123",
            "comment": "An IP address of C2 server."
        }
    ],
    "plus": {
        "victims_response": "Document, along with e-mail content,
           looked very legit."
    }
}
```

4.10.2. Incident #2

Zaposleniku M. C. poduzeća „S4M0 J4K0 d.o.o.“ nestalo je poslovno prijenosno računalo. Zadnji puta računalo je viđeno 23.8.2018. oko 15h na uredskom stolu zaposlenika M. C. (vlasnika). Pristup uredima nema nitko osim zaposlenika istog kata zgrade. Prijenosno računalo nema kriptiran diskovni prostor, no postoji lozinka za prijavu korisnika u operacijski sustav za koju vlasnik tvrdi da nitko osim njega ne zna. Vlasnik tvrdi da u spremištu računala postoje podaci koji su poslovna tajna. Cijela računala bila je 10.000 kn za vrijeme kupnje, ostali troškovi nisu još poznati.

```
{
    "incident_id":  2,
    "schema_version": "1.3.1",
    "security_incident": "Confirmed",
    "summary": "Employee's business notebook has been stolen, victim
       claims that business secrets are stored on it.",
    "confidence": "Medium",
```

```
    "victim": [
        {
            "country": "HR",
            "employee_count": "Small",
            "industry": 54161,
            "revenue": "22000000 HRK",
            "state": "KC",
            "victim_id": "S4M0 J4K0 d.o.o."
        }
    ],
    "actor": {
        "internal": {
            "motive": [
                "Unknown"
            ],
            "variety": [
                "Unknown"
            ]
        }
    },
    "action":{
        "physical": {
            "variety": [
                "Theft"
            ],
            "vector": [
                "Privileged access"
            ],
            "location":[
                "Victim work area"
            ]
        }
    },
    "asset": {
        "variety": [
            "1 U - Laptop"
        ],
        "ownership": [
            "Victim"
        ],
        "accessibility": [
            "Internal"
        ],
```

```
        "management": [
            "Victim"
        ]
    },
    "attribute": {
        "confidentiality": {
            "data_disclosure": "Potentially",
            "data": {
                "variety": [
                    "Secrets"
                ]
            },
            "state": [
                "Stored unencrypted"
            ]
        },
        "availability": {
            "variety": [
                "Loss"
            ],
            "duration": [
                "Unknown"
            ]
        }
    },
    "discovery_method": "Int - reported by user",
    "corrective_action": [
        "Checking security cameras",
        "Locking business stuff more securely"
    ],
    "targeted": "Unknown",
    "timeline": {
        "incident": {
            "year": "2018",
            "month": "08",
            "day": "23",
            "time": "15:00:00"
        }
    },
    "impact": {
        "loss":{
            "variety": "Business disruption",
            "amount": 10000
```

```
        },
        "overall_amount": "Unknown",
        "overall_rating": "2"
    },
    "plus": {
        "employee_initials": "M. C."
    }
}
```

4.10.3. Incident #3

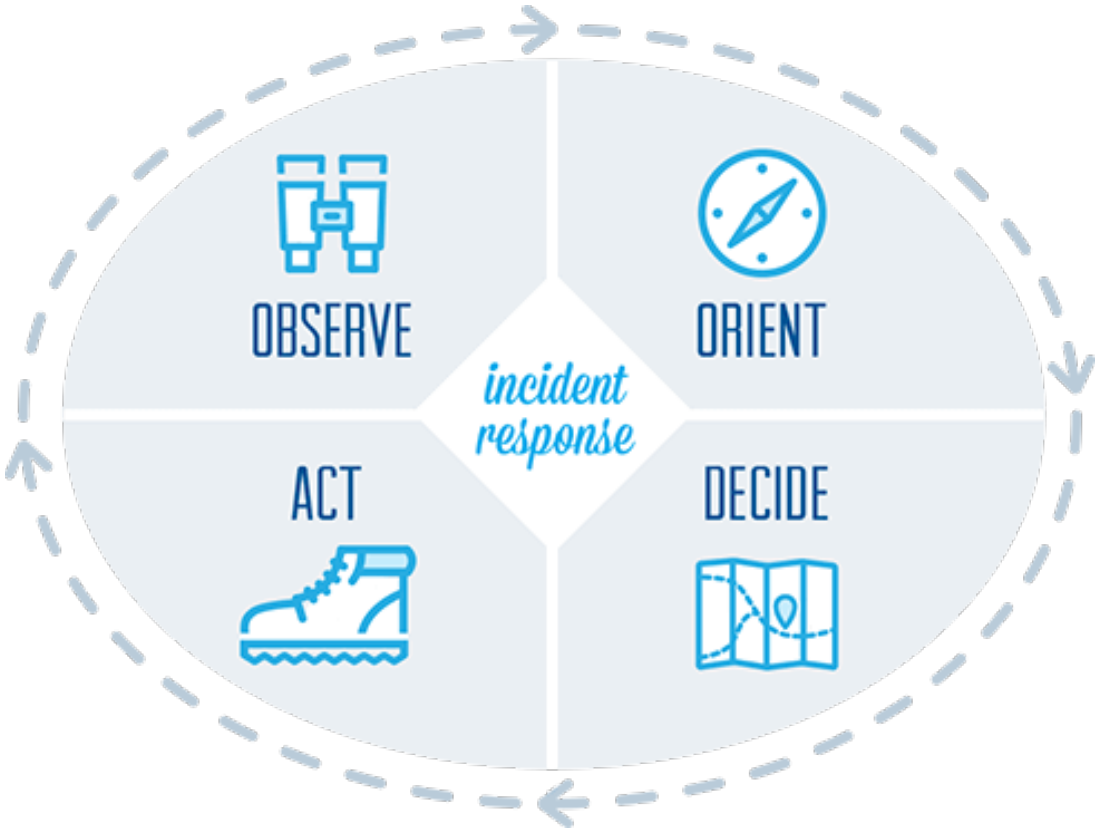
Unutar SOC odjela jednog poduzeća otkrivena je sumnjiva radnja (otvraranje sumnjive izvršne datoteke) unutar „temp“ mape na Windows računalu. SIEM je ovu anomaliju otkrio u 11:44:21 23.8.2018. te je poslao alarm kojega je analitičar pročitao oko 14h istog dana.

```
{
    "incident_id":  3,
    "schema_version": "1.3.1",
    "security_incident": "Suspected",
    "summary": "Suspicious executable in temp folder.",
    "confidence": "High",
    "actor": {},
    "action": {
        "misuse":{
            "variety": [
                "Unknown"
            ]
        }
    },
    "asset": {
        "variety": [
            "1 U - Desktop"
        ],
        "ownership": [
            "Victim"
        ],
        "accessibility": [
            "Internal"
        ],
        "management": [
            "Victim"
        ],
        "notes": "Asset runs Windows OS."
```

```
    },
    "attribute": {},
    "discovery_method": "Int - log review",
    "control_failure": "Windows 'temp' folder",
    "corrective_action": [
        "Software/Executable whitelisting"
    ],
    "timeline": {
        "incident": {
            "year": "2018",
            "month": "08",
            "day": "23",
            "time": "11:44:21"
        },
        "discovery": "3 Hours"
    },
    "impact": {}
}
```

5. OODA petlja - okvir za izradu metodologija

OODA petlja jedan je od prisupa definiranju metodologije u slučaju incidenta. Pristup je izmislio vojni strateg John Boyd [29]. Ono što čini ovu „petlju“ su četiri cikličke faze (prikazane na slici 2): promatranje (eng. „Observer), usmjerenje (eng. „Orient“), odluka (eng. „Decide“) i djelovanje (eng. „Act“). Za svaku fazu potrebno je definirati alate i tehnike, pitanja na koja je potrebno odgovoriti te ključne čimbenike koji se tokom faze moraju uzeti u obzir.



Slika 2: OODA petlja u izgradnji metodologija odziva na incidente [29]

5.1. Promatranje

Kako bi plan odziva na incidente bio učinkoviti, potrebno je uzeti u obzir sigurnost organizacije s holističke perspektive – teško je očekivati kvalitetan pronalazak prijetnji tako dugo dok plan ne uzima u obzir kritične točke unutar organizacije. Sukladno tome, ako plan ne uzima u obzir korištenje „Threat intelligence“, organizacija je ranjivija na prijetnje koje vrebaju na internetu. S druge strane, praćenje svih „Threat Intelligence Feed“-ova nije efikasno – ovisno o domeni, tehnologiji i sposobnostima organizacije, sigurnosni stručnjaci će sigurno trebati napraviti prioritete (poput definiranja razine ozbiljnosti za svaki nadolazeći „Intelligence“) te prema njima alarmirati SOC odjel. Kako bi se kvalitetno provela faza promatranja, u alate i tehnike potrebno je uključiti analize ranjivosti, rangiranje prijetnji, SIEM alarme, IDS alarme, sustave za praćenje aplikacijskih performansi i alate za analizu mrežnog toka podataka. Pitanja koja je potrebno postaviti su: Kako izgleda normalna aktivnost na mreži? Kako pronaći i kategorizirati

sumnjive događaje i potencijalne prijetnje. Kako podesiti infrastrukturu za praćenje sigurnosti unutar vlastite organizacije? Tokom faze promatranja, potrebno je zapamtiti da što se češće dokumentiraju zapažanja unutar poslovne mreže, veće su šanse ranijem sprječavanju incidenta.

5.2. Usmjerenje

Više „Intelligence“-a ne znači nužno kvalitetniju istragu. Može se stvoriti suprotan učinak – previše podataka može odužiti istragu, a samim time i odziv. Stoga je potrebno izdvojiti kontekst, korelirati razne događaje i usredotočiti se na upravo te podatke. Drugim riječima, potrebno je provesti fazu usmjerenja unutar OODA petlje. Jedan od primjera bio bi nenadano isključivanje rada poslužitelja u podatkovnom centru. Jedan od razloga mogao bi biti nestanak struje, ali i DDoS napad na poslužitelj. Bez konteksta, kao što bi bio e-mail od ISP-a koji obavještava o gubitku struje na jednom djelu podatkovnog centra, jako je teško riješiti incident na ispravan način. Od alata i tehnika tokom ove faze potrebna su sigurnosna istraživanja i korelacije, zatim alati za trijažu incidenata i tehnika dobivanja svjesnosti o situaciji. Pitanja koja se postavljaju vezana su uz prikupljene podatke iz prošle faze. Primjeri pitanja: Sprema li se organizacija na prelazak nove tehnologije ili će samo ažurirati postojeće? Je li itko od zaposlenika već prije primijetio ovu IP adresu? Koji je uzrok nastanka incidenta? Kakva i kolika je šteta nastala ovim incidentom? Ključna činjenica tokom ove faze je razmišljanje „van okvira“ – potrebno je premjestiti se u perspektivu napadača. Na taj način moguće je doznati nove kritične točke te ih na vrijeme zakrpati.

5.3. Odluka

Prošle faze najveću su korist imale od automatiziranih alata za prikupljanje i analizu podataka. Ova faza, odlučivanje, temelji se isključivo na sigurnosnim stručnjacima i sigurnosnim politikama organizacije zahvaćene incidentom. Često se timovi poput CSIRT-a susreću sa poteškoćama u odabiru između što boljeg očuvanja dokaza i što bržeg odziva zbog oporavka. Od alata i tehnika, potrebni su materijali za dokumentiranje i sigurnosne politike organizacije. Kada su sve činjenice na mjestu i istraga je završila, potrebno je dogovoriti se sa vlastitim timom stručnjaka koji su sljedeći potezi, tj. plan za fazu djelovanja. Preporučeno je napraviti spisak svih područja nad kojim će se djelovati te obavijestiti nadređene o daljinom postupku, jer ponekad je potrebno zatražiti dodatne dozvole za određene lokacije.

5.4. Djelovanje

Zadnja faza ima sljedeće ciljeve:

- Odraditi sanaciju štete što je brže moguće nad zahvaćenom imovinom.
- Ažurirati i po potrebni ponovo provesti „Security Awareness“ edukaciju unutar organizacije te ažurirati sigurnosne politike i/ili plan za odziv.

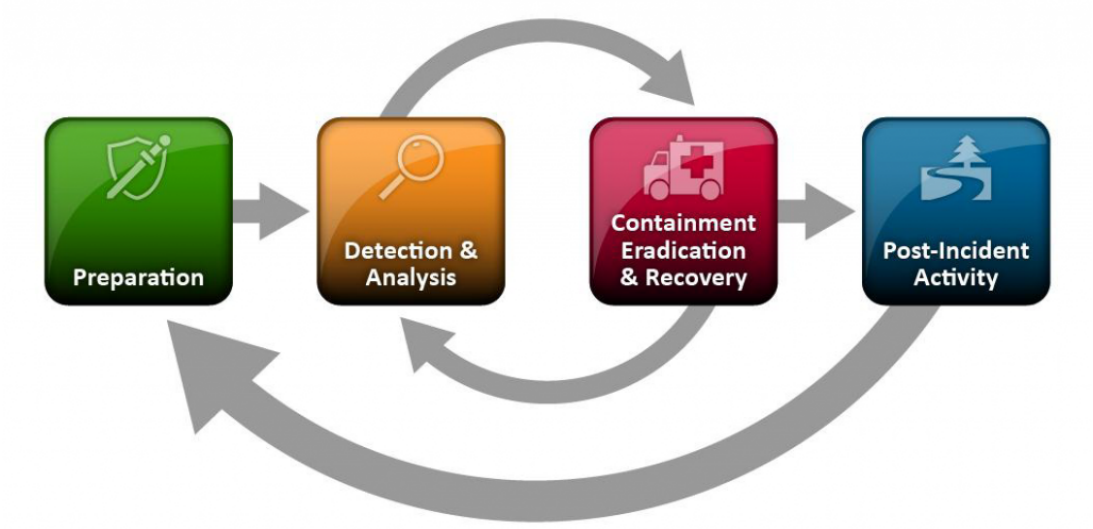
- Po potrebi ponovo konfigurirati alate za praćenje i analizu događaja, kako bi se prve dvije faze u slušaju sličnog događaja mogle učinkovitije odraditi.

Teško je nabrojati sve moguće aktivnosti sanacije, jer one ovise o vrsti incidenta ili prijetnje, ali i zahvaćenoj imovini. Na primjer, neke od aktivnosti sanacije mogu biti: postavljanje zakrpa na sustav, otklanjanje nepotrebnog softvera, ponovna konfiguracija sustavnih datoteka, implementacija ACL-a na usmjernike, pisanje novih pravila za vatroštit, ponovno postavljanje korisničkih lozinki, itd. Od alata i tehnika potrebnih za ovu fazu su alati za oporavak i vraćanje „backup"-a, forenzički alati, sustavi za upravljanje zakrpama te materijali za „Security Awareness". Pitanja koja se postavljaju tokom faze su: Koliko je potrebno vremena da se zahvaćeni sustavi saniraju i vrate u prvotno stanje? Kako možemo educirati zaposlenike radi smanjenja mogućnosti ponovne pojave ovog incidenta u budućnosti? Potrebno li je ponovno konfigurirati naše sustave praćenja ili je čak potrebno ažurirati naše poslovne procese? Ključni čimbenik ove faze je organizacija posla i saznanje o ulogama u timu koji odrađuje odziv. S vremena na vrijeme potrebno je osvježiti sposobnosti stručnog tima, kako bi bili „uz korak" sa novim tehnologijama, metodama i ostalim dobrim praksama za odziv na incidente.

6. NIST metodologija

Nacionalna ustanova za norme i tehniku, skraćeno NIST, bavi se mjeriteljstvom, normiranjem, fundamentalnim i primijenjenim istraživanjima, pomažući gospodarstvu uvođenjem novih tehnika i postupaka, poboljšanju kakvoće, povećanju učinkovitosti i smanjenju troškova proizvodnje [30]. Njihov pregledni članak „SP 800-61 (revision no2)“ u jednom od svojih poglavlja navodi metodologiju za odziv na incidente. Potrebno je spomenuti kako je ovaj članak jedan od onih literatura koje pojam „Handling“ koriste za oba procesa: upravljanje i odziv na incidente.

Inicijalna faza uključuje organiziranje, opskrbu i edukaciju stručnog tima za odziv na incidente. Tokom pripreme, cilj spomenutog stručnog tima je ograničiti broj potencijalnih incidenata tako da se implementiraju potrebne zaštitne mjere (politike, uređaji, svijest o sigurnosti. . .) temeljene na rezultatima procesa procjene rizika. Naravno, nije moguće obuhvatiti sve rizike, niti implementirati takve mjere koje će pokriti određene rizike u potpunosti. Otkrivanje upada ili bilo kakvog sigurnosnog prekršaja uvjet je alarmiranja organizacije o aktivnom incidentu. U odnosu na ozbiljnost incidenta, organizacija može sanirati štetu ograničavanjem njezina djelovanja, a zatim i vraćanjem zahvaćenog dijela organizacije u stanje prije incidenta. Tokom ove faze (3. faza po redu), nerijetko se stručnjaci vraćaju na prošlu fazu (otkrivanje i analiza) – npr. potrebno je pogledati postoji li i dalje zaraženih računala nakon suzbijanja djelovanja malicioznog koda sa računala na kojemu je zaraza najprije otkrivena. Nakon što je oporavak završen i incident kvalitetno suzbijen, potrebno je napisati izvještaj. Takav dokument mora sadržavati detalje poput uzroka incidenta i troškova, ali i kako poboljšati učinkovitost koraka unutar svih faza odziva na ovaj (ili slični) incident u budućnosti. Slika 3 prikazuje sažeti opis metodologije u obliku životnog ciklusa odziva na incidente.



Slika 3: Životni ciklus odziva na incidente [30]

6.1. Faza pripreme

Metodologije za odziv na incidente naglašavaju važnost faze pripreme. Kao što je prije navedeno, unutar faze pripreme koja se događa prije pojave incidenta, potrebno je organizirati stručne timove za odziv i osigurati IT infrastrukturu raznim sustavima za praćenje i sprečavanje anomalija. Organizacija bi trebala imati nekoliko komunikacijskih i koordinacijskih mehanizama i postrojenja za upravljanje incidentom. Neki od tih mehanizama i postrojenja mogu biti:

- Kontakt informacije zaposlenika (e-mail, brojevi telefona, javni ključevi za kriptiranje. . .).
- Raspored radnih smjena i eskalacija odgovornosti.
- Mehanizam za prijavu incidenta, poput foruma ili usluge instant poruka.
- Sustav praćenja informacija o incidentima (npr. „Ticketing“ sustav).
- Poslovni pametni telefoni.
- Softver za enkripciju.
- Stalna ili po potrebi soba za centralnu komunikaciju i koordinaciju u izvanrednim slučajevima (eng. „War Room“).
- Sigurno (kriptirano) skladište za pohranu dokaza.

Također, u fazi pripreme potrebno je opskrbiti stručni tim sa potrebnim alatima poput:

- Specijalne radne stanice za obavljanje forenzike (npr. operacijski sustavi konfigurirani sa svi potrebnim alatima).
- „Backup“ uređaji za kopije dokaza, datoteka zapisa, itd.
- Prijenosna računala za analizu podataka, praćenje mrežnih paketa i analizu protokola, pisanje izvještaja, itd.
- Prijenosni diskovni prostor, poput vanjskih ili USB diskova.
- Prijenosni printeri.
- Softver za potrebe digitalne forenzike.
- Pribor za sakupljanje dokaza.

Od materijala potrebnih za analizu incidenta (priprema za drugu fazu), tu su:

- Popis korištenih portova sa njihovim opisima.
- Dokumentacije operacijskih sustava, aplikacija, korištenih protokola te proizvoda poput IDPS, vatroštita, antivirusnog rješenja, itd.

- Mrežni dijagrami i popis kritične imovine (npr. baze podataka).
- Kriptografski saželi kritičnih datoteka, koji pomažu kod verifikacije, analize incidenta te pomoći kod faze iskorjenjivanja.

Vrlo često timovi poput CSIRT-a osmisle svoju posebnu naprtnjaču, popularno nazvanu „jump kit“ ili „jump bag“. Unutra se nalaze svi potrebni alati za istragu incidenta, neovisno o njegovoj lokaciji. Svaka takva naprtnjača najčešće se sastoji od prijenosnog računala (sa instaliram softverom za mrežnu i računalnu forenziku), zatim nekoliko „backup“ medija poput CD/DVD i USB uređaja te raznih drugih mrežnih uređaja i poveznica („kablova“). Pravilo je da se sadržaj naprtnjače ne „posuđuje“, već se svaki puta nakon korištenja brišu svi podaci koji su nakupljeni tokom aktivnosti forenzike i prijenosa dokaza u trajnije skladište/medije. Također, za sigurnosnog stručnjaka koji na lokaciju incidenta dolazi sa naprtnjačom, preporuča se nošenje dodatnog prijenosnog računala za pisanje izvještaja, komunikaciju sa ostatkom tima, itd.

Držanje što manjeg broja incidenata unutar organizacije je veoma bitno, prvenstveno radi očuvanja poslovnog kontinuiteta, ali i zbog sljedeće činjenice – ako sigurnosne kontrole nisu „na mjestu“, veća je šansa nastanka velikog broja incidenata, preplavljujući tako tim za odziv na incidente. To može biti uzrok sporim i nedovršenim odzivima, koji stvaraju negativan utjecaj na poslovanje. Neke od preporučenih praksi koje je potrebno provesti tokom osiguravanja mreže, aplikacija i ostatka IT infrastrukture su:

- procjene rizika
- sigurnost računala/poslužitelja
- sigurnost mreže
- prevencija zlonamjernog softvera
- podizanje svijesti korisnika uz edukaciju

6.2. Faza otkrivanja i analize

Druga faza ove metodologije temelji se na otkrivanju i analizi incidenta. Proces otkrivanja opisuje vektore napada, najčešće znakove i pokazatelje incidenta unutar organizacije, izvore indikatora kompromisa, odnosno agregatore datoteka zapisa i generatora alarma. Proces analize opisuje korištenje otkrivenog, dokumentiranje i prioritiziranje analiziranog i obavještavanje nadležnih tijela/odjela. Nemoguće je popisati sve vektore napada, tj. način ili medij koji je iskorišten kako bi se izveo napad. Ipak, NIST definira sljedeće vektore kao one najčešće:

- Vanjski fizički mediji poput USB uređaja.
- Napad surovom snagom ili „na silu“ (eng. „Brute force attack“) - poput uskraćivanja usluge ili pogađanja lozinke naslijepo.

- Web – pogotovo web aplikacije koje koriste korisničku interakciju i slanje podataka na poslužitelj.
- Email - pogotovo priložene datoteke koje mogu sadržavati maliciozni sadržaj.
- Lažno predstavljanje u svrhu prijevare žrtve.
- Nepravilna upotreba, često slučajna, od strane zaposlenika organizacije.
- Gubitak ili krađa opreme.

Za većinu organizacija, najteži dio upravljanja i odziva na incident je točna procjena ili identifikacija mogućih incidenata – utvrđivanje je li događaj stvarno incident te ako je, koje je vrste, opsega zahvata i jačine. Tri su faktora koja čine ovaj dio procesa kompliciranim:

- Incidenti mogu biti identificirani od strane automatiziranih alata poput SIEM-a koji šalje alarme, no također mogu biti prijavljeni od strane običnog zaposlenika. Ipak potrebno je u oba slučaja istražiti radi li se zbilja o incidentu ili ne.
- Broj znakova potencijalnih incidenata općenito je visok – ovisno o veličini mreže, nije neobično da IDS unutar organizacije zadobije nekoliko tisuća ili čak milijuna alarma na dan.
- Kako bi se događaji korelirali te prepoznali kao incident, potrebno je određeno znanje i iskustvo analitičara.

Znakovi koji upućuju na incident dijele se na dvije kategorije: glasnici i indikatori. Glasnici (eng. „Precursors“) su znakovi koji upućuju na pojavu incidenta u budućnosti. Indikatori upućuju na trenutno djelovanje incidenta ili na završeni incident. U praksi, indikatori su puno češći nego glasnici. Primjeri glasnika su:

- Datoteke zapisa poslane od strane web poslužitelja koje pokazuju aktivnost skenera ranjivosti.
- Dojava o novom malicioznom kodu koji iskorištava ranjivost mail poslužitelja unutar organizacije.
- Prijetnja unutar koje piše kako će se napasti organizacija.

Primjeri indikatora:

- Alarm antivirusnog rješenja koji upozorava na infekciju malicioznim kodom.
- Web aplikacija zapisuje nekoliko uzastopnih neuspjelih pokušaja prijave sa nepoznatog vanjskog sustava.
- Mrežni administrator primjećuje neobične devijacije kod protoka mrežnih paketa.

Obje kategorije znakova moraju biti zamijećene unutar IT infrastrukture ili golim okom sigurnosnih stručnjaka. Najčešći izvori glasnika i indikatora su:

- IDPS - uz pomoć svoje baze sažetaka malicioznih zapisa i sustava alarmiranja.
- SIEM - uz pomoć sustava alarmiranja.
- Antivirusna rješenja - uz pomoć generiranje izvješća i obavijesti o zarazi.
- Softver za provjeru integriteta datoteka.
- Usluge praćenja od trećih strana.
- Datoteke zapisa – operacijski sustav, procesi i aplikacijske datoteke zapisa.
- Mrežni paketi.
- Informacije o novim ranjivostima i malicioznim kodovima, odnosno „Threat Intelligence“.
- Zaposlenici i partneri.

Faza otkrivanja i analize bila bi lagana kada bi svaki glasnik ili indikator mogao garantirati preciznost u svojim najavama incidenta. U praksi to nije slučaj, stoga je potrebna ljudska analiza. Inicijalna analiza i validacija znaju biti zahtjevne, stoga NIST propisuje nekoliko savjeta za učinkovitu analizu:

- „Profiling“ mreža i računalnih sustava, gdje „Profiling“ označuje mjerenje karakteristika očekivane aktivnosti.
- Znanje o normalnim ponašanjima unutar mreže i računalnih sustava.
- Izrada pravila za zadržavanje datoteka zapisa.
- Korelacija događaja nakon sakupljanja mogućih dokaza.
- Korištenje NTP protokola kako bi vremena na računalima bila sinkronizirana.
- Korištenje baze znanja o pojedinom incidentu.
- Korištenje internetskih tražilica za istraživanje.
- Hvatanje mrežnih paketa zbog dohvaćanja dodatnih podataka, odnosno potencijalnih dokaza.
- Filtriranje podataka.
- Traženje pomoći i asistencije iz drugih dijelova organizacije.

Stručni tim koji sumnja na nastali incident, odmah mora početi zapisivati sve činjenice vezane uz taj incident. Alati za zapisivanje navedeni su unutar prve faze. Sustav praćenja informacija o incidentima trebao bi sadržavati sljedeće podatke:

- Trenutni status incidenta (otvoreno/zatvoreno ili detaljnija podjela stanja).
- Sažetak incidenta.
- Indikatori vezani uz incident.
- Drugi incidenti koji su vezani uz ovaj.
- Akcije koje je poduzeo stručni tim.
- Nadzor i hijerarhija, ako postoji.
- Procjena utjecaja i štete uzrokovane incidentom.
- Kontakt informacije za ostale sudionike u rješavanju incidenta.
- Popis dokaza sakupljenih tokom istrage.
- Komentari stručnog tima.

Incidentima se ne smije pristupiti FIFO redoslijedom rješavanja, već ih je potrebno poredati prema prioritetu. Prioritet se radi prema sljedećim faktorima:

- Utjecaj incidenta na poslovanje (funkcionalni utjecaj).
- Utjecaj incidenta na svojstva informacija (tajnost, integritet i dostupnost).
- Trajanje oporavka od incidenta.

Utjecaj incidenta na poslovanje može imati sljedeće vrijednosti:

- Nema utjecaja – organizacija može normalno pružati sve svoje usluge korisnicima.
- Nizak – organizacija može pružati sve svoje usluge korisnicima, uz malo izgubljene učinkovitosti.
- Srednji – organizacija je izgubila sposobnost pružanja usluga jednom dijelu svojih korisnika.
- Visoki – organizacija više nije u mogućnosti pružati svoje usluge korisnicima.

Utjecaj incidenta na svojstva informacija može imati sljedeće vrijednosti:

- Nema utjecaja – Nema podataka koji su ukradeni, promijenjeni, izbrisani ili na neki drugi način oštećeni.
- Kršenje privatnosti – Neovlašteno su pregledani ili preuzeti osjetljivi osobni podaci poreznih obveznika, zaposlenika, korisnika itd.
- Vlasnički prekršaj – Neovlašteno su pregledane ili preuzete nerazvrstane vlasničke informacije poput onih o zaštićenoj kritičnoj infrastrukturi.

- Gubitak integriteta – Neovisno o vrsti podataka, integritet je narušen.

Trajanje oporavka od incidenta može imati sljedeće vrijednosti:

- Regularno – Vrijeme potrebno za oporavak predvidljivo je sa postojećim resursima.
- Dodatno – Vrijeme potrebno za oporavak predvidljivo je sa dodatnim resursima.
- Prošireno – Vrijeme potrebno za oporavak nije predvidljivo, tj. potrebni su dodatni resursi i vanjska pomoć, no količina nije određena.
- Oporavak nije moguć – Primjer: tajni podaci su ukradeni i javno objavljeni.

Tokom koraka obavještavanja drugih odjela o incidentu, uobičajeno je obavještavati odjele/zaposlenike poput: direktora informacijskih tehnologija, direktora informacijske sigurnosti, ostale sigurnosne stručne timove unutar organizacije, vlasnike sustava zahvaćenog incidentom, odjel za ljudski potencijal, odnose s javnošću, pravnu službu, državni CERT, itd. Također potrebno je odrediti, ovisno o vrsti incidenta, kanal za obavještavanje. U nekim slučajevima dovoljno je koristiti tehnologije poput e-maila ili foruma, ali postoje i slučajevi gdje je moguće koristiti samo fizički kontakt na strogo povjerljivim lokacijama.

6.3. Faza ograničavanja, iskorjenjivanja i oporavka

Treća faza uključuje ograničavanje djelovanja incidenta, što je ključni proces u sprječavanju širenja zaraze ili činjenja daljnje štete organizaciji. Glavna aktivnost unutar ovog procesa je odlučivanje, koje mora biti što brže i što učinkovitije. Odlučivanja se temelje na strategijama ograničavanja, koja se razlikuju prema vrsti incidenta. Uglavnom, kriteriji za određivanje odgovarajuće strategije uključuju:

- Potencijalna šteta i krađa podataka ili ostalih resursa.
- Potreba za očuvanjem dokaza.
- Dostupnost usluge.
- Vrijeme i ostali resursi potrebni za implementaciju strategije.
- Efikasnost strategije.
- Trajanje rješenja.

Iako je glavni razlog sakupljanja dokaza pronalazak rješenja koje će „zatvoriti“ incident, jedan od razloga mogu biti i pravni postupci. U takvim slučajevima, važno je dokumentirati kako su sačuvani svi dokazi, uključujući i ugrožene dijelove sustava. Također, potrebno je voditi detaljan zapisnik svih dokaza, uključujući i zapise poput:

- Identifikacijskih podataka (lokacija, serijski broj, MAC adresa, itd.).
- Puno ime, titula/pozicija i kontakt broj svake osobe koja je sudjelovala u istrazi.
- Datum i vrijeme svake aktivnosti upravljanja dokazima.
- Lokacije gdje su dokazi spremljeni.

Opcija lova na napadače (eng. „Hunting“) nije isključena unutar treće faze, ali preporučuje se ostati usredotočen na procese ograničavanja, iskorjenjivanja i oporavka. Lov, odnosno identifikacija napadačevih resursa može stvoriti velike vremenske troškove i odvratiti stručni tim od njihovog primarnog cilja – minimiziranja utjecaja incidenta na poslovni rezultat. Najčešće aktivnosti kod identifikacije napadača:

- Validacija IP adrese napadačeva računala.
- Korištenje Internet tražilice sa upitima sastavljenim od napadačevih informacijama.
- Korištenje baza podataka o incidentima.
- Praćenje potencijalnih kanala komunikacije sa napadačevim C2 poslužiteljem.

6.4. Faza aktivnosti nakon riješenog incidenta

Svaki sigurnosni stručni tim sakuplja iskustvo i evoluira rješavajući incidente, ali i učeći na vlastitim greškama i ažuriranjem sigurnosnih politika i strategija. Držanjem „što smo naučili“ sastanka sa svim strankama uključenima u riješeni incident, moguće je znatno povećati učinkovitost sljedećeg rješavanja. Sastanak može uzeti u obzir i više incidenata, koji su na neki način u relaciji, bitno da ne postoji preveliki vremenski razmak. Pitanja koja je poželjno postaviti na sastanku su:

- Što se i kada točno dogodilo?
- Koliko učinkovito su osoblje i menadžment pristupali identifikaciji i otklanjanju incidenta? Jesu li se dokumentirane procedure slijedile i ispoštovale?
- Koje informacije su bile hitne, odnosno najprije potrebne?
- Jesu li počinjenje radnje koje bi spriječile oporavak?
- Što bi se trebalo promijeniti sljedeći puta kada nastupi isti ili sličan incident?
- Kako poboljšati dijeljenje informacija među organizacijama?
- Kakve znakove pratiti u budućnosti kako bi se što prije otkrili slični incidenti?
- Koji dodatni alati ili resursi su potrebni kako bi se otkrili, analizirali i sanirali budući incidenti?

Organizacije se trebaju usredotočiti na prikupljanje onih podataka koji čine organizaciju djelotvornijom, a ne na prikupljanje podataka jednostavno zato što su podaci kao takvi dostupni. Moguće metrike koje podatke označuju djelotvornima za buduće poslovanje, uključuju:

- Broj riješenih incidenata.
- Vrijeme potrebno za rješavanje incidenta.
- Objektivna i subjektivna procjena učinkovitosti tima kod rješavanja incidenta.

Sljedeća stvar koju organizacije moraju definirati su razlozi i vremenski period zadržavanja dokaza pojedinog incidenta. Često se radi o vremenskim periodima od nekoliko mjeseci ili godina nakon riješenog incidenta. Faktori koji često presuđuju o ovome su optužba o napadaču, pravne politike i troškovi (uglavnom kapacitetni, poput dodatnih diskovnih spremišta).

7. SANS metodologija

Sljedeća metodologija dolazi iz SANS-a (punog naziva „Escal Institute of Advanced Technologies“), koji glasi kao privatna američka profitna ustanova, gdje je pružanje edukacije o informacijskoj sigurnosti jedna od temeljnih usluga [31]. Metodologija je napisana u preglednom članku 2011. čiji je autor Patrick Kral [32], a sastoji se od šest faza koje navode preporučene korake upravljanja i odziva na incidente, iako naziv članka glasi „Priručnik za upravljanje incidentima“. Uspoređujući okvirno NIST i SANS metodologiju, ne primjećuje se puno razlika, osim u pojmovima i redoslijedu određenih koraka. Na kraju preglednog članka, nalazi se spisak provjere svih koraka opisanih unutar metodologije te predlošci za Windows i Unix operacijske sustave. Oni će unutar ovog dimplomskog rada biti izostavljeni, zbog svoje opsežnosti.

7.1. Faza pripreme

Unutar prve faze, koja se naziva pripremom, potrebno je implementirati i organizirati sljedeće ključne elemente:

- sigurnosne politike
- plan odziva na incidente
- komunikacijski plan
- dokumentacijske predloške
- stručni tim
- kontrole pristupa
- alate (potreban softver)
- edukacije i treninge

SANS-ova metodologija također definira „jump bag“, koji se mora sastojati od:

- Vodiča za upravljanje i odziva na incidente.
- Kontaktne liste svih C(S)IRT timskih suradnika.
- USB uređaja.
- Vanjskog prijenosnog spremnika sa softverom koji može čitati/pisati po datotečnom sustavu računala.
- Prijenosno računalo sa forenzičkim softverom.
- Ostalim alatima za pristup mreži i računalnim komponentama.
- Duplikatori tvrdog diska, kako bi se napravile kopije za istragu.
- Naprtnjača u koju se mogu spremiti sve navedene stvari.

7.2. Faza identifikacije

Druga faza zove se identifikacija, a glavni cilj stručnog tima je odrediti smatra li se incidentom promatrana devijacija određene normalne radnje unutar organizacije. Kako bi se to odredilo, potrebno je prikupiti događaje s raznih izvora, poput datoteka zapisa i poruka pogreške/upozorenja sa uređaja poput IDPS-a, SIEM-a, vatroštita, itd. Kada i ako se odredi jesu li jedan ili nekoliko događaja incident, potrebno je prijaviti nadležnima što prije. Preporučeno je imati barem dvije osobe na jednom incidentu, gdje jedna osoba sakuplja potencijalne dokazne materijale, dok druga utvrđuje jesu li ti materijali dokazi ili ne. Svaki postupak trebao bi se dokumentirati u formatu „tko, što, gdje, zašto i kako“. Nakon što stručni tim odredi opseg incidenta i prikupi/dokumentira sve dokaze, može se krenuti na sljedeću fazu.

7.3. Faza ograničavanja

Sprječavanje daljnjeg utjecaja incidenta glavni je cilj faze ograničavanja, koja se sastoji od nekoliko slijednih koraka. Prvi je kratkotrajno ograničavanje – što prije prekinuti daljnji utjecaj incidenta na organizaciju, npr. izolacija zaraženog dijela od ostatka mreže. Nakon toga potrebno je izvršiti forenziku nad zaraženim dijelom sustava, kako bi se uzeli i sačuvali dokazi. Sljedeće je dugoročno ograničavanje, gdje je potrebno napraviti privremeno rješenje kako bi se zaraženi dijelovi sustava vratili u produkciju. Najčešće, ovaj korak uzima u obzir brisanje zaraženih korisničkih računa i uspostavljenih veza sa napadačevim C2 poslužiteljem. Također, unutar ove faze mogu se instalirati potrebne zakrpe za iskorištene ranjivosti koje su omogućile nastanak incidenta.

7.4. Faza iskorjenjivanja

Tokom faze iskorjenjivanja, stručni tim mora u potpunosti odstraniti utjecaj i tragove incidenta u sustavu, uz očuvanje dokaza u posebnim spremištima. Ponekad je potrebno napraviti cijeli „reimage“ sustava, odnosno očistiti diskovne prostore i radnu memoriju u potpunosti te nanovo instalirati i konfigurirati potreban softver. Nekad će biti dovoljno pokrenuti skeniranje sustava od strane antivirusnog rješenja koje će ponuditi opciju odstranjenja malicioznog koda. Otkriju li se nove ranjivosti, potrebno ih je zakrpati. Svi napravljeni koraci moraju biti dokumentirani, budući da će se prema tome računati ukupni trošak nastalog incidenta.

7.5. Faza oporavka

Faza oporavka obuhvaća procese vraćanja očišćenih sustava natrag u produkciju. Netom prije samog koraka vraćanja, tj. spajanja u mrežu, potrebno je testirati, pratiti i verificirati sustav koji se spaja natrag. Važne odluke potrebne tokom ove faze su:

- Datum i vrijeme oporavka.

- Način testiranja i verificiranja kako bi se utvrdilo jesu li zahvaćeni sustavi potpuno očišćeni od zaraze i spremni za produkciju.
- Trajanje praćenja sustava radi mogućih anomalija.
- Odabir pravih alata za testiranje, praćenje i verificiranje ponašanja sustava.

7.6. Faza evidentiranja naučenog

Zadnja faza, koja obuhvaća evidentiranje naučenog, postoji kako bi se dovršila dokumentacija i zabilježila sva potencijalna poboljšanja koja bi se mogla implementirati u plan upravljanja i odziva na incidente. Glavni cilj je sakupiti iskustvo i saznanja iz nastalog incidenta kako bi se povećala učinkovitost stručnog tima te kako bi se dobili materijali koji mogu poslužiti kao referenca za sljedeću pojavu ovakvog incidenta, ali i kao edukacijski materijal za juniore u stručnom timu. Sastanak, na kojem bi se raspravljao o riješenom incidentu, treba održati unutar najduže 2 tjedna od njegova rješavanja. Činjenice i pitanja koja je potrebno raspraviti na sastanku su:

- Kada je anomalija/problem prvi puta otkrivena i tko ju je otkrio?
- Koliki je opseg zahvaćenosti incidenta?
- Koji su koraci provedeni tokom faza ograničavanja i iskorjenjivanja?
- Što je bilo potrebno napraviti tokom oporavka?
- Mjesta/točke gdje je stručni tim bio učinkoviti.
- Mjesta/točke gdje je potrebno poboljšati korake određene faze ili stručnog tima.

8. Kolekcija vodiča IR Consortium

Kolekcija vodiča koje definira „Incident Response Consortium“ bavi se odzivom na incidente. Ovaj vodič prati NIST metodologiju pristupa odzivu na sigurnosne incidente (u nastavku skraćeno IR), opisanu u NIST-ovom izdanju 800-61. Svaki od slučajeva za jednu fazu životnog ciklusa IR-a ima „rezerviran“ jedan dijagram tijeka, odnosno postoje dijagrami tijeka za pripremu, otkrivanje, analizu, ograničavanje, iskorjenjivanje, oporavak te radnji nakon incidenta, a opisani slučajevi su:

- Djelovanje malicioznog koda (eng. „Malware outbreak“)
- „Phishing“ napad
- Povreda podataka (eng. „Data theft“)
- Djelovanje virusa (eng. „Virus outbreak“)
- Uskraćivanje usluge (eng. „Denial of Service“)
- Nedozvoljeni pristup (eng. „Unauthorized access“)
- Nedozvoljeno mijenjanje privilegija korisnika (eng. „Elevation of privilege“)
- Nedozvoljeni „root“ pristup (eng. „Root access“)
- Nepravilna upotreba (eng. „Improper usage“)

Koraci unutar faza pripreme, oporavka i radnje nakon incidenta su gotovo isti kod svakog slučaja. Kod faza otkrivanja, analize i iskorjenjivanja koraci su većinom slični, no u pravilu se razlikuju kod definiranja indikatora prijetnji i faktora rizika te načina sprečavanja zaraze/napada. Najviše razlika između slučajeva događa se u fazi ograničavanja, gdje se identificiraju uzorci incidenta prema njegovoj prirodi.

8.1. Slučajevi malicioznog koda i djelovanja virusa

U fazi pripreme početka djelovanja malicioznog koda potrebno je odrediti upravitelje ranjivosti, prijetnje i rizika, zatim izvršno vodstvo i ljude potrebne za potporu odzivu (pravni poslovi, javni odnosi, itd.). Potrebno je definirati „eskalacije“, odnosno način traženja dozvola za željene radnje te napraviti dokument za unutarnje i vanjske partnere i upravu. U fazi otkrivanja potrebno je definirati indikatore prijetnji i faktore rizika. Neki od indikatora prijetnji mogu biti iznenadno povećanje korištenja procesora ili neočekivani/neidentificirani promet prema internetu. Neki od faktora rizika mogu biti zahvaćenost klijenata ili negativni utjecaj na javni brand. Nakon definiranja, unutar iste faze potrebno je napraviti zahtjev za pristup uhvaćenom mrežnom prometu, držeći se dokumenata „eskalacija“, a zatim napraviti potrebna skeniranja. U fazi analize, potrebno je redefinirati faktore rizika, nakon što je zadnji korak faze otkrivanja napravljen. Unutar faze ograničavanja, potrebno je identificirati:

- Imovinu (sustave, dijelove IT infrastrukture) koja je zahvaćena incidentom.
- Podatkovnu imovinu koja je oštećena (tajnost, integritet, dostupnost. . .).
- IT usluge koje su zahvaćene incidentom.
- Uzorke koji su doprinijeli u zarazi imovine malicioznim kodom.
- Ranjivosti koje su iskorištene. Koristiti SIEM, „Threat Intelligence“ i ostale izvore podataka (Web mjesta koja sadrže informacije o CVE) kako bi se generiralo izvješće.
- Opseg zaraze malicioznim kodom.
- Alate upotrjebljene tokom ove faze.

Unutar faze iskorjenjivanja potrebno je spriječiti daljnju zarazu. CSIRT se može podijeliti tako da jedan dio pokrene i analizira maliciozan kod u virtualnom okruženju, dok drugi alatima za forenziku pokuša napraviti korelaciju i jasno definirati što je potrebno ograničiti. Ako se koristi antivirusno rješenje, potrebno je zahtijevati zaustavljanje rada malicioznog koda. Ograničenja otkrivena unutar forenzike pretvoriti u SIEM pravila za SOC tim te zahtijevati isključivanje zahvaćenih usluga, konfiguriranje novih pravila vatroštita i ažuriranje/zakrpe sustava. Provesti sastanak sa potrebitima prema dokumentu „eskalacija“. Potrebno je pripaziti na kanal – ako se radi o povjerljivim podacima, instant poruke i mobilna telefonija nisu pouzdan kanal. Na kraju faze, potrebno je (sukladno dogovoru s provedenog sastanka) izbrisati maliciozan kod antivirusnim rješenjem ili nekim drugim mehanizmom. Unutar faze oporavka potrebno je vratiti zahvaćene dijelove sustava u stanje prije zaraze. Neke od čestih radnji za oporavak su korištenje „backup“-a, ponovna instalacija (eng. „Reimage“), ponovna izgradnja (eng. „Rebuild“), prekid privremenog ograničenja stvorenog u fazi ograničavanja. Također, potrebno je i vratiti podatke prije zaraze, npr. preko „Cloud“ sinkronizacije. Na kraju faze potrebno je ponovno skenirati IT infrastrukturu u potrazi za oznakama (eng. „Signature“) incidenta te zakrpati ranjivosti i ažurirati antivirusno rješenje. U zadnjoj fazi, radnje nakon incidenta, potrebno je izvršiti pregled incidenta i iznijeti podatke o šteti. Zatim, potrebno je „primijeniti naučeno“ – ažurirati sigurnosne politike i procese, zahvaćene riješenim incidentom, po potrebi. Na kraju, potrebno je ažurirati plan odziva na incidente i vodič za rješavanje djelovanja malicioznog koda.

Slučaj djelovanja virusa ima veoma slične korake, razlike se većinski pojavljuju u fazama otkrivanja i analize, odnosno u koracima definiranja indikatora prijetnji i faktora rizika. Na primjer, identifikatori prijetnji kod djelovanja virusa mogu biti neobjašnjena rušenja računalnih sustava, instalacije nepoznatih i neodobrenih izvršnih programa, otkrivanje novih i neobičnih zapisa unutar Windows registra, itd. Neki od faktora rizika mogu biti: virus se može iskoristiti za kriminalnu aktivnost, postoji saznanje o ovom virusu unutar i/ili van organizacije, sigurnost zaposlenika i/ili klijenata je ugrožena, itd.

8.2. Phishing slučaj

Kod „Phishing“ napada, faza pripreme jako je slična onoj kod slučaja djelovanja malicioznog koda, uz dodatnu aktivnost – intervju sa žrtvama napada. Kod faze otkrivanja, potrebno je definirati indikatore prijetnji. Aktivnosti vezane uz ovaj korak mogu biti identifikacija lažiranog e-maila, notifikacije zaposlenika koji su dobili sumnjiv e-mail, praćenje organizacijskih web stranica radi dobivanja saznanja o mogućem rudarenju web sadržaja (eng. „web scrapping“). Nakon definiranja indikatora prijetnji, potrebno je kategorizirati incident, zatražiti sakupljanje mrežnog prometa te provesti potrebna skeniranja. Unutar faze analize potrebno je definirati faktore rizika, zatim odrediti metode zakrpe, zatražiti datoteke zapisa i ostale dokazne materijale koji su povezani sa napadom i provesti analize (npr. korelacije događaja zbog otkrivanja pravog konteksta). Unutar faze ograničavanja, potrebno je identificirati:

- Imovinu (sustave, dijelove IT infrastrukture) koja je zahvaćena incidentom.
- Korisničke podatke koji su ugroženi ili su pod rizikom.
- IT usluge koje su zahvaćene incidentom.
- Maliciozne kodove koji su dohvaćeni tokom uspješnog „phishing“ napada.
- Utjecaj napada na poslovanje. Koristiti SIEM, „Threat Intelligence“ i ostale izvore podataka (Web mjesta koja sadrže informacije o CVE) kako bi se generiralo izvješće.
- Opseg zaraze malicioznim kodom.
- Alate upotrjebljene tokom ove faze.

Ostale faze uključuju iste ili slične korake kao i kod slučaja djelovanja malicioznog koda. Koraci se temelje na prikupljenim podacima iz prošlih faza. Naravno, razlikovati će se i načini implementacije ili provedbe tih koraka, kao na primjer korištenje forenzičkih alata koji su bolje prilagođeni trijaži i obdukciji „phishing“ napada.

8.3. Slučaj uskraćivanja usluge

Unutar faze pripreme kod scenarija uskraćivanja usluge, nakon prvog koraka organiziranja tima za incidente, slijedi rasprava o što bržem rješavanju napada (pod pretpostavkom da napad i dalje traje). Faze otkrivanja i analize veoma su slične onima iz scenarija „Phishing“ napada. Tokom faze ograničavanja, potrebno je identificirati:

- Imovinu koja je zahvaćena napadom.
- Imovinu čije su performanse narušene.
- IT usluge koje su zahvaćene napadom.
- Kritične sustave koji su i dalje pod rizikom od (D)DoS napada.

- Vrste paketa koje su korištene za izvršavanje (D)DoS napada.
- „Uska grla“ u mreži koja dodatno pospješuju napad.
- Izvore napada te mogućnost „Blackholing“-a (usmjeravanja prometa u prazno, tj. odbacivanje mrežnih paketa).
- Alate koji su potvrdili postojanje (D)DoS napada (SIEM, IDS, vatroštit, Antivirusno rješenje, itd.).

Faza iskorjenjivanja ima korake slične prošlim dvaju slučajima. Tokom trijaže, potrebno je maknuti dio napadnute mreže od ostatka, a zatim zahtijevati zakrpu ili ponovnu konfiguraciju oštećenog dijela mreže. Kod koraka ublažavanja napada, moguće je napraviti/obnoviti „whitelist“/“blacklist“ izvornih IP adresa i usluga koje moraju biti dozvoljene u mreži, raspraviti sa davateljem internetskih usluga o najboljem rješenju kako se ovaj napad ne bi dogodio u budućnosti, itd. Ostale dvije faze veoma su slične kao i prošla dva slučaja.

8.4. Slučaj povrede podataka

Kod povrede podataka, faze pripreme, otkrivanja i analize ne razlikuju se previše od ostalih slučajeva. U fazi analize, kod definiranja faktora rizika, preporuča se definirati utječu li (i koliko) ukradeni podaci na svakodnevno poslovanje i imidž organizacije, jesu li klijenti zahvaćeni i oštećeni incidentom, zatim koliki je rizik da ukradeni podaci budu objavljeni u javnosti, itd. Tokom faze ograničavanja, potrebno je identificirati:

- Imovinu koja je zahvaćena napadom.
- Korisničke podatke koji su ugroženi ili pod rizikom.
- Metoda korištena kod povrede podataka.
- Sustave koji su sudjelovali u povredi podataka.
- Lateralna kretanja onih korisnika koji su zahvaćeni incidentom.
- Alate koji su potvrdili povredu podataka (SIEM, IDS, vatroštit, Antivirusno rješenje, itd.).

Faza iskorjenjivanja slična je onima iz prijašnjih slučajeva, sa iznimkom gdje je nakon koraka iskorjenjivanja malicioznih radnji potrebno pratiti mrežni promet u slučaju neprekidne povrede podataka (stvaranje posebno konfiguriranih alarma, postavljanje posebnih uređaja za kontrolu, itd.).

8.5. Slučajevi nedozvoljenog pristupa i mijenjanja privilegija

Slučajevi nedozvoljenog pristupa, mijenjanja privilegija korisnika te „root“ pristupa, imaju veoma slične korake kao slučaj djelovanja virusa. Najviše se razlikuje korak definiranja

indikatora prijetnji, od kojih su mogući sljedeći scenariji: zaposlenik se odjednom ne može prijaviti sa svojim korisničkim računom, evidencije korištenja isključenih ili latentnih korisničkih računa, neobjašnjivi e-mailovi sa čudnih korisničkih računa, datoteke zapisa koje sadrže evidenciju o radnji na lokaciji kojoj zaposlenik nema prava pristupa, itd. Slučajevi mijenjanja privilegija korisnika i „root“ pristupa unutar faze iskorjenjivanja imaju dodatni korak instalacije senzora za prikupljanje mrežnog prometa, kako bi se učinkovitije pratili događaji mijenjanja prava kroz mrežu. Također, faza ograničavanja sadrži korake identifikacije dijelova sustava koji su „zaključani“ zbog nekoliko krivih pokušaja upisa korisničkih podataka i identifikacije neautoriziranih alata koji su korišteni tokom pokušaja upada u neovlašteni dio sustava ili tuđe korisničke račune.

8.6. Slučaj nepravilne upotrebe

Unutar koraka definiranja indikatora prijetnji kod slučaja nepravilne upotrebe, moguće je sljedeće: prekomjerna količina prometa pregledavanja ili preuzimanja s weba, identifikacija velikog broja e-maila poslana ili primljena od strane korisnika, notifikacija od strane drugih zaposlenika/partnera/dobavljača o sumnjivoj radnji, neobjašnjen maliciozni kod ili zaraza virusom na računalnom sustavu, korištenje nedozvoljenih komunikacijskih metoda ili mrežnih protokola te alarmi o posjećivanju nedozvoljenih web stranica. Korak definiranja faktora rizika u slučaju nepravilne upotrebe kombinacija je istog koraka kod ostalih slučajeva, jer je „nepravilna upotreba“ širok pojam koji može poprimiti prirodu incidenta bilo kojeg već navedenog slučaja, uz činjenicu da je incident ovog puta vrlo vjerojatno nastao slučajno. Tokom faze ograničavanja, potrebno je identificirati:

- Imovinu koja je zahvaćena napadom.
- Korisničke podatke koji su ugroženi ili su pod rizikom.
- Maliciozne kodove unutar sustava zahvaćenog incidentom.
- Mrežne protokole koji su korišteni tokom nepravilne upotrebe.
- Sredstva i metode koje je koristio osumnjičeni zaposlenik/vanjska osoba.
- Zaposlenike/partnere/klijente koji su zaprimili bilo kakav oblik podataka od osumnjičenog.
- Lokaciju gdje se nalaze preostali podaci povezani s prekršajem.
- Sve sumnjive datoteke, softver ili podatke povezane s prekršajem.
- Alate koji su potvrdili nepravilnu upotrebu, tj. prekršaj (SIEM, IDS, vatroštit, Antivirusno rješenje, itd.).

Unutar faze iskorjenjivanja, tokom koraka trijaže i potvrde izvješća o incidentu, potrebno je napraviti zahtjev za ažuriranje mrežne konfiguracije te ažuriranje politika za krajnje točke u

mreži. Nakon toga slijedi korak razgovora sa zaposlenikom (osumnjičenim) i odjelom za ljudske resurse. Zadnji korak faze, tj. suzbijanje prijetnje, sastoji se od aktivnosti: upozoriti zaposlenika o njegovom prekršaju (po potrebi i dogovoru pružati „Security Awareness“), vratiti ili odstraniti prekršajem zahvaćeni dio sustava/mreže te izvršiti forenziku nad podacima. Ostale faze, koje nisu navedene, slične su fazama ostalih slučajeva.

9. Kolekcija vodiča CERT Societe Generale IRM

Druga kolekcija vodiča koja opisuje upravljanje sigurnosnim incidentima naziva se CERT IRM, a dolazi iz CERT Societe Generale grupe. „M“ u nazivu kolekcije predstavlja „Methodologies“, no usprkos tome, CERT IRM je klasificiran kao vodič najboljih praksi upravljanja incidentom [33]. Iako strukturno slična vodičima iz IRC-a, ova kolekcija temelji se na priručniku o upravljanju i odzivu na incidente kojeg je izdao SANS 2011. godine. CERT IRM sastoji se od 16 dokumenata koji predstavljaju vrste incidenta, odnosno moguće slučajeve. Svaki dokument sastoji se od dvije stranice na kojima se nalazi šest faza, čiji koraci su prezentirani u natuknicama:

- priprema – početak upravljanja incidentom
- identifikacija – koraci otkrivanja incidenta
- ograničavanje – upravljanje utjecajem incidenta na organizaciju
- sanacija – odstranjivanje prijetnji
- oporavak – koraci oporavka organizacije u stanje prije incidenta
- posljedice – načini kako povećati učinkovitost upravljanja incidentom u budućnosti

Slučajevi opisani unutar vodiča su:

- infekcija mreže crvom
- neovlašten upad u Windows OS
- neovlašten upad u Linux OS
- uskraćivanje usluge
- maliciozno ponašanje na mreži
- napad na web stranicu
- maliciozan kod u Windows OS
- ucjenjivanje
- maliciozan kod na pametnom telefonu
- socijalni inženjering
- curenje podataka
- zloupotreba radnog mjesta
- „Phishing“
- prijevara

- kršenje zaštitnog znaka
- „Ransomware“

Budući da postoji velik broj slučajeva, opisati će se vodiči za samo nekoliko slučajeva.

9.1. Slučaj napada na web stranicu

Unutar faze pripreme kod slučaja napada na web stranicu, potrebno je imati ažurirane sheme komponenata web aplikacije koje su povezane s web poslužiteljem. Nakon toga potrebno je pustiti u produkciju „backup“ web stranicu na koju se može objaviti vjerodostojan sadržaj za klijente, ali onda te iste klijente obavijestiti o lokaciji „backup“ web stranice. Nakon toga potrebno je što brže postaviti alate za praćenje kako bi se pronašla bilo kakva abnormalna ponašanja na web stranicama koja uvelike utječu na razvoj poslovanja. Također, poželjno je napraviti izvoz datoteka zapisa web poslužitelja na vanjsko spremište (potrebno je napraviti vremensku sinkronizaciju). Tu se još nalaze četiri koraka, koji govore o referenciranju vanjskih sadržaja i davatelja usluga „hosting“-a.

Korisni kanali kod otkrivanja uzorka incidenta su napadnute web stranice (njihov sadržaj koji može unutar sebe imati maliciozni kod), korisnici i sigurnosne provjere sa alatima poput Google SafeBrowsing. Glavni cilj ove faze je potvrditi ilegalne radnje na vlastitoj web stranici i otkriti izvor problema. Zato su tu koraci poput provjere statičkog sadržaja, hiperveza i ostalih referenci, zapisa datoteka te baza podataka u potrazi za malicioznim sadržajem.

Faza ograničavanja sadrži korake preslike podataka sa web poslužitelja za potrebe forenzike i prikupljanja dokaznih materijala. Potrebno je uvjeriti se da ranjivost iskorištena kod napadnute web stranice ne može biti iskorištena negdje drugdje unutar IT infrastrukture. Sljedeći korak je pronaći vektor/način na koji je napadač ušao u sustav te onemogućiti ponovni upad sanirajući taj propust. Ako je potrebno, podignuti i privremeni web poslužitelj sa vjerodostojnim web stranicama i aplikacijama.

Faza sanacije je jasna – odstraniti sav sadržaj koji je lažiran. Kod faze oporavka, odrediti je li potrebno mijenjati korisničke podatke korisnicima (ako su ukradeni putem lažirane web stranice). Ako je backup poslužitelj korišten u produkciji, potrebno ga je povući natrag i postaviti primarni poslužitelj.

Tokom posljedične faze, koja služi i za „prihvaćanje naučenog“ tokom upravljanja incidentom, potrebno je obavijestiti javnost o incidentu ili određenu zajednicu koja je imala pristup napadnutoj web stranici. Naravno, potrebno je na kraju izraditi izvještaj sa sljedećim elementima:

- vrijeme inicijalnog otkrića incidenta
- odrađene akcije i vremenska crta
- što je pošlo po zlu

- trošak koji je uzrokovao incident

U slučaju spoznaje ranjivosti, potrebno je obavijestiti autora programskog koda (ili softvera) unutar kojega je nađena ranjivost.

9.2. Slučaj zloupotrebe radnog mjesta

Kod faze pripreme na incident u slučaju zloupotrebe radnog mjesta, potrebno je osigurati pouzdan kontakt sa određenim zaposlenicima iz odnosa s javnošću, ljudskih resursa i odjela za pravne poslove. Sljedeće, potrebno je centralizirati način sakupljanja datoteka zapisa (SIEM), zatim osigurati kvalitetan način autorizacije i organiziranja korisničkih računa. Unutar faze identifikacije nalazi se napomena koja upozorava na činjenicu da se ovakav incident jako teško otkriva i ne postoje koraci koji garantiraju uspjeh. Ipak, koraci identifikacije dijele se na tehničku i ljudsku. Tehnička identifikacija uključuje alarme iz SIEM-a i/ili IDPS sustava. Ljudsku identifikaciju čine uprava, GRC odjel (eng.„Governance, risk management, and compliance“), kolege i vanjski članovi organizacije. Faza ograničavanja incidenta za sljedeće korake navodi kako je za daljnje izvršavanje potrebno zatražiti odobrenje CISO-a ili druge odgovorne osobe sa istim pravima. Koraci su: uključiti ljude u asistenciju otklanjanja incidenta i pronalaska krivca. Kod potencijalnog krivca, potrebno je dogovoriti sastanak sa njim i osobom za ljudske resurse, kako bi mu se objasnilo što se događa te kako bi potencijalni krivac dao svoj iskaz. Preporuča se potencijalnom krivcu smanjiti privilegije, a po potrebi i onesposobiti korisnički račun ili bilo koji drugi oblik autorizacije, udaljeni pristup, pa čak i zapljena poslovnih uređaja. Ako nakon ovih koraka (nakon određenog vremena) nisu pronađene maliciozne radnje, može se početi sa forenzikom nad poslovnim uređajima potencijalnog krivca. Ako je maliciozna radnja potvrđena, potrebno je posavjetovati se sa odjelom za pravne poslove i pružiti im dovoljno dokaza koji potvrđuju identitet krivca. Faza sanacije je ograničena kod ovog slučaja, jer uvelike ovisi o odjelu za pravne poslove. Faza oporavka obuhvaća vraćanje ograničenih sustava na normalni način rada. Potrebno je obavijesti svakoga tko je zahvaćen incidentom te po potrebi održati „Security Awareness“ koji će pokrivati ovakve slučajeve. Posljedična faza ista je kao i kod slučaja napada na web stranicu.

9.3. „Ransomware“ kao slučaj

Kako bi se pripremili na incident koji uključuje „Ransomware“, potrebno je jako dobro poznavati sigurnosne politike organizacije koje se tiču operacijskih sustava i korisničkih profila, zatim sigurnosne proizvode koji su postavljeni unutar mreže radi otkrivanja i sprečavanja anomalija. Budući da je incident najčešće otkriven od strane samih zaposlenika koji postaju žrtve, potrebno je održati „Security Awareness“ o ovakvoj vrsti prijetnje. Najvažniji korak je imati kvalitetnu politiku izrade „backup“-a podataka lokalnih i mrežnih korisnika.

Postoji nekoliko čimbenika koji tokom faze identifikacije mogu odati „Ransomware“ prijetnju, a to su čudni poslovni e-mailovi sa priloženim datotekama, poruke ucjene poput „uplatiti (kripto)novac za dekripciju podataka“, zaposlenici prigovaraju kako im je pristup određenim

datotekama odbijen te sumnjiv broj datoteka koje se mijenjaju unutar kratkog vremenskog perioda. S druge strane, faza identifikacije sastoji se od računalne i mrežne identifikacije. Računalna identifikacija sastoji se od:

- Pretrage čudnih izvršnih datoteka unutar korisničkih profila, npr. unutar
- Pretrage čudnih nastavka datoteka poput .abc .xyz .aaa ili bilo kakvih poruka ucjene.
- Uzimanja preslike radne memorije.
- Pretrage čudnih procesa.
- Pretrage čudnih uzoraka unutar predložaka e-maila.
- Pretrage čudnih aktivnosti vezanih uz web pretrage i mrežu općenito (npr. povezivanje na Tor čvorove).

Mrežna identifikacija sastoji se od pretraga:

- Uzoraka povezivanja na alate za eksploataciju.
- Povezivanja na napadačev C2 poslužitelj (poslužitelj sa kojim „Ransonware“ komunicira tokom zaraze).
- Čudnih aktivnosti vezanih uz web pretrage i mrežu općenito (npr. povezivanje na Tor čvorove).
- Čudnih uzoraka unutar predložaka e-maila.

Faza ograničavanja čini korake isključivanja svih zaraženih računala iz mreže te blokiranje prometa sa C2 poslužitelja uz slanje ostalih identifikatora (npr. URL unutar poruke ucjene) pružatelju usluga sigurnosti. Sanacija čini aktivnost brisanja svih binarnih datoteka i zapisa iz registra koji su u korelaciji sa zaraženim korisničkim profilima. Ako ovaj postupak nije izvediv, potrebno je napraviti ponovnu instalaciju operacijskog sustava uz prethodno čišćenje diskovnog prostora. Koraci unutar faze oporavka moraju se izvršavati jedan za drugim uz dogovoreno praćenje. Ti koraci su:

- Ažuriranje antivirusnog rješenja sa potpisima malicioznih kodova koji predstavljaju „Ransomware“.
- Provjera prisustva „Ransomware“-a u sustavu, prije ponovnog povezivanja zaraženih računala (sada već očišćenih) u mrežu.
- Uvjeriti se u normalno stanje mrežnog prometa.
- Vratiti iz „backup“-a korisničke podatke.

Posljedična faza je ista kao i kod prošla dva slučaja.

10. TheHive + Cortex

TheHive je platforma sa grafičkim korisničkim web sučeljem napravljena za odziv i upravljanje sigurnosnim incidentima. Otvorenog je programskog koda pod licencom „GNU Affero General Public License v3.0“, a platformu je razvila neprofitna organizacija TheHive-Project. Platforma pomaže svim SOC, CSIRT, CERT i ostalim IT timovima koji su zaduženi za analizu i sigurnost sustava, upravo zato jer objedinjuje i omogućuje upravljanje incidentima, automatiziranu analizu, upravljanje alarmima, prikaz statistike i vizualizacije, ali i djeluje kao „društvena mreža“ za odziv i upravljanje incidentima.

Cortex je također proizvod „otvorenog koda“ TheHive-Project organizacije, a napravljen je s namjerom da TheHive platformu obogati sa jednostavnim načinom analize indikatora kompromisa poput datoteka, domena, URL-ova, kriptografskih sažetaka, itd. Može se koristiti i zasebno – postoji web sučelje samo za funkcionalnosti koje pruža Cortex, a ono se sastoji od podjele i upravljanja analitičarima, analizatorima te samim analizama.

Spomenute funkcionalnosti, kao i njihov položaj te korištenje samih alata, biti će opisani u sljedećim poglavljima.

10.1. TheHive funkcionalnosti

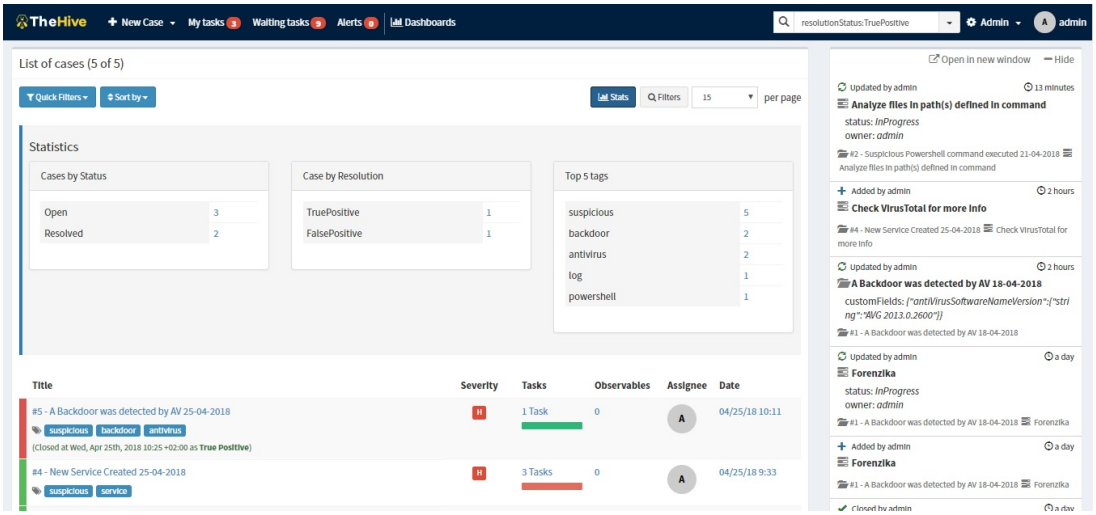
Unutar ovog poglavlja slijedi opis svih funkcionalnosti koje je autor prepoznao unutar rada TheHive-a (prikazane će biti autorove slike, testni podaci, položaj kontrola za obavljanje raznih funkcionalnosti...) te uz pomoć službene dokumentacije sa GitHub stranice TheHive platforme na adresi: https://github.com/TheHive-Project/TheHiveDocs

10.1.1. Upravljanje incidentima - slučajevi

Upravljanje incidentima: stvaranje, opisivanje i zatvaranje slučajeva (eng. „case“) koji identificiraju potencijalne i stvarne sigurnosne incidente. Stvaranje i obavljanje zadataka (eng. „task“) koji opisuju korake odziva na sigurnosni incident. Stvaranje i vođenje evidencije zadataka (eng. „task logs“), koji detaljno opisuju rad osobe koja je zadužena za određeni zadatak unutar slučaja.

Lista slučajeva (slika 4) početni je prozor nakon prijave u TheHive web sučelje. Slučajevi se prikazuju vremenskim redoslijedom te je moguće dodatno prikazati sažetu statistiku (top 5 oznaka, zbroj otvorenih i zatvorenih slučajeva, itd.) i filtrirati prema složenim kriterijima: ključne riječi, status, oznake, korisnici koji su napravili slučaj, ozbiljnost slučaja, naziv te vremenski interval za traženje slučajeva otvorenih/stvorenih unutar određenog vremena. Za svaki slučaj prikazan je naziv slučaja i njegove oznake te dodane informacije o statusu slučaja (stupac „Title“), ozbiljnosti slučaja (eng. „Severity), koliko zadataka postoji te koji je njihov status (stupac „Tasks“), koliko je artefakata pridijeljeno slučaju (eng. „Observables“), korisnik koji je stvorio/otvorio slučaj (eng. „Assignee“) te datum stvaranja/otvaranja slučaja (eng. „Date“). Ovisno o

području klika, sa ovog prozora korisnika se preusmjerava na konkretni zadatak, slučaj uz koji je zadatak vezan ili se pokreće pretraga uz brzi filter.

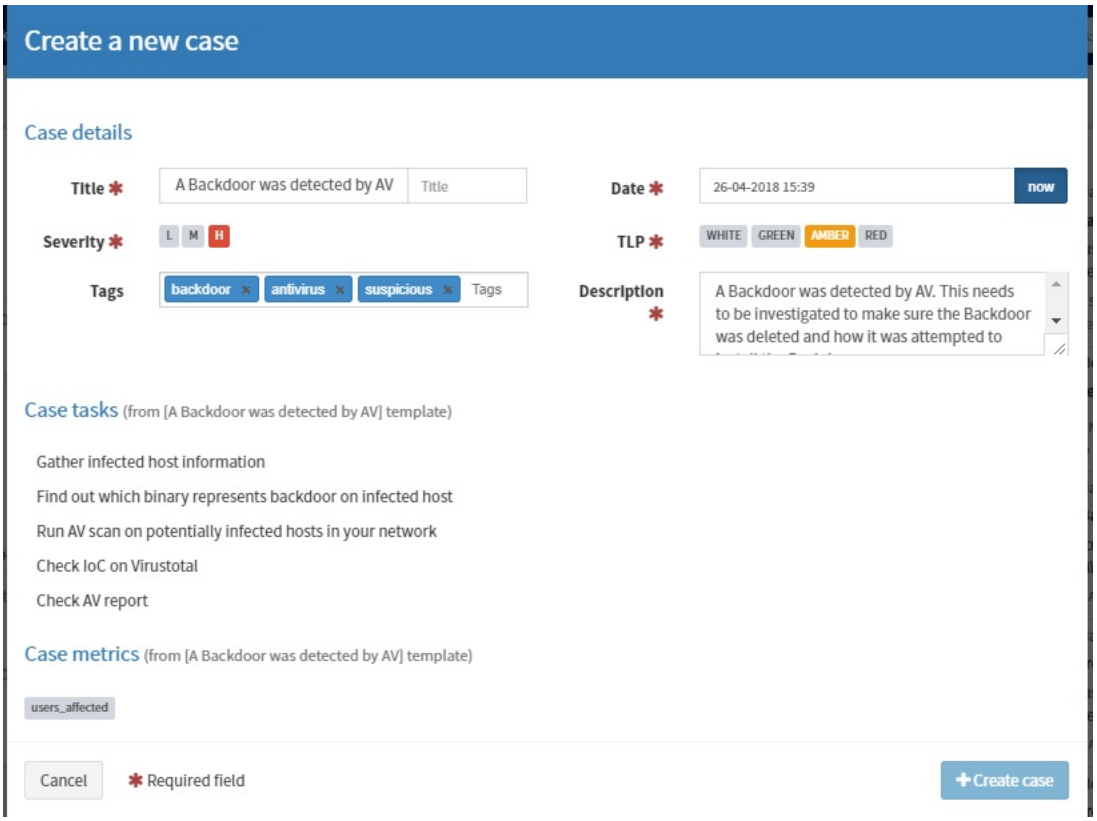


Slika 4: Lista slučajeva (autorski rad)

Novi slučaj stvara se klikom na opciju "New Case" unutar gornje alatne trake. Za slučaj je potrebno definirati detalje slučaja (eng. „Case details“), dok su zadaci (eng. „Case tasks“) i metrike (eng. „Case metrics“) opcionalni. Kod detalja slučaja definira se:

- „Title“: Naziv slučaja.
- „Severity“: Ozbiljnost slučaja. L = niska ozbiljnost (najmanji prioritet), M = srednje velika ozbiljnost (srednje veliki prioritet), H = velika ozbiljnost (najveći prioritet).
- „Tags“: Oznake po kojima se slučaj identificira. Također dopušta kategoriziranje unutar statistika te brzo pretraživanje.
- „Date“: Vrijeme stvaranja slučaja.
- „TLP“: Punim engleskim nazivom „Traffic light protocol“. Dokumentacija koja vrlo dobro opisuje ovaj protokol nalazi se na adresi: https://www.us-cert.gov/tlp Sažeto: WHITE = slobodna distribucija informacija o slučaju u javnosti, GREEN = distribucija informacija unutar cijele zajednice kojoj pripada organizacija, AMBER = distribucija informacija samo unutar organizacije, RED = distribucija samo među određenim ljudima.
- „Description“: Tekst koji pobliže opisuje slučaj Ako se stvara prazan slučaj, tada se mogu dodavati zadaci (samo njihovo ime), a dok se stvara slučaj iz predloška, tada se ne mogu stvarati zadaci u ovom prozoru, nego tek kasnije kada je slučaj stvoren.

Ako se stvara prazan slučaj, tada se mogu dodavati zadaci (samo njihovo ime), a dok se stvara slučaj iz predloška (slika 5), tada se ne mogu stvarati zadaci u ovom prozoru, nego tek kasnije kada je slučaj stvoren.



Slika 5: Stvaranje novog slučaja uz pomoć predloška (autorski rad)

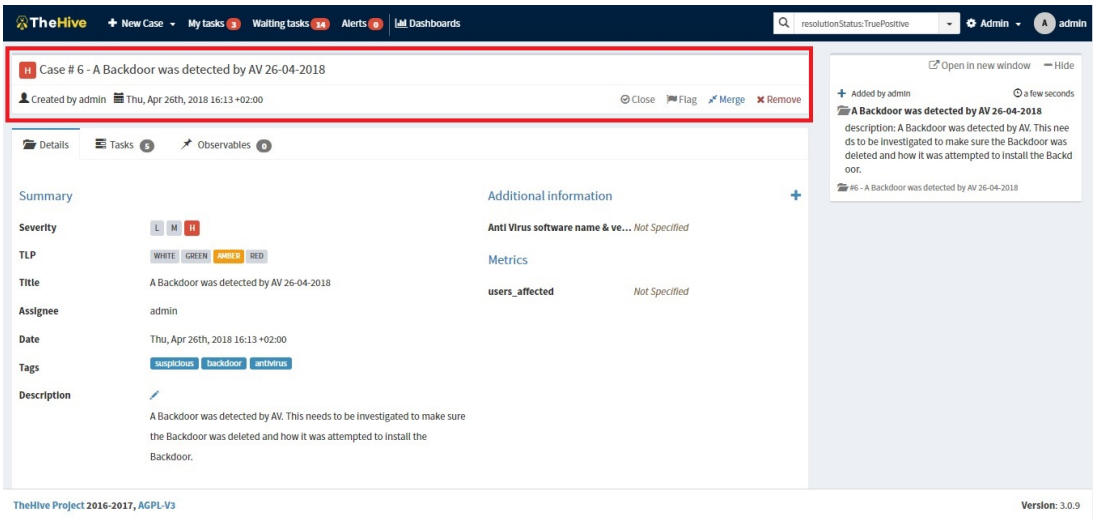
Kada je slučaj stvoren, korisnika se prebacuje na taj slučaj, gdje je prozor podjeljen na dva dijela (slika 6): gornji dio koji je uvijek prisutan i tri kartice (sekcije prozora): detalji, zadaci i artefakti.

Spomenuti gornji dio sastoji se od ozbiljnosti slučaja, naziva, naziva korisnika koji ga je stvorio, datumom kad je slučaj stvoren i sljedeće četiri opcije:

- „Close“: Zatvori slučaj, ako je riješen (ili iz bilo kojeg drugog razloga). Ako nisu svi zadaci završeni, prvo se korisnika pita želi li stvarno zatvoriti slučaj iako nisu svi zadaci riješeni. U slučaju da korisnik želi zatvoriti takav slučaj, svi zadaci koji nisu završeni se brišu, te se prelazi na sljedeći prozor. Taj prozor prikazuje se i kod zatvaranja slučaja u kojemu su svi zadaci već riješeni. Unutar prozora potrebno je definirati status zatvaranja i završni opis. Kao status zatvaranja moguće je odabrati „True Positive“ (incident je ozbiljan i istraživanje je pokazalo maliciozno prisustvo), „False Positive“ (za incident se smatralo da pokazuje nešto maliciozno, ali je istraživanje pokazalo drugačije), „Indeterminate“ (nije još poznato postoji li nešto maliciozno unutar incidenta ili ne) i „Other“ (slučaj nije incident, kao što se prvotno mislilo).
- „Flag“: Označi zastavicom. Zastavica određuje prioritet, tj. ako je nešto označeno zastavicom, tada je to visokog prioriteta (prije će se pokazati na listi slučajeva, moći će se brže pretražiti...). Pritiskom na „Flag“ gumb on se pretvara u „Unflag“ te je tako moguće maknuti zastavicu i vratiti slučaj u „normalni“ prioritet.
- „Merge“: Spoji dva postojeća slučaja. Postoji scenarij u kojem se mogu stvoriti dva

slučaja koja pokazuju na isti incident, s time da takva dva slučaja ne moraju biti ista (prema predlošku slučaja). Ako se to dogodi, tada se spajaju naziv, oznake i opis, dok TLP i ozbiljnost budu jednaka slučaju za koji se pokrenuo postupak spajanja, a ne onaj s kojim se spaja. Također, spajaju se svi zadaci i njihovi statusi (završen, otvoren, označen zastavicom, . . .).

- „Remove“: Ovaj gumb je jedino vidljiv ako još nikakva akcija nije napravljena nad slučajem (tj. ako je u povijesti slučaja samo zapisano njegovo stvaranje). „Remove“ nepovratno briše slučaj iz TheHive.



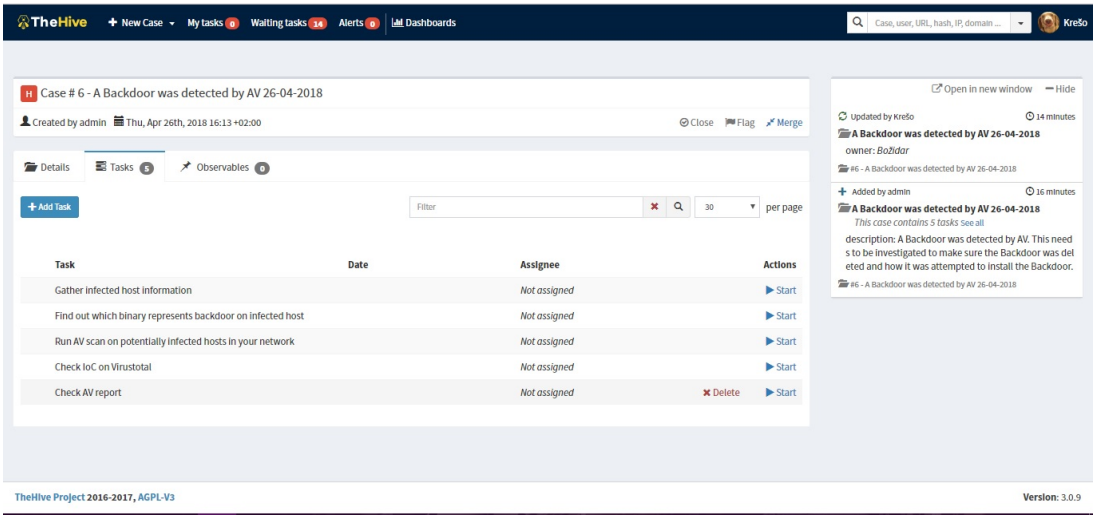
Slika 6: Prozor sa konkretnim slučajem. Crvenim obrubom označen gornji dio (autorski rad)

Kartica detalji (trenutna kartica sa slike 6) sastoji se od sažetka (eng. „Summary“) i dodatnih informacija. Korisnik sa "write" ulogom može mijenjati obje komponente u bilo kojem trenutku.

10.1.2. Upravljanje incidentima - zadaci i evidencije

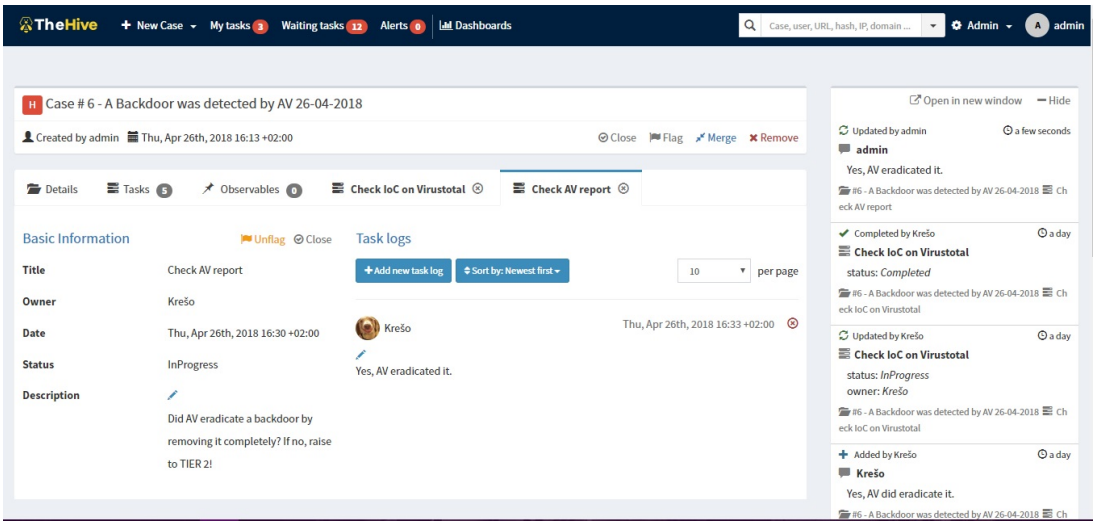
Unutar kartice „Tasks“ nalaze se svi zadaci koji su stvoreni tokom stvaranja slučaja (slika 7). Također, klikom na plavi gumb „Add tasks“ mogu se dodati novi zadaci ako je potrebno. Kada se korisnik želi angažirati za neki zadatak, potrebno je kliknuti na „Start“. Ako korisnik želi obrisati zadatak (ne završiti, nego baš maknuti sa slučaja), tada mora kliknuti na „Delete“.

Kada se korisnik angažira za neki zadatak, prebacuje se u novu sekciju pod imenom tog zadatka (slika 8). Unutar te sekcije može uređivati osnovne podatke o zadatku (sve osim „Status“-a, on se sam ažurira otvaranjem i zatvaranjem zadatka) te može upisivati nove i uređivati/brisati postojeće evidencije (eng. „Task logs“). Klikom na plavi gumb „Add new task log“ prikazuje se blok za pisanje sa nekoliko opcija referenciranja programskog koda, slika, poveznica, itd. Također moguće je staviti privitke uz određenu evidenciju. Jedan zadatak može imati neograničeni broj evidencija. Kao i slučaj, pojedini zadatak može se označiti zastavicom pritiskom na „Flag“, odnosno maknuti zastavicu pritiskom na „Unflag“. Ovom radnjom izražava se prioritet (slično svojevrsnoj oznaci „hot“ na raznim informacijskim portalima), te se može



Slika 7: Lista zadataka konkretnog slučaja (autorski rad)

unutar pametne tražilice pretraživati sve što je označeno zastavicama. Također, unutar određenih prozora, svaki element označen zastavicom će biti ili na vrhu liste ili samo označen žutom bojom.



Slika 8: Zadatak, označen zastavicom, sa jednim evidencijskim zapisom (autorski rad)

Pokraj gumba „Flag“, nalazi se i gumb „Close“ koji označava zadatak završenim. Na sekciji sa svim zadacima, može se svaki završeni zadatak ponovo otvoriti (gumb „Reopen“) ako se smatra da on još nije završen. Napomena: nije potrebno ponovno otvarati zadatak samo da bi se dodala nova ili uređivala stara evidencija. Potrebno je dogovoriti kontekst kada je potrebno, a kada nije, ponovno otvaranje zadatka, jer se ponovnim otvaranjem briše datum kada je zadatak prvotno završen.

10.1.3. Automatizirana analiza

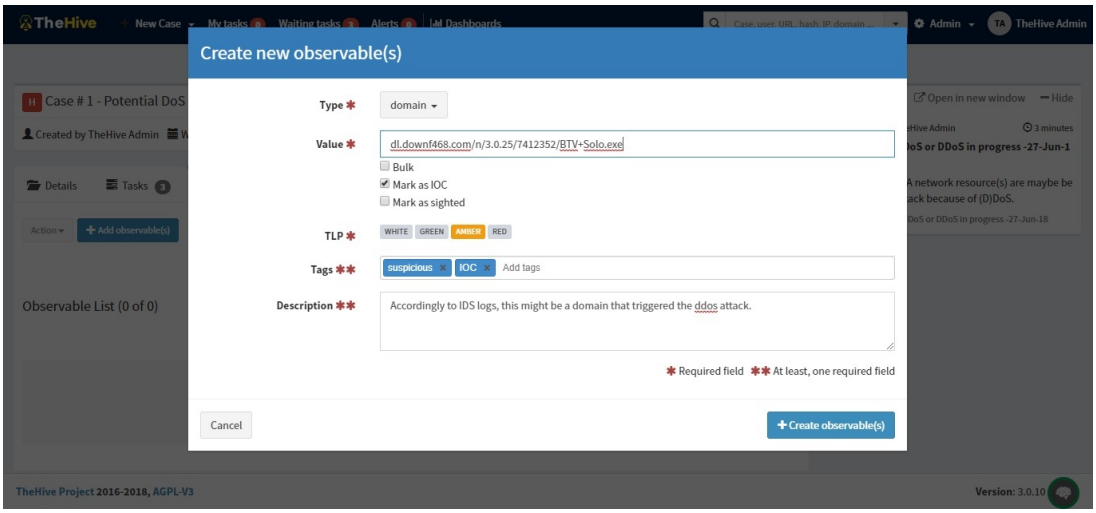
Kao podrška timu za odziv na incidente, napravljena je dodatna usluga TheHive platformi, naziva Cortex, a tiče se pružanja zajedničkog sučelja za analizu artefakata (eng. „artifacts“) poput datoteka, e-maila, IP adresa, domena, sažetaka, itd. Cortex omogućuje, putem komunikacije sa API-jem, korištenje raznih web usluga, kao što su: VirusTotal, Virusshare, PassiveTotal, Nessus, Shodan, Cuckoo Sandbox i mnogi drugi. Artefakti se mogu dodijeliti na razini slučaja. Napomena: funkcionalnost automatizirane analize koju pruža Cortex nije sastavni dio TheHive platforme, odnosno TheHive može raditi samostalno i bez ove funkcionalnosti.

Unutar kartice „Observables“ nalaze se svi artefakti vezani uz slučaj. Artefakti se dodavaju pritiskom na gumb „Add observable(s)“ (slika 9). Svaki artefakt može imati sljedeća svojstva:

- „Type“: Vrsta artefakta. Ovisno o njoj, TheHive zna koje analizatore smije koristiti, tj. ponuditi korisniku za analizu. Neke od vrsta: domena (eng. „Domain“), url, sažetak (eng. „Hash“), datoteka (eng.„File“), itd.
- „Value“: Vrijednost artefakta. Mora biti vezana uz vrstu, inače će analize biti netočne.
- „Bulk“: Opcionalna kučica koja označuje je li unesena jedna vrijednost (količinski) ili više njih.
- „Mark as IOC“: Opcionalna kučica koja označuje predstavlja li unesen artefakt razlog nastajanja sigurnosnog događaja ili nanesene štete. Artefakt kasnije u listi dobiva oznaku zvjezdice."
- „Mark as sighted“: Opcionalna kučica koja označuje artefakt kao tražen i nađen unutar datoteka zapisa koje opisuju incident. Na primjer, pronađe li neki analizator dvije maliciozne adrese unutar e-mail poruke (koja je prenesena na TheHive kao datoteka), osoba koja rukuje incidentom može prenjeti te dvije adrese kao dodatna dva artefakta i potražiti ih u SIEM-u ili zapisima IDpS alata. „Mark as sighted“ će biti onaj artefakt koji je pronađen unutar navedenih lokacija.
- „TLP“: „Traffic light protocol“ (opisan u poglavlju 10.1.1.)
- „Tags“: Oznake prema kojima se artefakt lakše identificira, a kasnije i pretražuje.
- „Description“: Proizvoljni opis.

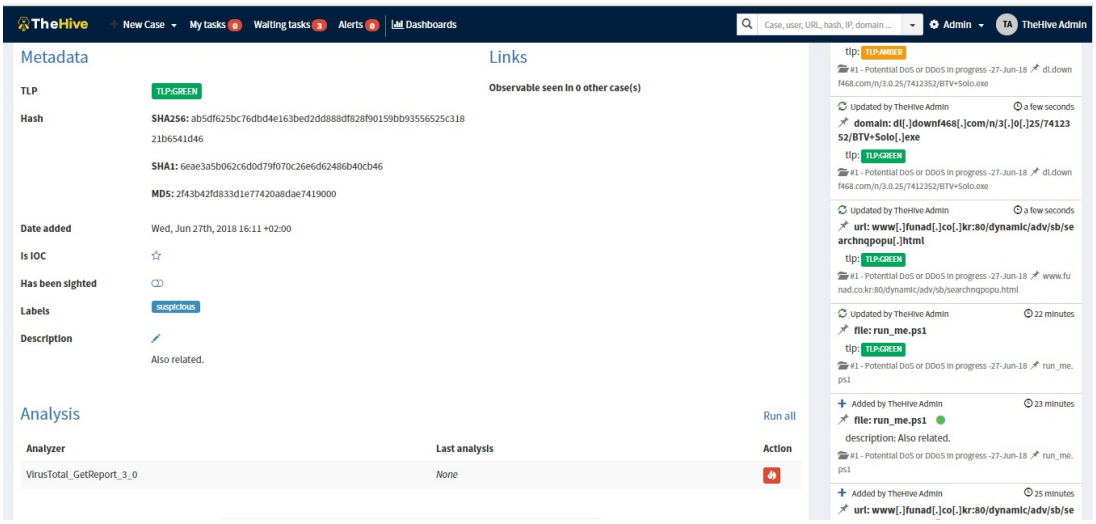
Nakon što je artefakt napravljen, moguće ga je izvesti (unutar csv, txt ili misp datoteke), obrisati, promjeniti oznake, TLP i opcionalne kučice (gore navedene) te pokrenuti dozvoljene analizatore. Potrebno je uzeti u obzir da, ovisno o analizatorima, analize mogu vremenski potrajati. Klikom na naziv pojedinog artefakta, otvara se kartica sa tri sekcije (slika 10):

- „Metadata“: Svi podaci o artefaktu koji su definirani tokom njegova stvaranja (uz datum stvaranja) te dodatnih polja poput sažetaka datoteka. Dozvoljeno ih je uređivati (iznimka su saželi datoteka).



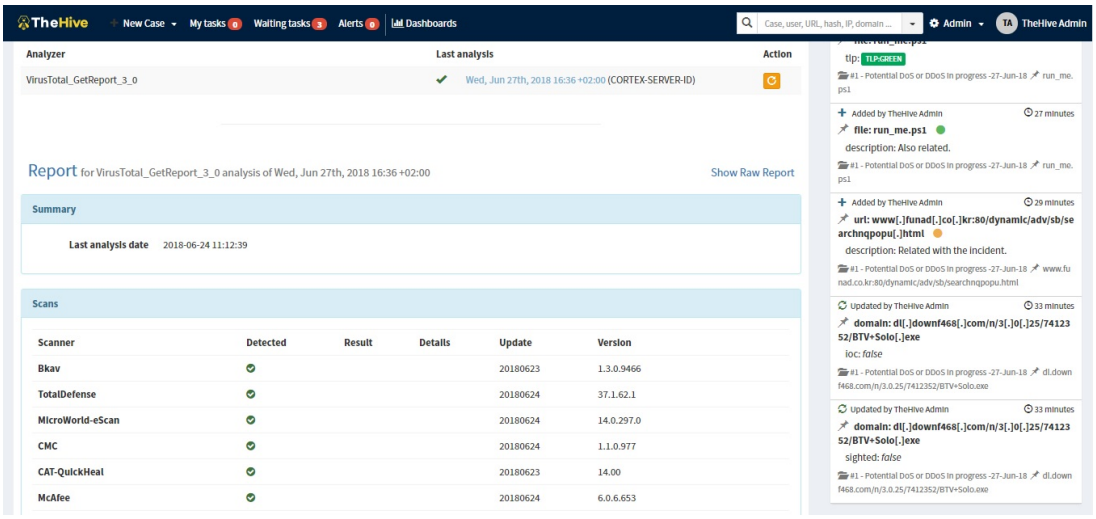
Slika 9: Prozor za stvaranje novog artefakta (autorski rad)

- „Links“: Pregled svih onih slučaja u kojima se nalazi trenutni artefakt.
- „Analysis“: Pregled svih omogućenih analizatora, datuma zadnjih analiza (uz status i poveznicu na generirano izvješće) te opcija (ponovnog) pokretanja analizatora.



Slika 10: Kartica konkretnog artefakta (autorski rad)

Iako se analizatori konfiguriraju na instanci Cortex poslužitelja, potrebno je unutar konfiguracije TheHive instance ispravno definirati IP adresu Cortex poslužitelja, zatim API ključ za pristup TheHive-a prema Cortex-u te uključiti Cortex modul. Kako bi se generirao izvještaj, korisnici sa Admin pravima moraju definirati predloške izvještaja. TheHive-Project pruža svoje službene predloške na GitHub stranici Cortex projekta. Na slici 11 pokazan je primjer izvještaja koji je generiran obavljenom analizom (usluga VirusTotal) nad datotekom prenesenom kao artefakt u TheHive.



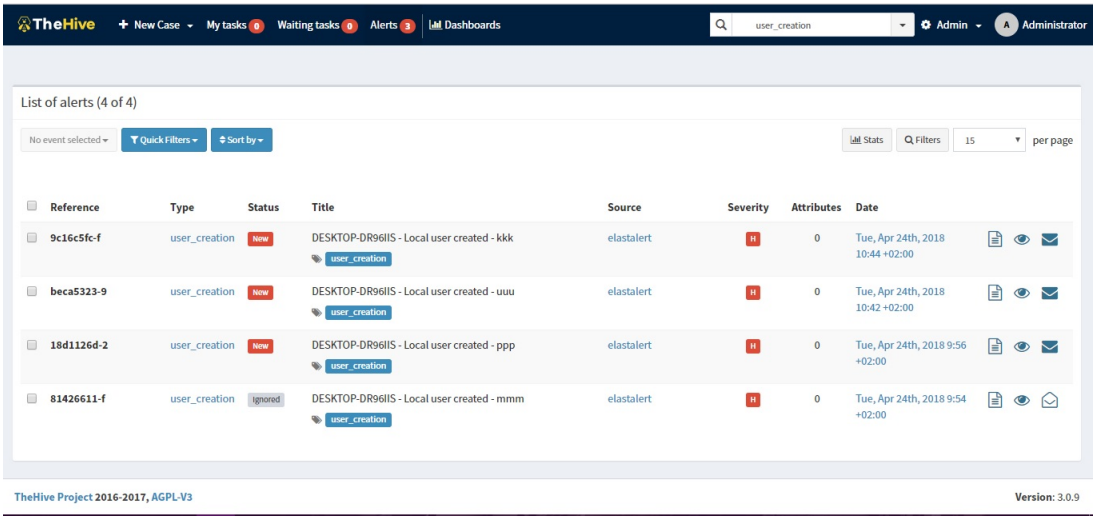
Slika 11: Primjer generiranog izvještaja kao odgovor vraćen od VirusTotal usluge za prenesenu datoteku. (autorski rad)

10.1.4. Upravljanje alarmima

Stvaranje, praćenje i konverzija alarma u slučajeve. Alarme stvaraju platforme za javno dijeljenje sigurnosnih prijetnji (eng. „Threat sharing platform“), platforme za upravljanje informacijama i događajima vezanim uz sigurnost (eng. „SIEM platform“), itd. Alarmi su onaj element TheHive-a na koje platforma „sluša“ od strane drugih usluga, koje se ne moraju nužno nalaziti na istoj infrastrukturi poput one na kojoj je instalirana TheHive platforma. Ako osoba zadužena za odziv na incidente smatra da je alarm potencijalni incident, može ga pretvoriti u slučaj, uz dopunjavanje potrebnih polja za slučaj.

TheHive vodi evidenciju o pristiglim alarmima u obliku liste (slika 12) Za svaki alarm zapisano je sljedeće:

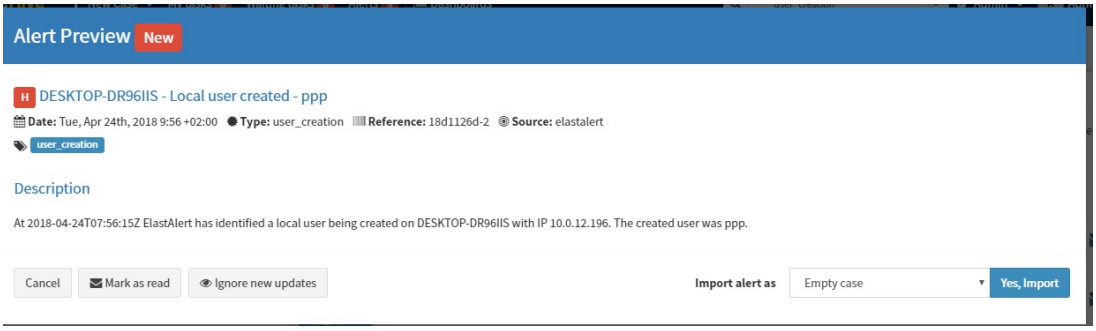
- „Reference“: Jedinstvena oznaka (dio sažetka) koja identificira alarm. Služi kod provjere duplikata i kod ažuriranja postojećih alarma od strane istog izvora alarma.
- „Type“: Polje koje je definirano od strane izvora alarma. Slično je oznakama (eng. „tags“) kod stvaranja slučaja.
- „Status“: Označava je li alarm nov (nepročitan), star (pročitan) ili se više ne ažurira („Ignored“).
- „Title“: Naziv alarma i oznake.
- „Source“: Naziv izvora alarma.
- „Severity“: Ozbiljnost, prioritet alarma.
- „Attributes“: Dodatna svojstva koja dolaze s alarmom, a definira ih izvor alarma.
- „Date“: Datum kada je alarm zaprimljen u TheHive-u.



Slika 12: Lista alarma unutar TheHive sučelja (autorski rad)

S desne strane unutar svakog retka, koji predstavlja jedan alarm, postoje tri ikone koje omogućuju daljnje upravljanje alarmima. Za svaki alarm moguće je napraviti sljedeće akcije:

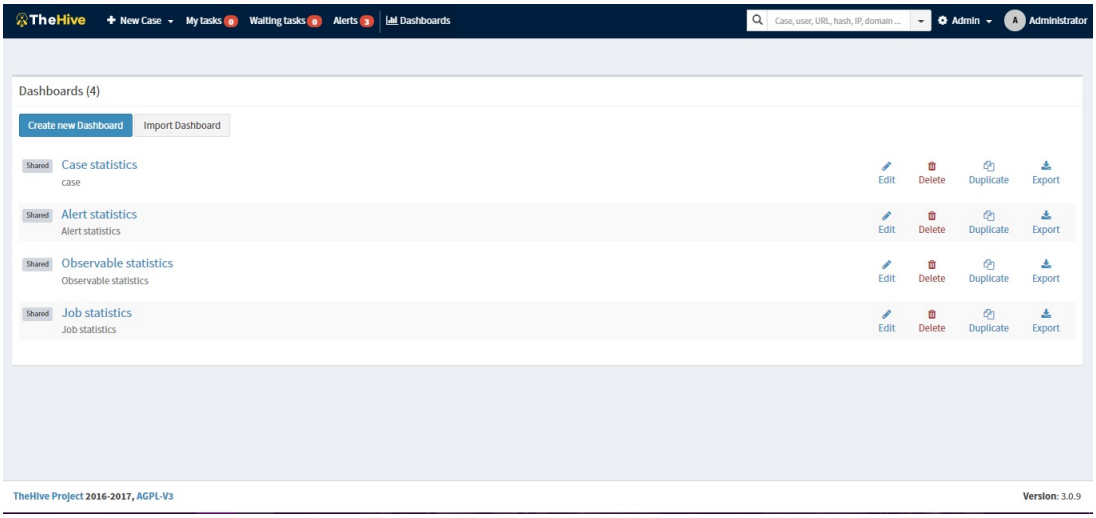
- „Alert preview & import“: Otvara novi prozor koji omogućuje uvez alarma kao praznog slučaja (zasad uvez alarma kroz predložak nije dozvoljen), s time da će se određeni detalji alarma preslikati u odgovarajuća polja novog slučaja. Unutar ovog prozora moguće je odabrati i sljedeće dvije akcije: „Mark as read“ i „Ignore new updates“. Primjer na slici 13.
- „Mark as read“: Označuje alarm kao pročitan, no i dalje će se primati novosti o ovom alarmu, tj. njegova ažuriranja od izvora alarma koji ga je stvorio (u tom slučaju će kod ažuriranja alarma on ponovo dobiti status „New“).
- „Ignore new updates“: Zahtjeva se prestanak praćenja i zaustavljanje ažuriranja od strane pošiljaoca za konkretni alarm. Pošiljaoc će slati ažuriranja, no ona će biti odbačena.



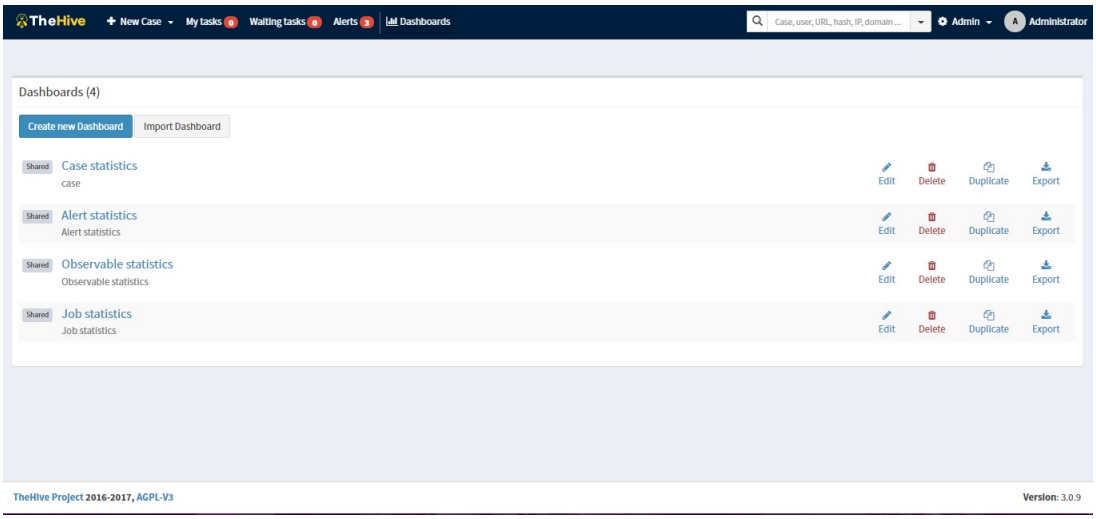
Slika 13: „Alert preview & import“ prozor (autorski rad)

10.1.5. Upravljanje statistikom i njenom vizualizacijom

Kontrolna ploča (eng. „Dashboard“), sa različitim vrstama i brojem grafova koji prikazuju statistike, kategorizirana je prema slučajevima, analizama artefakata, samim artefaktima i alarmima. Uz filtriranje po vremenskom periodu, moguće je spremiti grafove u datoteke formata csv i png, dok je pojedinu kategoriju statistike moguće spremiti u json formatu datoteke. Kontrolna ploča početno dolazi sa četiri različite statistike: statistike slučajeva, alarma, artefakata i analiza (slika 14).



Slika 14: Kontrolna ploča (autorski rad)



Slika 15: Primjer jedne statistike (autorski rad)

Na slici 15 prikazan je primjer prozora statistike. Postoji neograničen broj redaka (eng. „Row“) u koji je moguće smjestiti između nula i tri vizualizacije. Također, unutar svakog prozora statistike postoji alatna traka (s desne strane prozora) koja se prikazuje pritiskom na gumb „Edit“. Pristup alatnoj traci ima korisnik sa pravom Admin ili sa pravom Write (jedino ako je taj isti korisnik stvorio tu statistiku). Unutar alatne trake moguće je odabrati sljedeće akcije/elemente:

- „Save“: Spremi dosad napravljene izmjene.

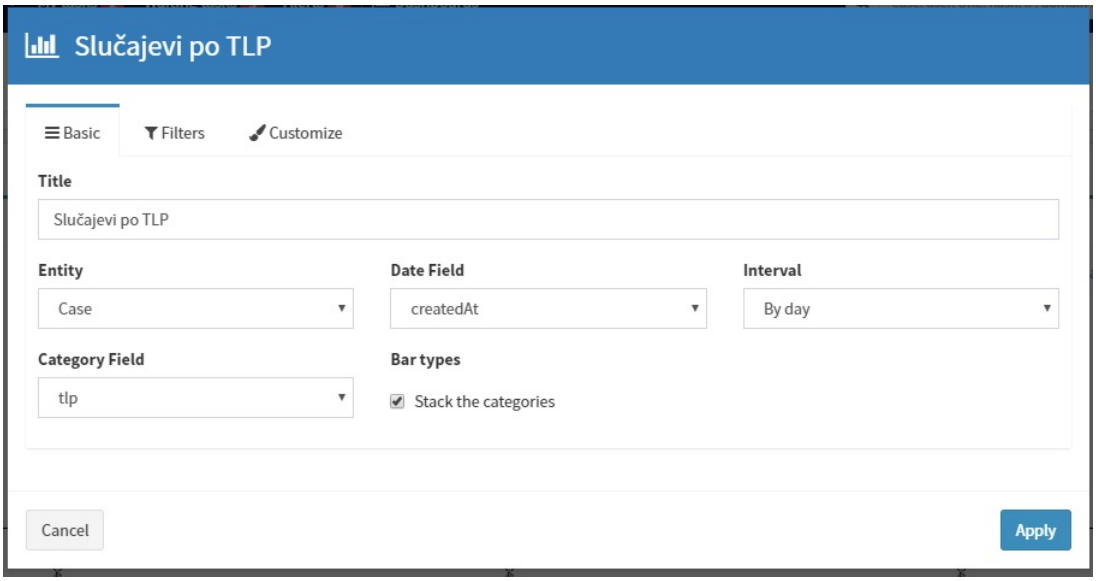
- „Cancel“: Poništi dosad napravljene izmjene.
- „Row“: Redak, odnosno svojevrsna „ladica“ za vizualizacije. Može se postaviti bilo gdje u kategoriji statistike (iznad ili ispod postojećih „ladica“). U nju stanu između nula do tri grafa istih ili različitih vrsta.
- „Bar“: Stupčasti graf. Ovaj graf je implementiran tako da može prikazivati linijski, više linijski i stupčasti graf (ovim grafom moguće je postići isti prikaz kao i kod „Bar“ i „Multi Lines“).
- „Line“: Linijski graf.
- „Multi Line“: Višelinijski graf. Ovaj graf je implementiran tako da može prikazivati linijski, više linijski i stupčasti graf (ovim grafom moguće je postići isti prikaz kao i kod „Bar“ i „Lines“).
- „Donut“: Kružni graf.
- „Counter“: Brojčani graf. Reprezentiran je kao tablica bez obruba, gdje je lijevi stupac zapravo cijeli broj, dok je desni stupac oznaka.

Vizualizacije su jedna od "najbitnijih" značajki ovakvih vrsta alata. Jedan razlog je taj što pružaju vizualni uvid u rad stručnog tima, događanja te vlastitih čimbenika (stvaraju dodatnu vrijednost za konkretnog korisnika). Drugi razlog je možda još i bitniji – gradivni su element završnih izvještaja. Grafovi te općenito statistike „vidljivi“ su dio poslovanja i učinkovitosti stručnog tima, stoga ću česti dokazi kvalitete rada stručnog tima unutar izvještaja i ostalih dokumentacija. Zbog omogućene visoke razine konfiguracije pojedinih grafova, vizualizacije su veoma korisna funkcionalnost, ali samo ako korisnik zna što želi prikazati, a zatim i „kako to prikazati“.

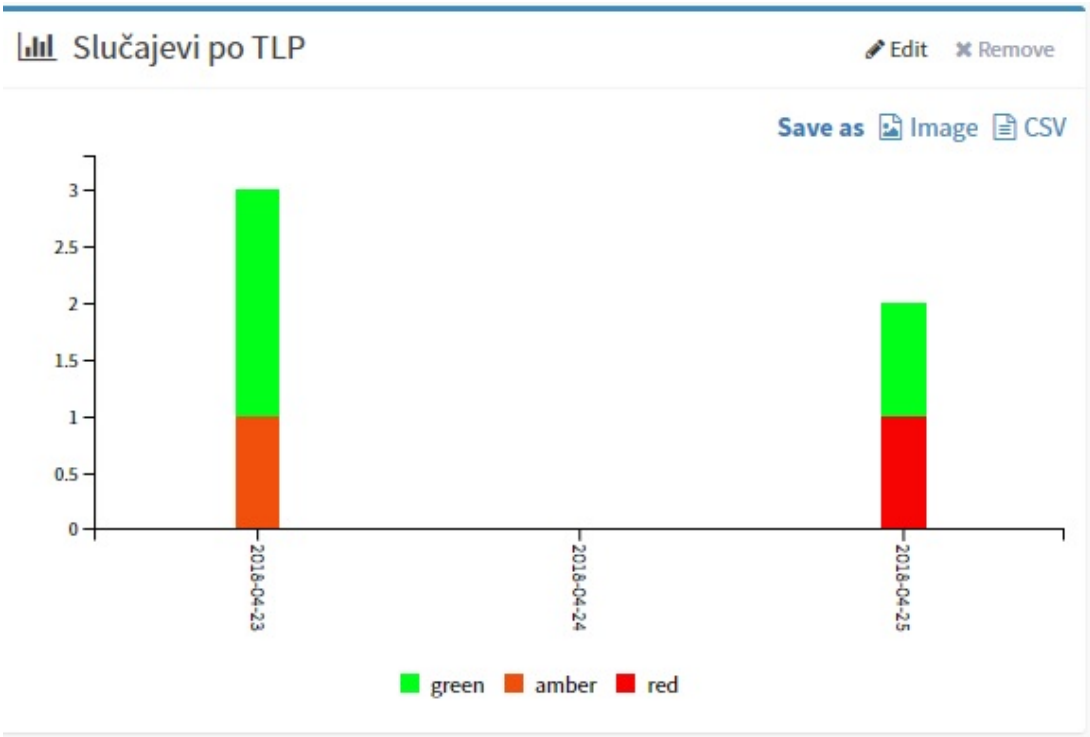
Slika 16 prikazuje prozor za stvaranje stupčastog grafa, dok slika 17 prikazuje primjer takvog grafa nakon stvaranja. Kako bi se pravilno definirao stupčasti graf potrebno je popuniti više ili sva polja (ovisno o kontekstu i kombinaciji odabranih polja):

- „Title“: Naziv grafa koji se pojavljuje u statistici.
- „Entity“: Baza koju će graf kontekstno prikazivati, a o njoj ovise odabiri kategorija za prikaz (konkretni podaci koji su reprezentirani kao stupci na grafu) i filteri. Moguće baze: alarm, slučaj, artefakt, posao (analiza artefakta), zadatak.
- „Date Field“: Polje koje označuje vrijeme, koje ovisi o bazi te smještava kategorije za prikaz na grafu. Moguće je odabrati datum stvaranja, datum ažuriranja, datum isteka zadatka, datum početka, datum završetka/zatvaranja.
- „Interval“: Vremenski interval koji definira prikazivanje podataka na grafu u vremenskom rasponu. Ne ovisi o bazi, tj. za svaku konfiguraciju grafa može se odabrati vremenski interval po danima, tjednima, mjesecima ili godinama.

- „Category field“: Kategorija za prikaz na grafu, tj. konkretni podaci na grafu. Izbor kategorija ovisi o bazi, no ne postoji kategorija u izboru koja se ne pojavljuje unutar TheHive sučelja. Neke kategorije moguće je dodano podesiti, poput oznake i boje stupca na grafu (u alatnoj traci se pojavljuje novi kartica odabira - „Customize“).
- „Bar types“: Kod prikaza, svaki stupac zasebno ili naslagani jedan na drugog.



Slika 16: Prozor za stvaranje stupčastog grafa (autorski rad)

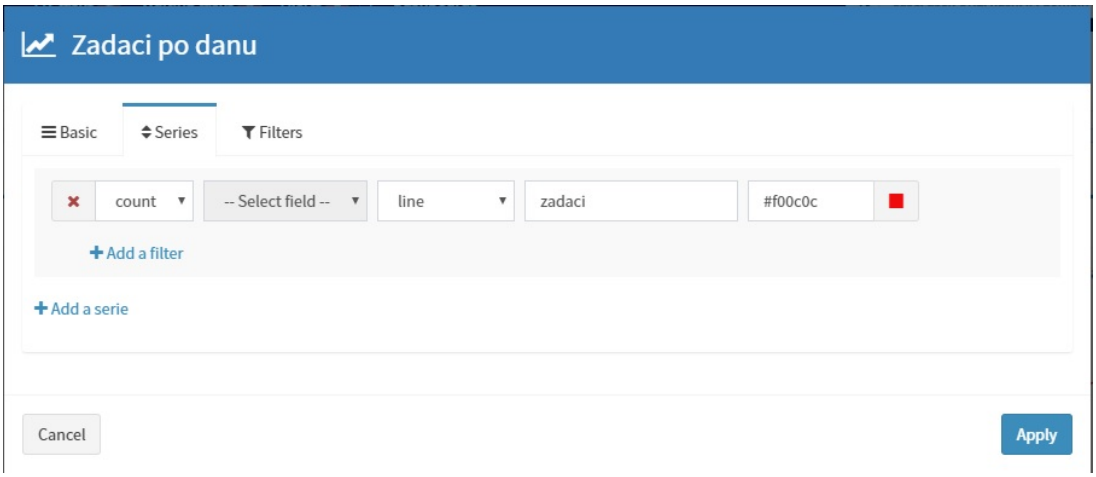


Slika 17: Primjer stvorenog stupčastog grafa (autorski rad)

Slika 18 prikazuje prozor za stvaranje linijskog grafa (kartica „Series“), dok slika 19 prikazuje

primjer takvog grafa nakon stvaranja. Kako bi se pravilno definirao linijski graf potrebno je popuniti više ili sva polja (ovisno o kontekstu i kombinaciji odabranih polja):

- „Title“: Naziv grafa koji se pojavljuje u statistici.
- „Entity“: Baza koju će graf kontekstno prikazivati, a o njoj ovise odabiri kategorija za prikaz (konkretni podaci koji su reprezentirani kao stupci na grafu) i filteri. Moguće baze: alarm, slučaj, artefakt, posao (analiza artefakta), zadatak.
- „Date Field“: Polje koje označuje vrijeme, koje ovisi o bazi te smještava kategorije za prikaz na grafu. Moguće je odabrati datum stvaranja, datum ažuriranja, datum isteka zadatka, datum početka, datum završetka/zatvaranja.
- „Interval“: Vremenski interval koji definira prikazivanje podataka na grafu u vremenskom rasponu. Ne ovisi o bazi, tj. za svaku konfiguraciju grafa može se odabrati vremenski interval po danima, tjednima, mjesecima ili godinama.
- „Series“: Konkretni podaci koji će biti prikazani na grafu. Potrebno je definirati agregacijsku funkciju, zatim ovisno o bazi i agregacijskoj funkciji definirati kategoriju za prikaz, vrstu grafa, oznaku i boju.

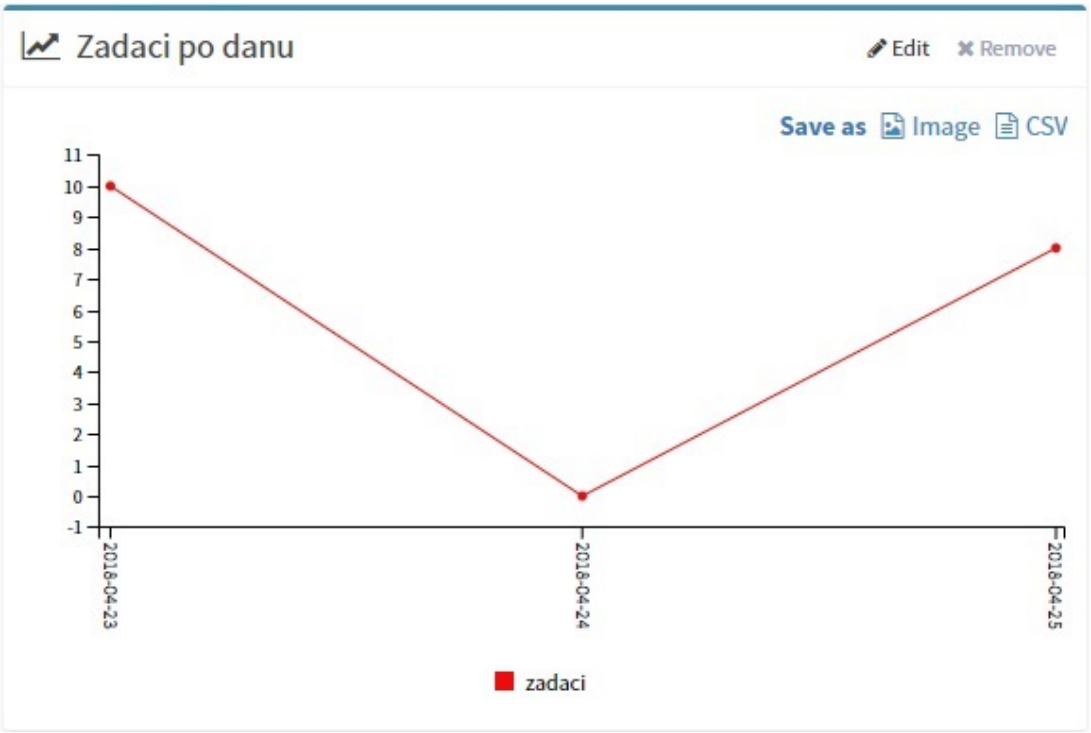


Slika 18: Prozor za stvaranje linijskog grafa (autorski rad)

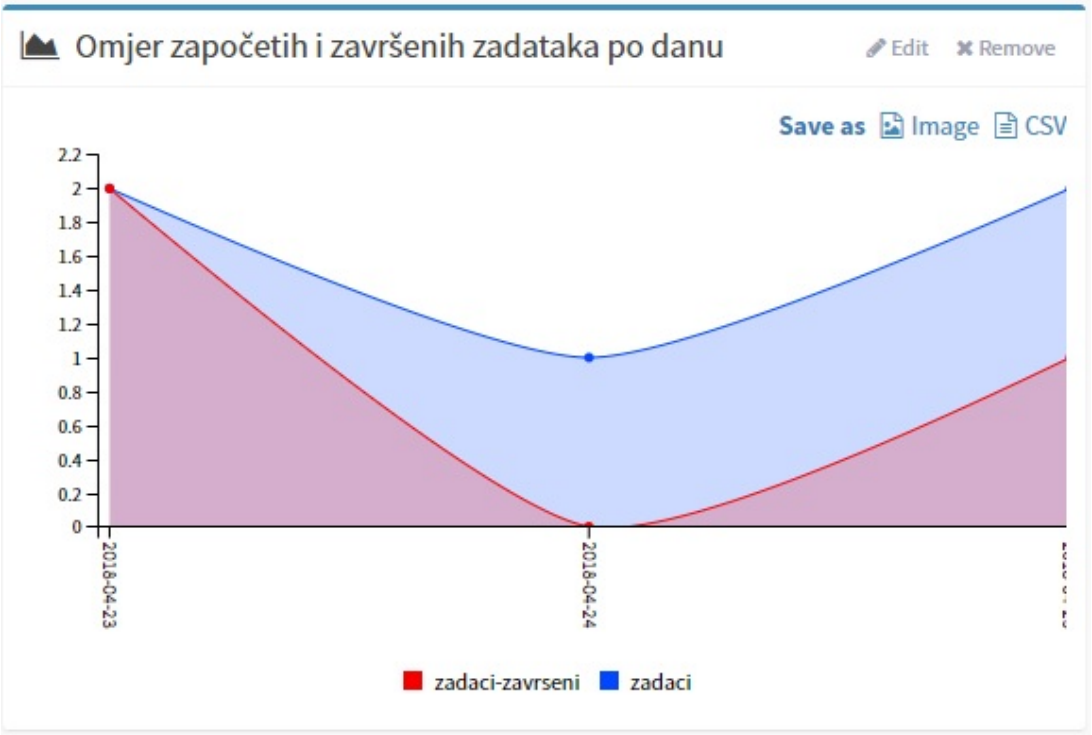
Osim što nema „Entity“ i „Date Field“, višelinijski graf se previše ne razlikuje od linijskog. Napomenuto je kako se korištenjem jedne vrste grafa može implementirati drugi i obrnuto. Jedan primjer višelinijskog grafa nalazi se na slici 20.

Slika 21 prikazuje prozor za stvaranje kružnog grafa (kartica „Series“), dok slika 21 prikazuje primjer takvog grafa nakon stvaranja. Kako bi se pravilno definirao kružni graf potrebno je popuniti više ili sva polja (ovisno o kontekstu i kombinaciji odabranih polja):

- „Title“: Naziv grafa koji se pojavljuje u statistici.
- „Entity“: Baza koju će graf kontekstno prikazivati, a o njoj ovise odabiri kategorija za prikaz (konkretni podaci koji su reprezentirani kao stupci na grafu) i filteri. Moguće baze: alarm, slučaj, artefakt, posao (analiza artefakta), zadatak.



Slika 19: Primjer stvorenog linijskog grafa (autorski rad)

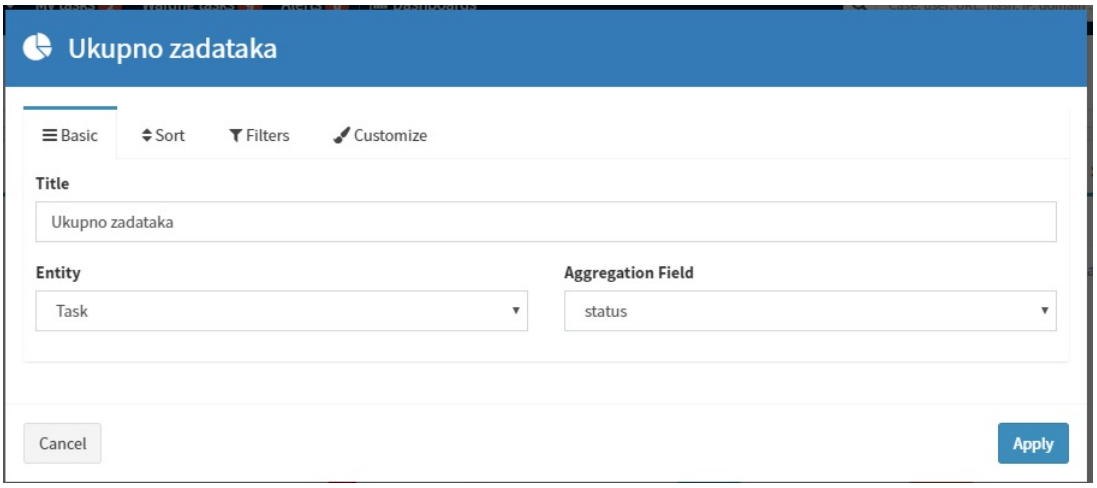


Slika 20: Primjer stvorenog višelinijskog grafa (autorski rad)

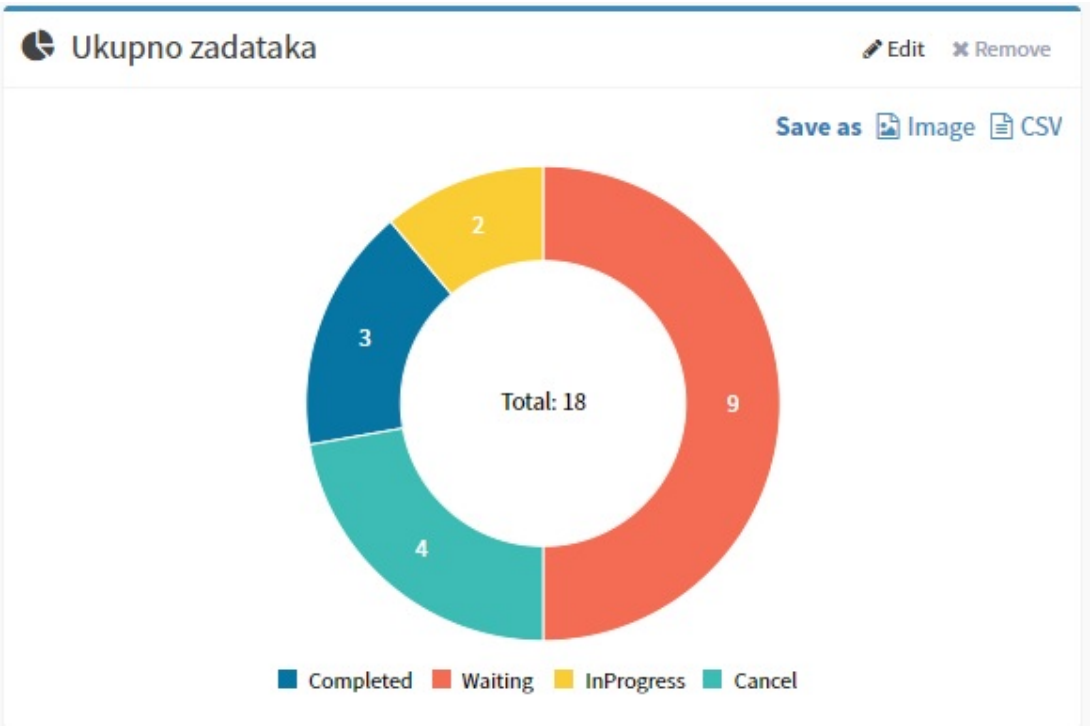
- „Date Field“: Polje koje označuje vrijeme, koje ovisi o bazi te smještava kategorije za prikaz na grafu. Moguće je odabrati datum stvaranja, datum ažuriranja, datum isteka zadatka, datum početka, datum završetka/zatvaranja.
- „Aggregation Field“: Odabir polja koje će reprezentirati kategoriju (ili više njih) za prikaz.

Nad svakom kategorijom radi se prebrojavanje (analogno radnjama GROUP BY -> COUNT unutar termina relacijskih baza podataka). Izbor polja ovisi o bazi.

- „Sort“: Limitiranje prikaza vrijednosti kategorija. „Limit“ označuje koliko grupa vrijednosti kategorija se uzima (cijeli broj) u obzir, a „Sort by“ uzima grupe od manje vrijednosti prema većoj ili obrnuto.

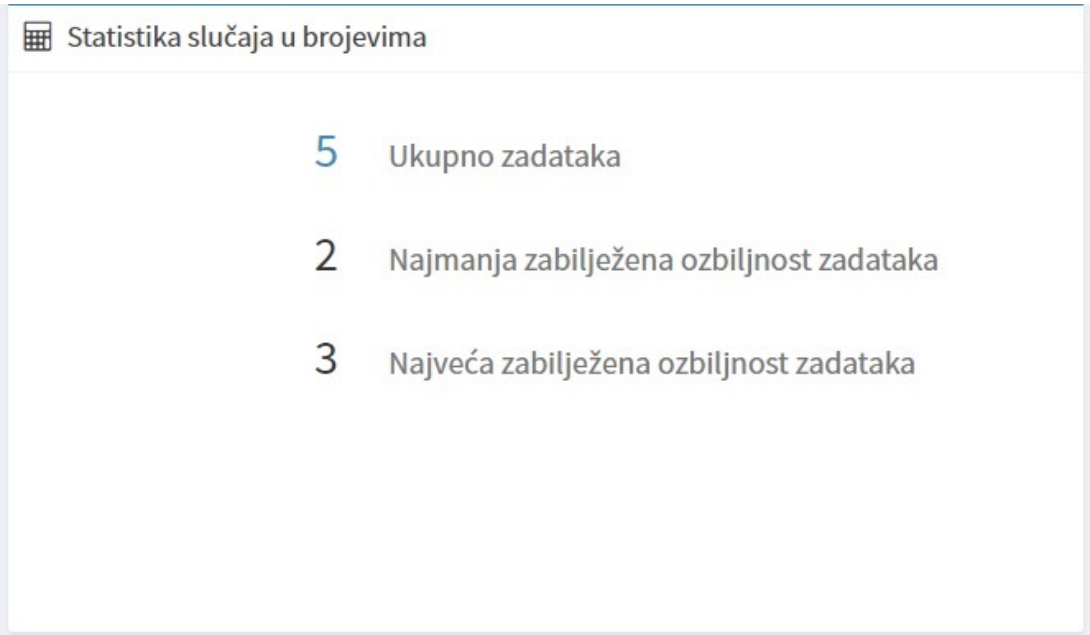


Slika 21: Prozor za stvaranje kružnog grafa (autorski rad)



Slika 22: Primjer stvorenog kružnog grafa (autorski rad)

Brojčani graf definira se isto kao i višelinijski. Razlika je što više linijski prezentira podatke vizualno kroz graf, dok brojčani to radi improvizirajući tablicu sa parovima naziv (niz znakova) - brojčana vrijednost (slika 23).



Slika 23: Primjer stvorenog brojčanog grafa (autorski rad)

Napomena: Neke od brojčanih vrijednosti označene su plavom bojom (kao, na primjer, broj 5 na slici 23). U tom slučaju, klik na tu brojčanu vrijednost prebacuje korisnika na rezultate tražilice prilagođenog upita.

Kod svake vrste grafa mogu se definirati i filteri – na samoj razini grafa (opcija u alatnoj traci prozora stvaranja/uređivanja grafa) ili na razini konkretnih podataka koji će biti prikazani na grafu („Series“). Kod vremenskih podataka za filtriranje, moguće je filtrirati prema vrijednosti gdje su uvjeti od-do. Kod tekstualnih podataka za filtriranje, moguće je filtrirati eksplicitno prema tekstu ili prema operatorima „any of“, „all of“ ili „none of“. Kod metrika moguće je filtrirati vrijednosti po relacijskim operatorima (kao kod programskih jezika). Slijedi tablica opcija (1)) u alatnim traka za pojedinu vrstu grafa. Znak plus označava prisutnost opcije, znak minus odsutnost, dok oba znaka podjeljena kosom crtom označavaju da prisutnost ovisi o drugim odabranim parametrima/poljima tokom stvaranja vizualizacije.

Tablica 1: Tablica opcija u alatnim traka za pojedinu vrstu grafa

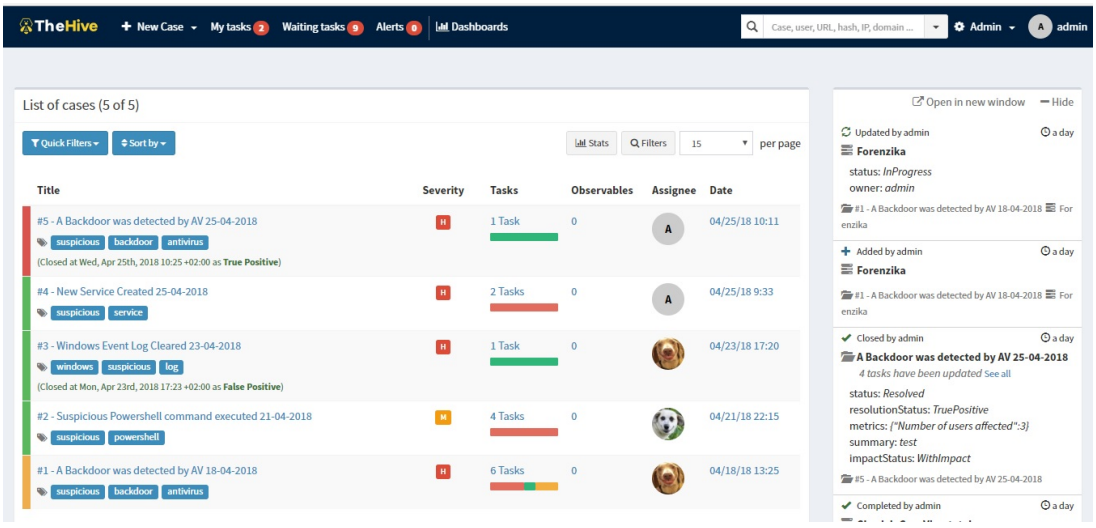
	Osnovno (eng. „Basic“)	Podaci (eng. „Series“)	Filtri (eng. „Filters“)	Dodatno (eng. „Customize“)
Stupčasti	+	-	+	+/-
Linijski	+	+	+	-
Višelinijski	+	+	-	-
Kružni	+	+	+	+/-
Brojčani	+	+	+	-

(Izvor: Autorski rad)

10.1.6. Upravljanje „Društvenom mrežom“

TheHive platformu slobodno se može nazvati „Facebookom“ za odziv i upravljanje incidentima. Administrator može stvoriti nekoliko vrsta korisnika (različitih prava), gdje svaki korisnik ima svoje puno ime, korisničko ime, lozinku, avatar te API ključ. Ovisno o pravima korisnika, moguće su različite akcije, no svaki tip korisnika može vidjeti sve trenutne slučajeve, zadatke i njihove evidencije (može dodavati čak i dodavati evidencije zadacima kod zatvorenih slučajeva), kontrolnu ploču, alarme te filtrirati pregled prema nekoliko opcija. U sljedećem potpoglavlju uloge korisnika biti će detaljnije objašnjene. Također, postoji i ažurirajuća statusna traka događaja/objava sortirana prema vremenu te tražilica koja omogućuje traženje bilo kojeg elementa TheHive platforme (korisnik, slučaj, zadatak, evidencija, artefakt, itd.).

Alatna traka, skupa sa tražilicom te statusna traka događaja („povijest događaja“ u nastavku rada) vidljive su već na početnom prozoru TheHive sučelja, tj. nakon prijave korisnika. Slika 24 prikazuje primjer početnog prozora, čija je glavna zadaća prikazati korisniku listu svih slučajeva ažuriranih vremenskim redoslijedom. Moguće je dodatno prikazati sažetu statistiku (top 5 oznaka, zbroj otvorenih i zatvorenih slučajeva, itd.) i filtrirati je prema složenim kriterijima: ključne riječi, status, oznake, korisnici koji su napravili slučaj, ozbiljnost slučaja, naziv te vremenski interval za traženje slučajeva otvorenih/stvorenih unutar određenog vremena. Za svaki slučaj prikazan je njegov naziv i oznake te dodatne informacije o statusu slučaja (stupac „Title“), ozbiljnost slučaja (eng. „Severity), koliko zadataka postoji te koji je njihov status (stupac „Tasks“), koliko je artefakata pridijeljeno slučaju (eng. „Observables“), korisnik koji je stvorio/otvorio slučaj (eng. „Assignee“) te datum stvaranja/otvaranja slučaja (eng. „Date“). Ovisno o području klika, sa ovog prozora korisnika se preusmjerava na konkretni zadatak ili slučaj uz koji je zadatak vezan. Ovisno o području klika, sa ovog prozora korisnika se preusmjerava na konkretni slučaj,a u slučaju vrijednosti datuma ili ozbiljnosti, namješta se i pokreće filtar.

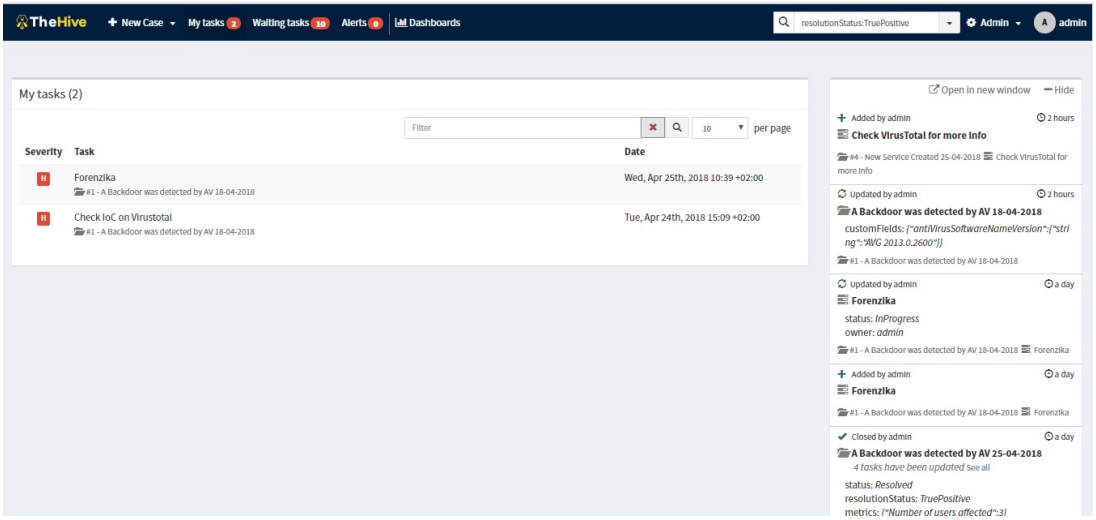


Slika 24: Početna stranica TheHive platforme (autorski rad)

Alatna traka (na slici 24 vidljiva kao gornji dio sučelja obojen u tamno plavo), s lijeva prema desno, sastoji se od opcija:

- „TheHive“: prozor sa listom slučajeva.
- „+ New Case“: padajući izbornik sa izborom izrade novog slučaja bez predloška ili s predloškom.
- „My tasks“: svi zadaci iz slučajeva kojima je trenutni korisnik pridijeljen (zadaci koje obavlja).
- „Waiting tasks“: svi zadaci iz slučajeva koji još nisu nikom pridijeljeni.
- „Alerts“: prozor sa listom alarma.
- „Dashboards“: prozor sa listom kategorija statistika.
- Pametna tražilica: prostor za pisanje upita prema ElasticSearch bazi podataka.
- „Admin“: padajući izbornik za administratora. Ovaj izbornik se ne vidi ako trenutni prijavljeni korisnik nije administrator.
- Osobne postavke: ovisno o avataru (mala slika korisnika) i punom imenu korisnika, ovaj padajući izbornik izgleda drugačije. Klikom na njega, moguće su sljedeće opcije „Personal settings“, „About TheHive“ i „Logout“.

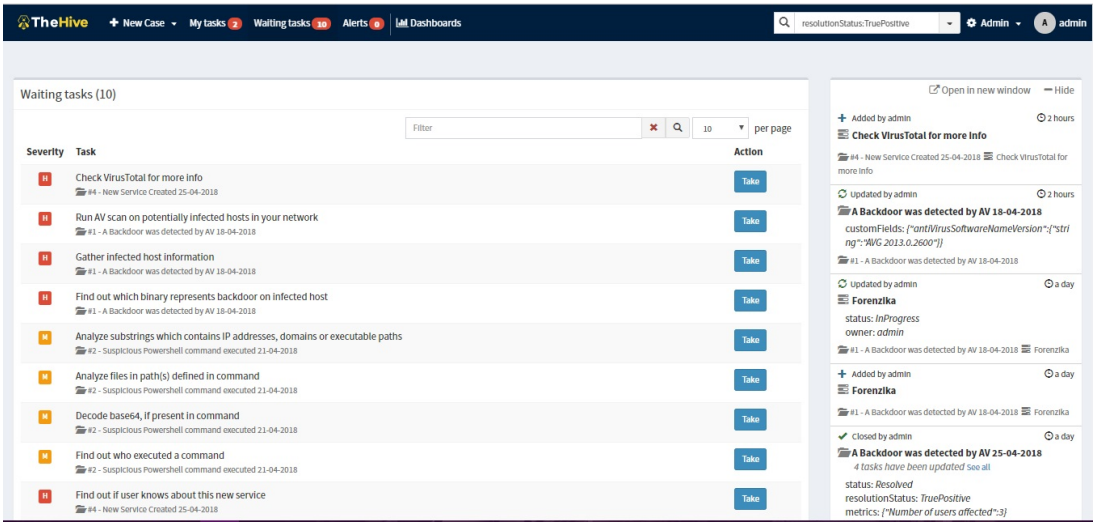
Klikom na „My tasks“ unutar alatne trake otvara se prozor sa svim zadacima iz slučajeva kojima je trenutni korisnik pridijeljen (zadaci koje obavlja). Sortirani su vremenski te je omogućeno filtrirati prema pojmu ili upitu na samom prozoru. Za svaki zadatak prikazana je ozbiljnost slučaja (eng. „Severity“), naziv zadatka i naziv slučaja u stupcu naziva „Task“, te vrijeme kada je zadatak korisnik preuzeo (eng. „Date“). Ovisno o području klika, sa ovog prozora korisnika se preusmjerava na konkretni zadatak ili slučaj uz koji je zadatak vezan. Slika 25 prikazuje primjer ovog prozora.



Slika 25: „My tasks“ prozor (autorski rad)

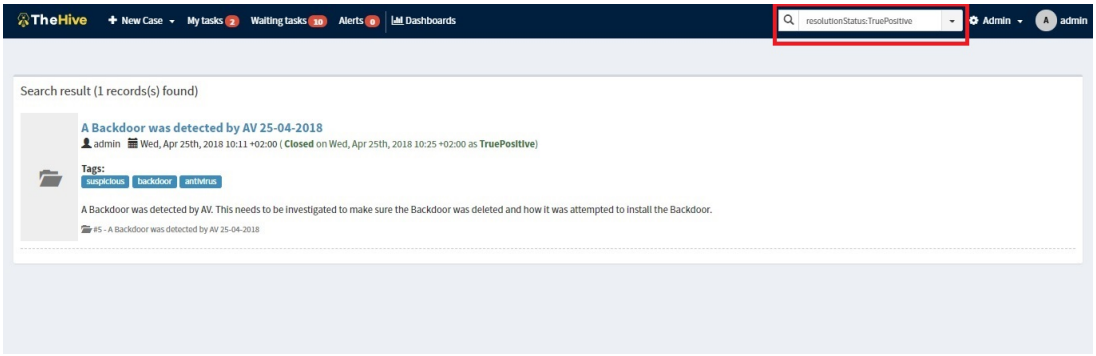
Klikom na „Waiting tasks“ unutar alatne trake otvara se prozor sa svim zadacima iz slučajeva koji nisu nikom pridijeljeni. Korisnik može na ovom prozoru na brži način preuzeti na sebe

zadatak koji god želi. Naravno, takav korisnik mora imati barem „write“ prava. Zadaci su sortirani vremenski te je omogućeno filtrirati prema pojmu ili upitu na samom prozoru. Za svaki zadatak prikazana je ozbiljnost slučaja (eng. „Severity“), naziv zadatka i naziv slučaja u stupcu naziva „Task“, te plavi gumb koji omogućuje preuzimanje zadatka (stupac „Action“). Ovisno o području klika, sa ovog prozora korisnika se preusmjerava na konkretni zadatak ili slučaj uz koji je zadatak vezan. Slika 26 prikazuje primjer ovog prozora.



Slika 26: „Waiting tasks“ prozor (autorski rad)

U alatnoj traci postoji prostor za upis pojmova i jednostavnih upita prema ElasticSearch-u. Od pojmova mogu se pretraživati slučajevi, zadaci, oznake (eng. „tags“), korisnici i artefakti. Od upita mogući su oni u zapisu tipa atribut:vrijednost ili oni upiti koji su već predefinirani (padajući izbornik pametne tražilice). Ako su pojmovi odvojeni zarezom, traži se svaki element koji sadrži barem jedan pojam iz tražilice. Ako su ti pojmovi zajednički omeđeni navodnicima, tada se traži svaki od tih pojmova, istim redoslijednom kako je napisano. Također, moguće je koristiti logičke operatore AND, OR i NOT ili skraćeno &&, || i ! Jedan od korisnih upita je „resolutionStatus:TruePositive“ koji pokazuje sve zatvorene slučajeve koji su označeni kao incident (eng. „TruePositive“). Primjer postavljenog upita nalazi se na slici 27.



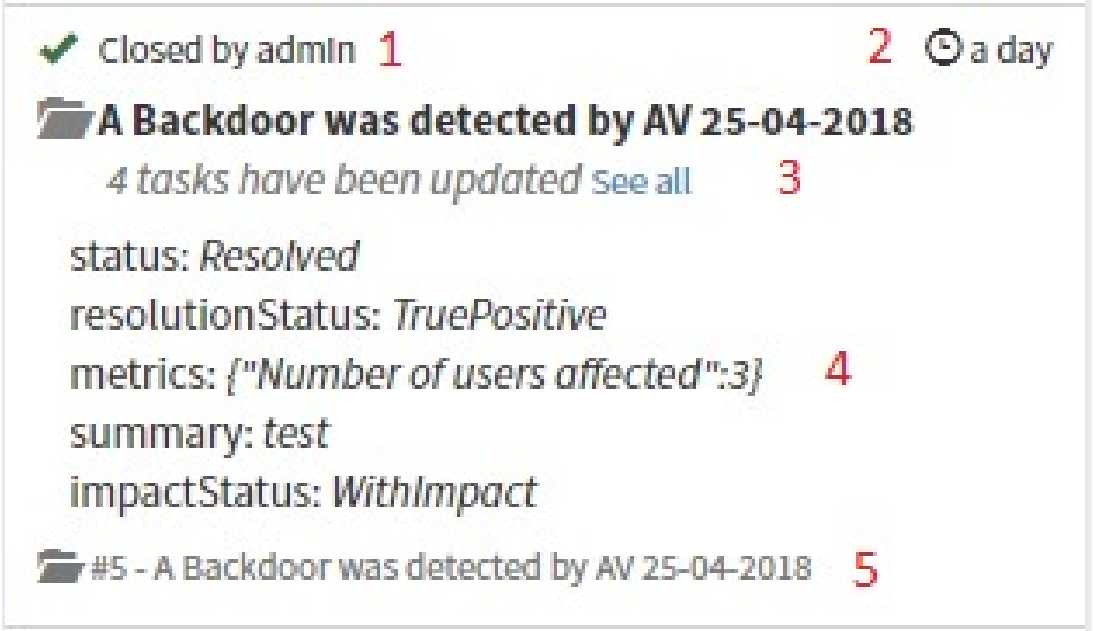
Slika 27: Pametna tražilica sa upitom „resolutionStatus:TruePositive“ i rezultatom upita (autorski rad)

Povijest događaja (primjer na slici 29) služi kao evidencija radnji koje korisnici obavljaju unutar TheHive-a. Ono što se ne evidentira su čitanje i radnje iz opcija Admin izbornika. Radnje koje

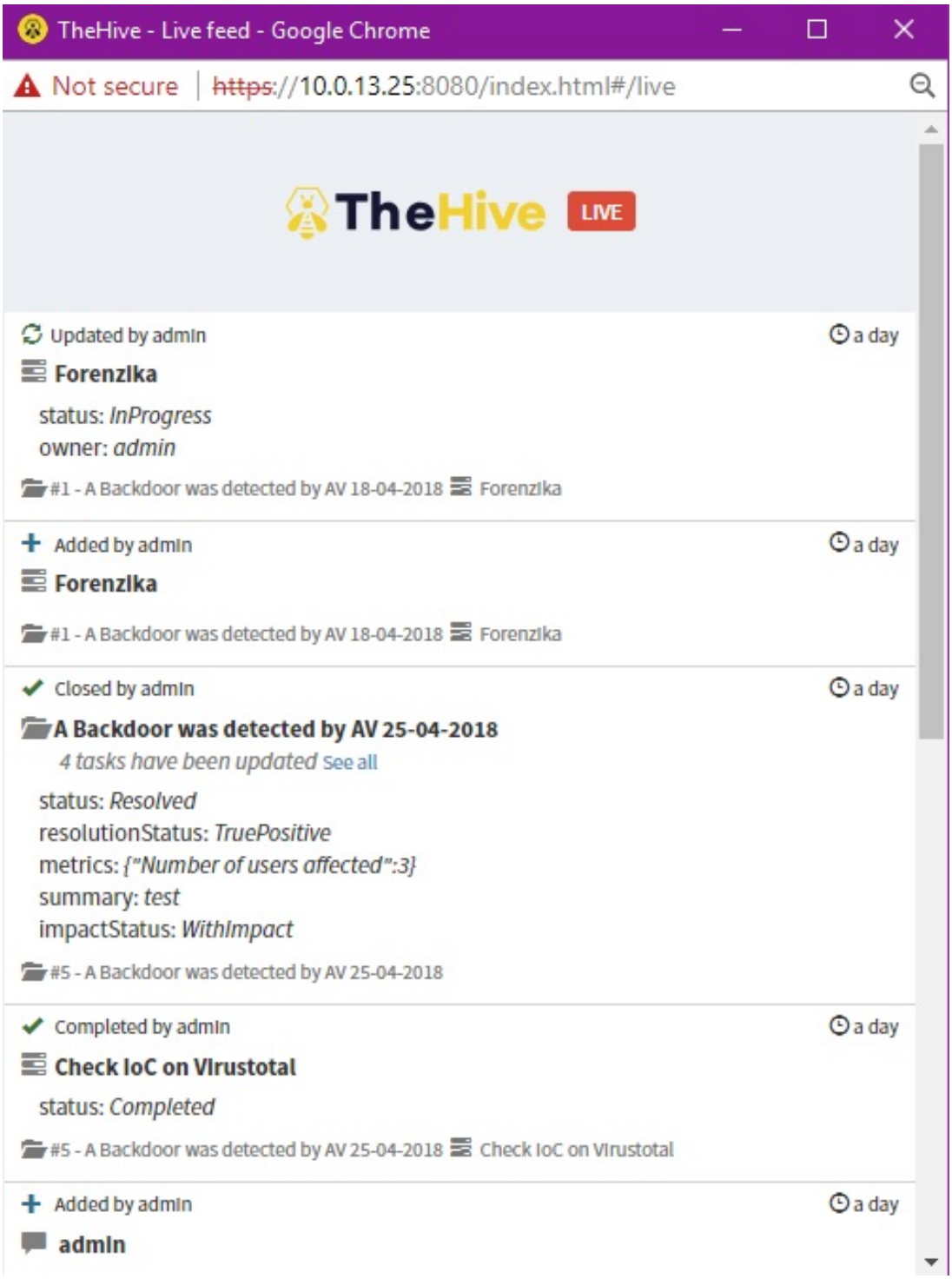
se prikazuju ovise o kontekstu (tj. prozoru) u kojem se korisnik trenutno nalazi. Na primjer, u prozoru povijesti će se prikazivati radnje nad svim slučajevima, ako je korisnik unutar prozora sa listom slučajeva. Tek dok će korisnik kliknuti na jedan slučaj, povijest će se promijeniti tako da će prikazivati radnje vezane samo uz taj slučaj (i njegove zadatke, analize, evidenciju zadataka, itd.). Predefinirano je da se povijest događaja pojavljuje s desne strane, no može se (kao što je prikazano na gornjoj slici) izvući van u posebni prozor pritiskom na opciju „Open in new window“. Za vraćanje potrebno je kliknuti na „Show live stream“. Sukladno tome, opcija „Hide“ sakriva povijest događaja.

Povijest događaja „puni“ se događajima FIFO redom (prvi unutra, prvi van), s time da je moguće imati maksimalno 10 događaja u redu. Jedan događaj (kao na primjer ovaj sa slike 28) sastoji se od:

1. Globalna radnja: glagol (dodano, ažurirano, zatvoreno. . .) + korisnik
2. Vrijeme nastanka događaja (u odnosu na sadašnje vrijeme)
3. Element nad kojim je radnja izvršena. To može biti slučaj, zadatak, evidencija zadatka ili analiza
4. Specifična radnja, kao dopuna globalnoj: atribut + vrijednost (u json formatu, zbog podrške ElasticSearch-u) ili samo tekst (evidencija zadatka)
5. Ako je radnja vezana uz kontekst slučaja ili analize artefakta, onda piše naziv slučaja. Ako je radnja vezana uz kontekst zadatka (ili njegovu evidenciju) nekog slučaja, tada piše naziv slučaja i naziv tog zadatka



Slika 28: Primjer događaja iz povijesti događaja (autorski rad)

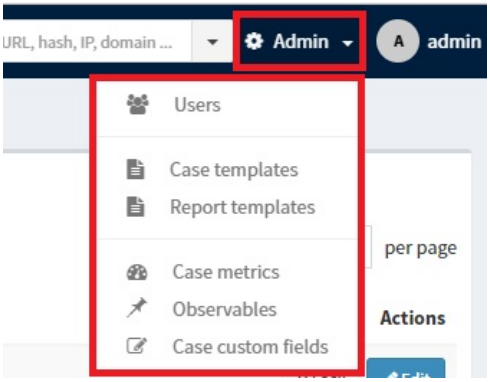


Slika 29: Primjer povijesti događaja (autorski rad)

10.1.7. Upravljanje platformom (Administracija)

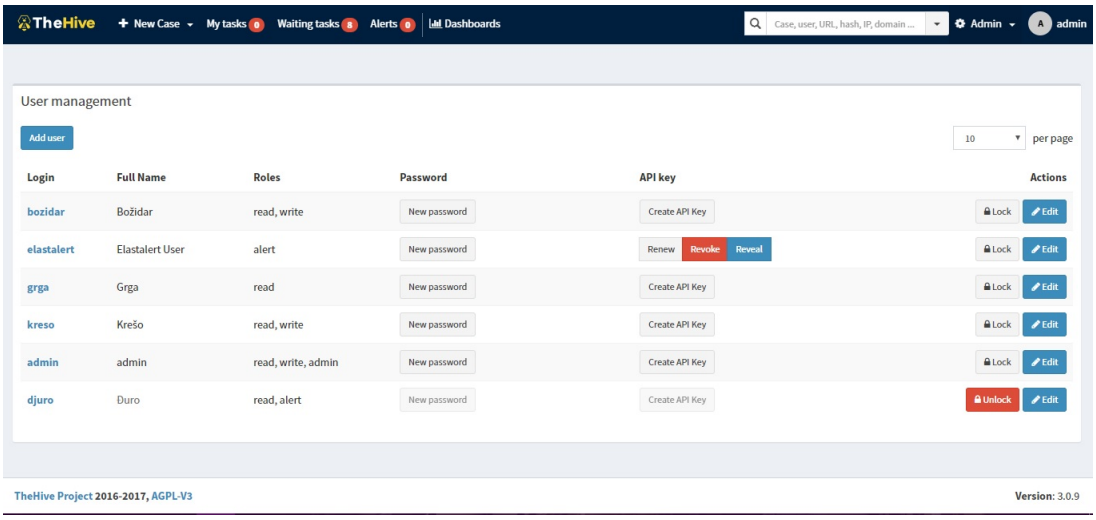
Administrator može koristiti sve funkcionalnosti TheHive platforme, koje su pružane kroz web sučelje (prema predefiniranim konfiguracijskim postavkama). Za razliku od uloge urednika (eng. „Write“), administrator može i potpuno uređivati kontrolnu ploču (uređivati ili brisati bilo koju statistiku na razini grafova, ali i kategorije statistike) te koristiti dodatni padajući izbornik u desnom gornjem kutu kao što je prikazano na slici 30 (ikona mehaničkog kotačića i naziva

administratora).



Slika 30: Izbornik za administratora (autorski rad)

Klikom na „Users“ opciju otvara se sučelje za pregled postojećih korisnika (primjer na slici 31), s time da se njima mogu, redom s lijeva prema desno, dodijeliti nove zaporke, API ključ (npr. za slanje alarma ili korištenje TheHive4py*), zatim mogu se o(ne)mogućiti korisnički računi (budući da TheHive ne dopušta striktno brisanje korisničkog računa), te urediti podaci o korisnicima (sve osim korisničkog imena, jednom definirano ono se ne smije mijenjati).

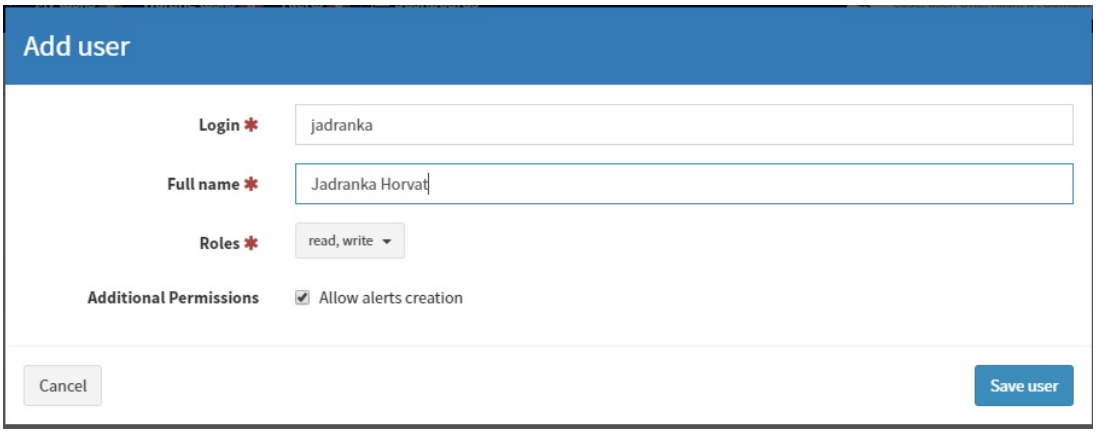


Slika 31: Prozor za upravljanje korisnicima (autorski rad)

Klikom na plavi gumb „Add user“ ili „Edit“ otvara se sučelje u kojem se stvara novi korisnik, odnosno uređuje postojeći, kao što je prikazano na slici 32. Polja koja se moraju definirati su korisničko ime koje se koristi kod prijave na TheHive sučelje (eng. „Login“), puno pravo ime kojim se korisnik identificira unutar TheHive sučelja (eng. „Full name“), zatim uloge koje korisnik može imati („none“, „read“, „read-write“, „read-write-admin“), a identificiraju korisnika prema njegovim pravima i pristupima određenim funkcionalnostima (eng. „Roles“) i dodatna prava poput mogućnosti stvaranja alarma (eng. „Additional Permissions“).

Opis pojedine uloge:

- „Write“: Uloga urednika unutar TheHive platforme. Korisnik sa „write“ pravima može čitati i izvoziti „sve što se može“ (tj. sve što TheHive prikazuje kao djeljivo), te stvarati



Slika 32: Prozor za stvaranje ili uređivanje korisnika (autorski rad)

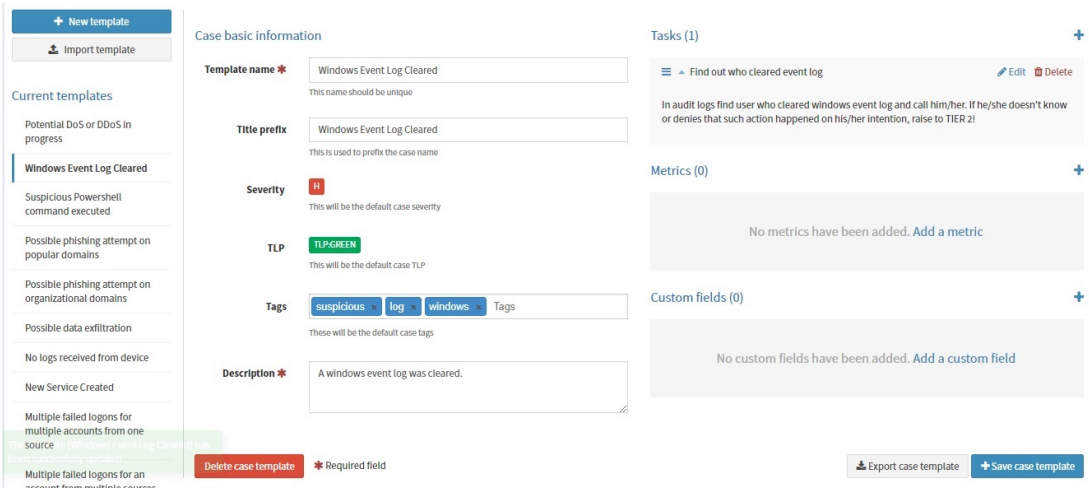
slučajeve, zadatke, logove, prenijeti artefakte, pokrenuti analize te stvarati, uvesti, uređivati i brisati svoje statistike (ali ne i tuđe – to može samo administrator).

- „Read“: Uloga čitatelja/promatrača. Korisnik sa „read“ pravima može čitati i izvoziti „sve što se može“ (tj. sve što TheHive prikazuje kao djeljivo). Pisanje/izmjena nije mu dozvoljena.
- „None“: Korisnik nema trenutno nikakva prava. Ne preporučuje se napraviti korisnika samo sa „none“ pravima, osim ako se želi napraviti korisnika koji će moći stvarat alarme preko API-ja (kombinacija sa „alert“ pravom).
- „Alert“: Dobiva se označivanjem opcije „Allow alerts creation“. Takav korisnik ima prava stvarati alarme.

„Case templates“ su predlošci za slučajeve, koji ubrzavaju izgradnju slučaja temeljem predefiniranih vrijednosti polja. Polja koja se mogu definirati su ista ona polja koja su prikazana i kod stvaranja novog „praznog“ slučaja. Predlošci se mogu stvoriti, obrisati (jednom obrisani ne mogu se više vratiti), uvesti (specificirana datoteka json formata), te izvesti u json format. Napomena: bilo koja promjena unutar predloška, osim brisanja, neće biti spremljena ako se ne klikne na gumb „+ Save case template“. U protivnom, izmjene će nestati već kod klika na drugi predložak te vraćanja na prošli. Primjer jednog predloška nalazi se na slici 33.

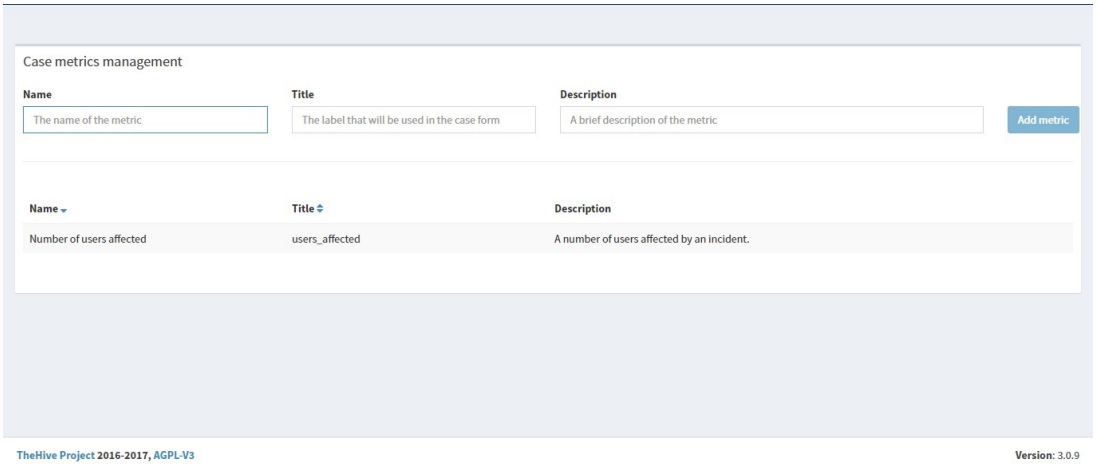
„Report templates“ su predlošci za Cortex, tj. predlošci u kojima navedena polja pojedini alat za analizu artefakata može koristiti za „mapiranje“ - radnja u kojoj se podaci, koje generira alat kod rezultata analize, prepisuju u polja koja su navedena u predlošku za taj konkretni alat. Budući da Cortex dolazi s nekoliko predefiniranih alata za analizu, postoji repozitorij u kojem se mogu preuzeti napravljeni predlošci za te alate. Bez predložaka za Cortex, automatizirane analize prikazivati će onakav oblik odgovora kakav analizator vrati (vrlo često je odgovor težak za čitanje ljudskom oku).

„Case metrics“ predstavlja stvaranje i pregled metrika – brojčanih svojstava koje se mogu pridružiti bilo kojem slučaju (što ovisi o kontekstu). Metrika se sastoji od punog naziva (eng. „Name“), naziva koji se prikazuje na sučelju kod pojedinog slučaja (eng. „Title“) te opisa koji



Slika 33: Prozor za stvaranje ili uređivanje predložaka za slučajeve (autorski rad)

„upotpunjuje“ naziv unutar sučelja (eng. „Description). Metrika može poprimiti samo cjelobrojne vrijednosti između -9223372036854775000 i 9223372036854775000. Prozor „Case metrics“ može izgledati poput ovoga na slici 34.

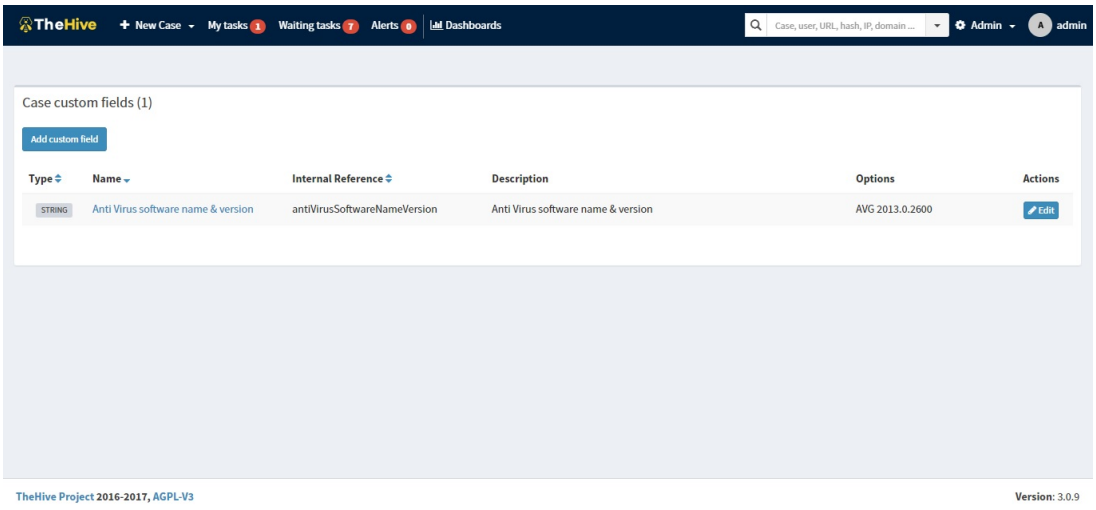


Slika 34: Prozor za stvaranje ili uređivanje metrika (autorski rad)

„Observables“ otvara prozor za stvaranje i pregled tipova podataka koji mogu biti pridijeljeni pojedinom artefaktu. Budući da su predefinirani tipovi podataka sasvim dovoljni, vrlo rijetko je potrebno uređivati ili dodavati nove tipove.

Dodatna polja, po ulozi slična metrikama, mogu se definirati unutar „Case custom fields“ (slika 35). Razlike su u tome što ona mogu biti tekstualnog, logičkog (istina/laž) te vremenskog (datum) tipa podatka, a ne samo brojčanog. Kod tekstualnog i brojčanog tipa, moguće je definirati predefinirane vrijednosti i tako pretvoriti element dodatnog polja u padajući izbornik. Ostale razlike su u tome što dodatna polja mogu u bilo kojem trenutku biti nedefinirana te se mogu izbrisati (ne vrijednosti koje sadrže, već kao sama polja iz slučaja). Dodatno polje sastoji se od punog naziva koji se prikazuje na sučelju kod pojedinog slučaja (eng. „Name“), naziva koji se prikazuje na definiranju grafova kod statistike (eng. „Reference“), opisa koji „upotpunjuje“ naziv na sučelju (eng. „Description), tipa podatka (eng. „Type“) te dodatnih

opcija u kojima se mogu definirati moguće vrijednosti tekstualnih i brojčanih podataka (eng. „Possible values“).



Slika 35: Prozor za stvaranje ili uređivanje dodatnih polja (autorski rad)

10.2. Cortex funkcionalnosti

Unutar ovog poglavlja slijedi kratak opis svih funkcionalnosti koje je autor prepoznao unutar rada Cortex-a (prikazane će biti autorove slike, testni podaci, položaj kontrola za obavljanje raznih funkcionalnosti. . .) te uz pomoć službene dokumentacije sa GitHub stranice Cortex platforme na adresi: https://github.com/TheHive-Project/CortexDocs

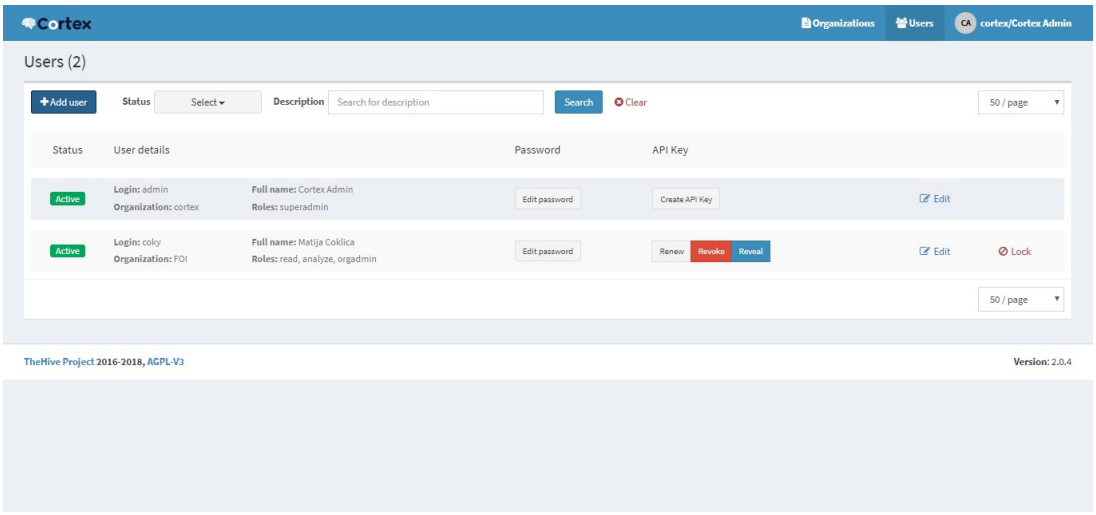
Budući da korisnik Cortex platformom može upravljati unutar posebnog prozora unutar sučelja TheHive-a, nabrojati će i ukratko opisati funkcionalnosti koje su dostupne isključivo iz web sučelja Cortex-a.

10.2.1. Podjela i upravljanje analitičarima

Cortex nakon instalacije i prve prijave stvara administratora, čiji je zadatak napraviti organizacije i njihove korisnike sa pripadajućim ulogama. Organizacija je svojevrsna kolekcija analiza i analitičara, gdje analitičare stvara i dodjeljuje administrator, dok analize pokreću upravo ti korisnici (administratoru nije dopušteno). Također, korisnici jedne organizacije ne mogu pokrenuti nove ili pogledati gotove analize neke druge organizacije. Postupak dodavanja korisnika je vrlo sličan kao i kod TheHive platforme. Slika 36 pokazuje primjer liste sa korisnicima i opcije preko kojih se upravlja postojećim ili novim korisnicima.

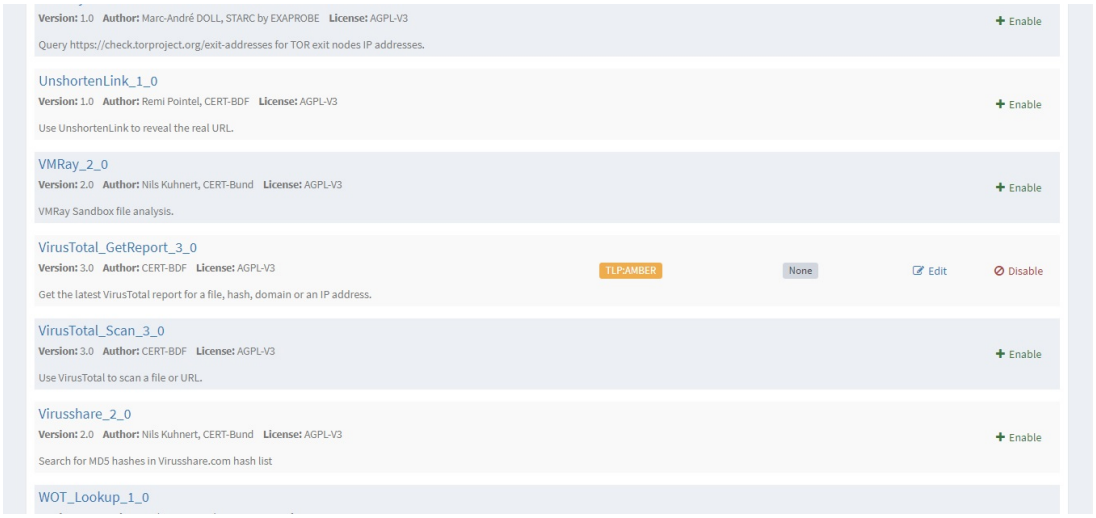
10.2.2. Upravljanje analizatorima

Uz sedamdesetak ponuđenih analizatora od strane Cortex-a (Python agenti koje su TheHive-Project razvojni inženjeri napisali i ponudili krajnjem korisniku), mogu se napraviti novi analizatori te se dodatno konfigurirati unutar web sučelja. Jedan dio liste analizatora prikazan



Slika 36: Primjer prozora sa upravljanjem korisnicima (autorski rad)

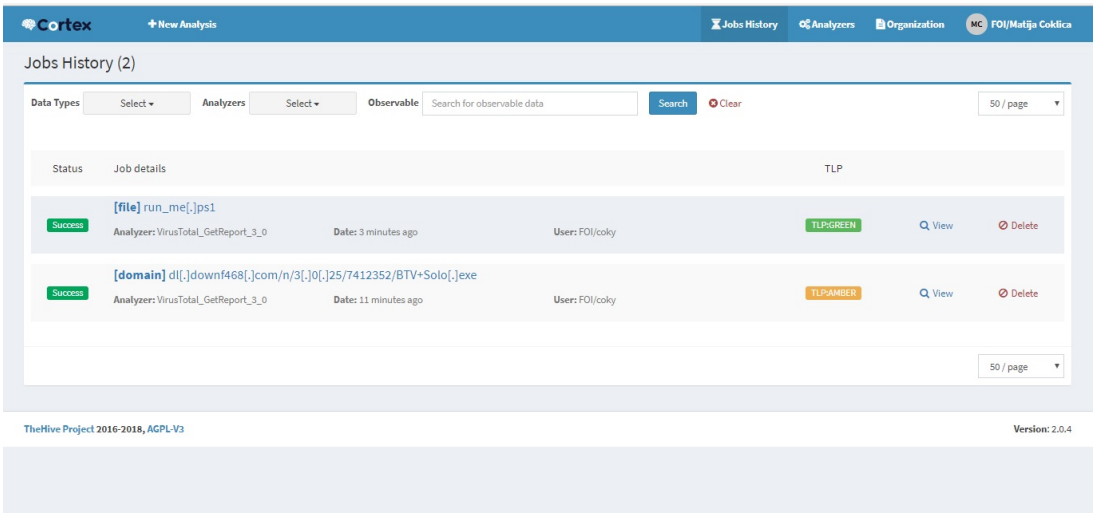
je na slici 37. Dodatne konfiguracije mogu biti upis API ključa za komunikaciju s poslužiteljem usluge, zatim postavke proxy poslužitelja, autentifikacijskih podataka, itd. Početno, nakon instalacije Cortex platforme, svi analizatori su isključeni. Korisnik ih mora ručno uključiti te po potrebi konfigurirati. Taj postupak objašnjen je u poglavlju instalacije i konfiguracije obje platforme.



Slika 37: Dio liste analizatora (autorski rad)

10.2.3. Upravljanje analizama

Ako je analizator, koji se želi koristiti za analizu, ispravno postavljen, korisnik koji ima ulogu vršitelja analize može pokrenuti analizu nad artefaktom. Obavljene analize mogu se pogledati u sekciji povijesti analiza (primjer na slici 38).



Slika 38: Povijest analiza [autorski rad]

10.3. Instalacija i konfiguracija

Oba proizvoda, TheHive i Cortex, koriste bazu podataka naziva Elasticsearch. Elasticsearch je distribuirana, RESTful baza podataka s naglaskom na brzo i efikasno traženje i analiziranje generiranih i ručno spremljenih podataka. Tehnologija iza poslužiteljske strane (eng. „Backend“) TheHive-a i Cortex-a sastoji se od programskog jezika Scala i njezinih web okvira Akka i Play. Oba web sučelja pisana su u AngularJS. Komunikacija s analizatorima vrši se preko Python programskog jezika. Cijelu tehnološku arhitekturu opisuje slika 39 sa službene GitHub stranice TheHive-Project.

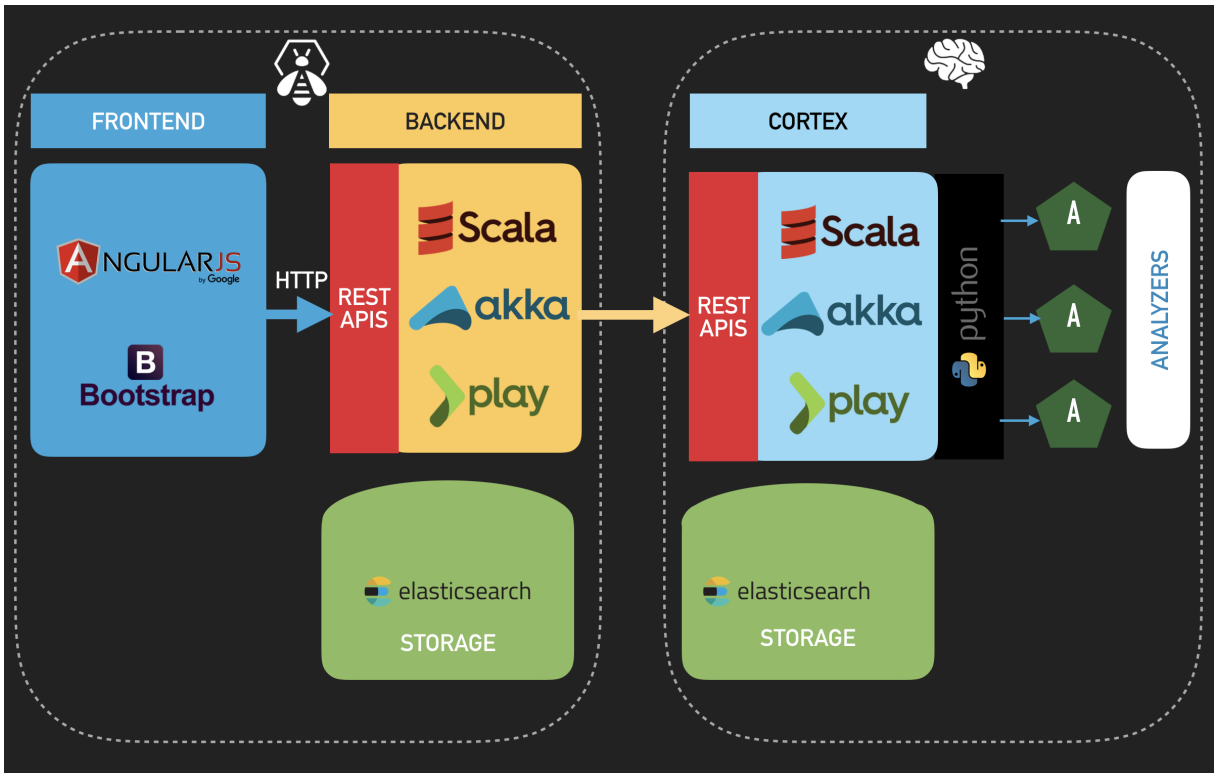
U nastavku biti će objašnjen postupak instalacije i minimalne konfiguracije TheHive 3.0.10 i Cortex 2.0.4 platforme na Ubuntu Server 16.04 Xenial Xerus. Potrebno je sa službene Ubuntu web stranice preuzeti ISO datoteku Ubuntu Server-a te ju instalirati unutar virtualnog okruženja ili fizičkog poslužitelja. Primjer jedne konfiguracije virtualnog stroja nalazi se na slici 40. Primjer postupka instalacije Ubuntu Server 16.04. operacijskog sustava može se pronaći na web stranici https://www.tecmint.com/installation-of-ubuntu-16-04-server-edition/

Kada je instalacija završila, potrebno je prijaviti se sa svojim korisničkim imenom i lozinkom na novokreiranu Ubuntu Server virtualnu mašinu. Prije instalacije Cortex-a i TheHive-a, potrebno je instalirati Elasticsearch. Prvo, potrebno je namjestiti virtualnu memoriju kako se ne bi tokom pokrenute instance Elasticsearch-a javljale „Out-of-memory“ iznimke koje bi ju ujedno i rušile. Potrebno je upisati i izvršiti sljedeću naredbu:

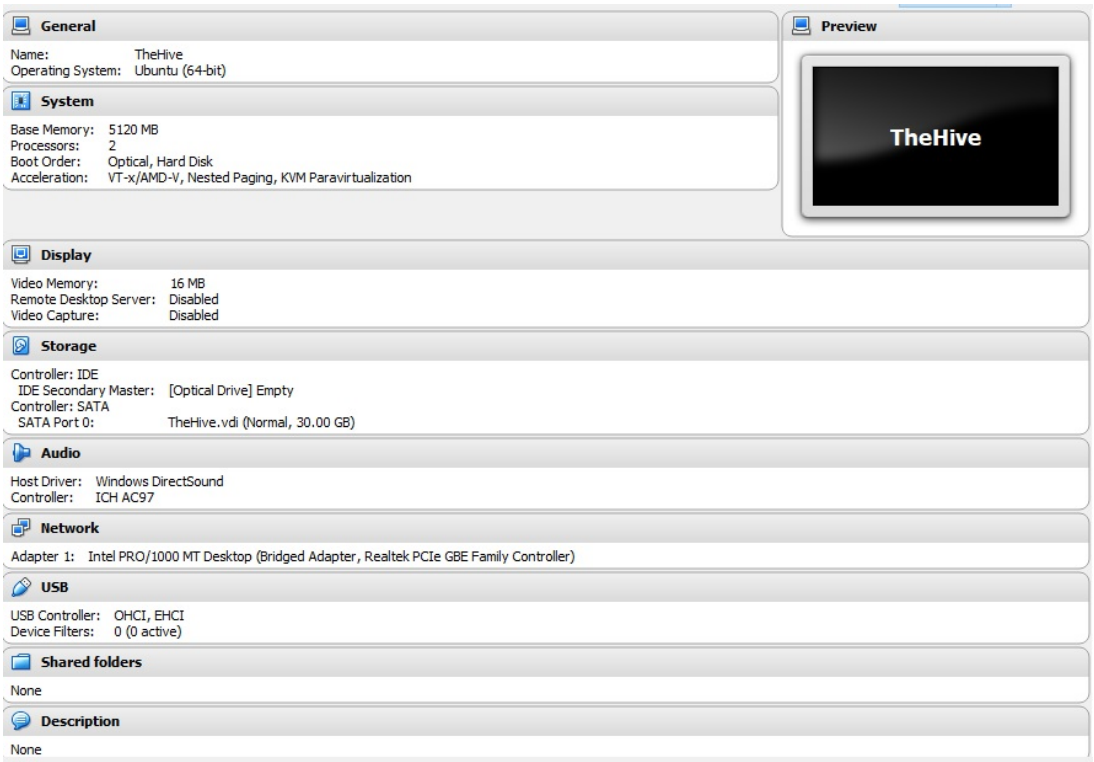
```
sudo sysctl -w vm.max_map_count=262144
```

Također, potrebno je promijeniti vm.max_map_count za stalno, na način da se unutar datoteke /etc/sysctl.conf pronađe redak sa vm.max_map_count=[cijeli_broj] te se [cijeli_broj] promjeni u 262144. Nakon toga potrebno je dohvatiti PGP ključ za instalaciju Elasticsearch-a, konfigurirati Debian repozitorij za preuzimanje, a zatim i instalirati pakete potrebne za rad Elasticsearcha:

```
sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv-key D88E42B4
```



Slika 39: Tehnološki stog TheHive + Cortex + Elasticsearch [34]



Slika 40: Primjer konfiguracije virtualnog stroja za TheHive + Cortex [autorski rad]

```
echo "deb␣https://artifacts.elastic.co/packages/5.x/apt␣stable␣main" |
sudo tee -a /etc/apt/sources.list.d/elastic-5.x.list
sudo apt install apt-transport-https
```

```
sudo apt update && sudo apt install elasticsearch
```

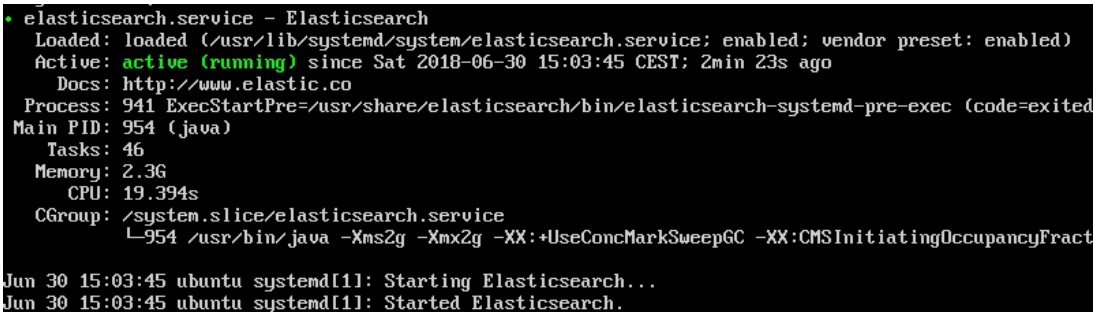
Elasticsearch nije automatski pokrenut nakon instalacije. Potrebno je najprije konfigurirati /etc/elasticsearch/elasticsearch.yml datoteku, upisom sljedećih linija na kraj:

```
network.host: 127.0.0.1
script.inline: on
cluster.name: hive
thread_pool.index.queue_size: 100000
thread_pool.search.queue_size: 100000
thread_pool.bulk.queue_size: 100000
```

Potrebno je spremiti promjene u datoteci te izvršiti sljedeće naredbe kako bi se Elasticsearch omogućio i pokrenuo kao usluga (svaki puta kada se podigne Ubuntu Server).

```
sudo systemctl enable elasticsearch.service
sudo systemctl start elasticsearch.service
```

Elasticsearch bi sada trebao ispravno raditi. Nakon nekoliko sekundi, poželjno je izvršiti naredbu sudo systemctl status elasticsearch.service kako bi se provjerilo postoje li greške u radu. Slika 41 pokazuje primjer ispravno pokrenute Elasticsearch instance.



Slika 41: Rezultat izvršavanja naredbe sudo systemctl status elasticsearch.service [autorski rad]

Zatim, potrebno je instalirati i konfigurirati Cortex. Sljedeće naredbe konfiguriraju Debian repozitorij za preuzimanje, preuzimaju PGP ključ za instalaciju, a zatim instaliraju Cortex:

```
echo 'deb https://dl.bintray.com/cert-bdf/debian any main' |
sudo tee -a /etc/apt/sources.list.d/thehive-project.list
sudo apt-key adv --keyserver hkp://pgp.mit.edu --recv-key 562CBC1C
sudo apt-get update
sudo apt-get install cortex
```

Slijedi instalacija analizatora. Potrebno je izvršiti naredbe za preuzimanje paketa potrebnih za konfiguraciju analizatora (nadogradnja Python-a i preuzimanje potrebnih modula):

```
sudo apt-get install -y --no-install-recommends python-pip python2.7-dev python3-pip
     python3-dev ssdeep libfuzzy-dev libfuzzy2 libimage-exiftool-perl libmagic1
    build-essential git libssl-dev
sudo pip install -U pip setuptools && sudo pip3 install -U pip setuptools
```

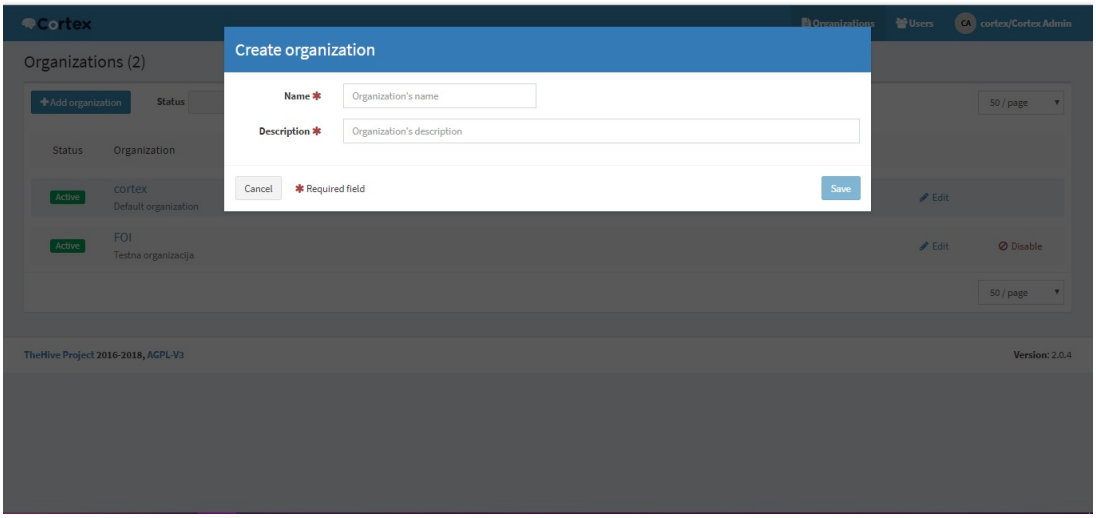
Nakon toga potrebno je preuzeti sa GitHub stranice repozitorij za analizatore, premjestiti se u preuzetu mapu te izvršiti sljedeće dvije "for" petlje koje će ispravno instalirati i konfigurirati analizatore:

```
git clone https://github.com/TheHive-Project/Cortex-Analyzers
for I in Cortex-Analyzers/analyzers/*/requirements.txt; do sudo -H pip2 install -r
    $I; done && for I in Cortex-Analyzers/analyzers/*/requirements.txt; do sudo -H
    pip3 install -r $I || true; done
```

Nakon toga potrebno je konfigurirati /etc/cortex/application.conf datoteku. Od minimalnih uvjeta, potrebno je promijeniti redak play.http.secret.key (upisati nasumičan niz brojeva te velikih i malih slova) i analyzer.path (/putanjaDoMapeSaCortexRepozitorijem/Cortex-Analyzers/analyzers). Na kraju, sljedećom naredbom potrebno je pokrenuti Cortex:

```
sudo service cortex start
```

Na adresi virtualnog stroja (pogledati naredbom ifconfig, sekcija s Internet adapterom), port 9001, nalazi se Cortex web sučelje. Prvi puta, pojaviti će se prozor sa obavijesti za ažuriranje Elasticsearch baze podataka, tj. za stvaranje i konfiguriranje indeksa. Nakon toga prikazuje se prozor za stvaranje administratorskog računa: korisničko ime, pravo puno ime i lozinka. Nakon toga, administrator se mora prijaviti te napraviti organizaciju. Slika 42 prikazuje prozor za stvaranje organizacije. Zatim potrebno je napraviti korisnika koji će biti lokalni administrator, tj. upravitelj organizacije (uloga „orgAdmin“). Također potrebno je generirati i spremiti (za kasnije) API ključ koji će se koristiti kod TheHive konfiguracije.



Slika 42: Stvaranje organizacije unutar Cortex platforme [autorski rad]

Zatim, potrebno je instalirati i konfigurirati TheHive.

```
sudo apt-get install cortex
```

Nakon toga potrebno je konfigurirati /etc/cortex/application.conf datoteku. Minimalni uvjeti su promijeniti redak play.http.secret.key (upisati nasumičan niz brojeva te velikih i malih slova) i omogućiti Cortex, pozicioniranjem na sekciju „cortex“ koja mora izgledati ovako:

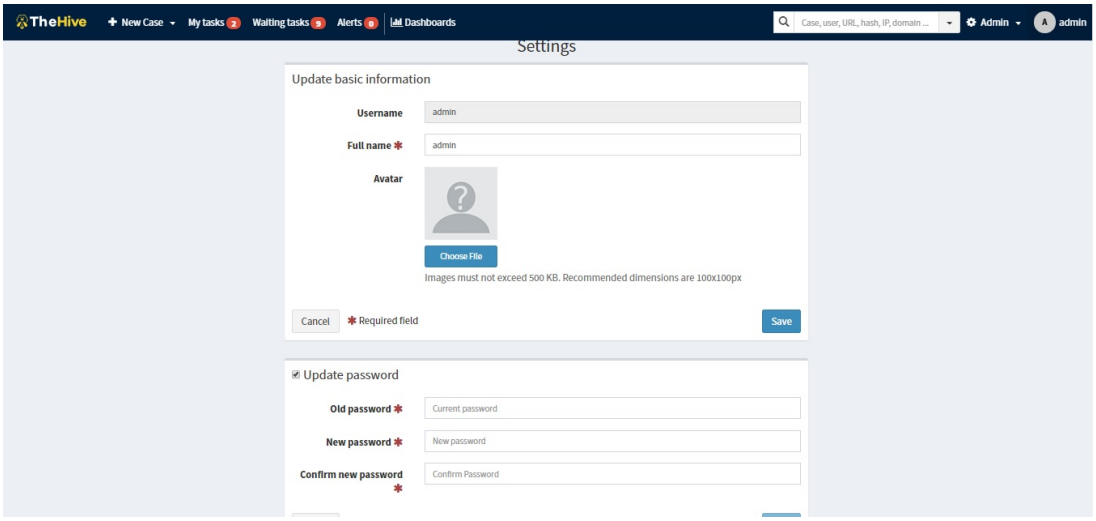
```
play.modules.enabled += connectors.cortex.CortexConnector
cortex {
  "CORTEX-SERVER-ID" {
      url = "http://127.0.0.1:9001"
```

```
        key = "API_KLJUC"
    }
}
```

Gdje je API_KLJUC generirani ključ jednog od Cortex korisnika (administratori mogu dodijeliti Cortex API ključ). Slijedi pokretanje TheHive platforme naredbom:

```
sudo service thehive start
```

Postupak ažuriranja baze i stvaranja administatora je isti kao i kod Cortex-a. Svaki od prijavljenih korisnika može pristupiti osobnim postavkama (alatna traka, prva opcija desno, primjer na slici 43). Poželjno je da korisnik nakon prve prijave provjeri te prema potrebi promijeni svoje puno ime, avatar i lozinku.



Slika 43: Primjer prozora sa detaljima korisnika unutar TheHive platforme [autorski rad]

10.4. TheHive + Cortex recenzija

U trenutku ovog pisanja (4.7.2018.), upisavši u Google tražilicu „open source incident response tools“ već kao 6. rezultat (2. ako se gledaju samo GitHub projekti) pojavljuje se službena stranica TheHive-Project tima, s njihovim proizvodima. Već kratak opis alata daje uvid u moguće funkcionalnosti. Odabirom poveznice na GitHub stranicu TheHive izvornog programskog koda, osoba može vizualnim elementima dobiti uvid u tehnološki stog i dijagram tijeka rada TheHive platforme. Budući da tijek rada nije striktno slijedan, slika koja prikazuje spomenuti dijaram može izgledati nejasno. Odabirom poveznice na GitHub stranicu TheHive dokumentacije, znatiželjan čitatelj dobije reference na razne vodiče: instalacija, administracija, konfiguracija, itd. Instalacijski vodič vrlo dobro je napisan, iako koraci instalacije ne idu potpuno slijedno. Potrebno je pročitati cijeli vodič kako bi se shvatilo koji je ispravan postupak i redoslijed instalacije. Vodiči administracije i konfiguracije također su vrlo dobro napisani, gdje je vodič administracije TheHive platforme bogat slikama, dok je vodič konfiguracije objašnjen na način da je osobi eksplicitno navedeno što (i gdje) mora namjestiti za određene/željene postavke.

Postupak instalacije je relativno jednostavan i brz (ovisi o vještinama korisnika, jačini računala...). TheHive i Cortex nisu zahtjevne platforme po pitanjima performansi te ne zahtijevaju velik broj zavisnih programa prije instalacije. Način održavanja (ažuriranje novom inačicom) je relativno lagan i dobro objašnjen unutar službene dokumentacije. Autori TheHive platforme redovito provjeravaju i odgovaraju na pitanja korisnika oko poteškoća instalacije i korištenja, ali i zahtjeva na dodavanje novih funkcionalnosti.

Što se tiče trenutno postojećih funkcionalnosti, TheHive opravdava svoje postojanje kao platforma koja je „designed to make life easier for SOCs, CSIRTs, CERTs“ [34]. Sučelje izgleda jednostavno za korištenje, nema dvosmislenih naziva akcija/opcija, laka krivulja učenja upravljanja platformom te jedna vrlo korisna opcija – brza tražilica unutar platforme. Izrada slučaja uz ponuđena ispitna polja dovoljna je za identifikaciju, analizu i dokumentiranje tih slučaja. Također, izrada slučaja iz predložaka dodatno olakšava posao te formalizira način obrade i dokumentiranja slučaja. Sloboda upravljanja slučajevima kroz zadatke i evidencije te „prisvajanje“ zadataka pojedinom korisniku daju fleksibilnost koja je potrebna kod ovakve vrste platforme. Statistike i njihove vizualizacije također su prednost TheHive-a. Svi prozori sadržavaju filtere za lakše pronalaženje relevantnijih informacija.

Glavni nedostatak TheHive platforme je zapravo nedostatak opcije generiranja izvješća. Svaki „export“ koji je ponuđen, zapravo je izvoz podataka za „backup“ ili drugu instancu TheHive platforme (datoteke su JSON formata). Vizualizacije se mogu izvesti u slikovnom formatu. Drugi nedostatak je profinjenije upravljanje „društvenim“ elementom TheHive platforme. Trenutno korisnik može samo sebe dodijeliti nekom zadatku. Poželjno bi bilo da određen korisnik (npr. onaj koji je napravio slučaj) ili određena rola (npr. neka podvrsta administratora) mogu dodjeljivati druge korisnike zadacima. Zadnji (prema subjektivnom mišljenju) nedostatak je nemogućnost stvaranja predložaka zadataka, jednog ili u grupi. Trenutno, zadaci se mogu predefinirati u predlošku slučaja ili dodavati ručno jedan po jedan u prozoru trenutnog slučaja.

Cortex, kao zasebna platforma, također je jednostavna za upravljanje i lagana za naučiti. Ono što je potrebno izdvojiti je to da pruža lagan način konfiguracije službenih analizatora. Cortex daje zajedničko sučelje za postavljanje artefakata na analizu s jednim ili više analizatora istovremeno. To je ukratko glavna i jedina funkcionalnost Cortex-a. Zato je vrlo korisna stvar koristiti ga kao „modul“, tj. komunicirati sa njegovim API-jem iz TheHive platforme. Budući da je Cortex tako skromna platforma, koja dobro odrađuje posao za koji je stvorena, nisu pronađeni nedostaci.

10.5. TheHive alternative

SOC i CSIRT trebaju jednostavna i robusna web sučelja u kojima mogu vidjeti anomalije unutar organizacije koju pokrivaju, budući da SIEM prikuplja i prikazuje sve zapise i događaje. Također, tim anomalijama bi se trebalo upravljati kroz kolaboraciju članova stručnog tima uz bilježenje cijelog postupka rukovanja. TheHive nudi točno to, ali nudi i korak dalje - automatiziranu analizu (zahvaljujući dodatnoj usluzi - Cortex), vizualizacije vezane uz kolaboraciju, upravljanje događajima/incidentima sa ili bez alarma kao početnog sadržaja. Čini se teškim pronaći alter-

native koje nude sve ove funkcionalnosti, no ipak, razvijeno je nekoliko sličnih platformi: Fast Incident Response, Sandia Cyber Omni Tracker, RT for Incident Response, CrowdStrike Falcon Orchestrator.

10.5.1. Fast Incident Response

Kao alternativa TheHive platformi, Fast Incident Response (skraćeno FIR) nudi iste funkcionalnosti, ali u drugačijem „ruhu“. Otvorenog je programskog koda pod licencom „GNU General Public License v3.0“, razvijen od strane CERT Société Générale razvojnog tima. Platforma ne sadrži integraciju sa uslugama poput Cortex platforme, no slučajevima se upravlja uz dodatne parametre. Tako slučajevi unutar FIR-a mogu bili događaji ili incidenti, odnosno ovo dvoje se međusobno razlikuje i drugačije kategorizira. Funkcionalnosti:

- Stvaranje događaja (eng. „events“): Glavni element FIR-a, opisan naslovom, zahvaćenim entitetima (eng. „bussines line“), kategorijom/vrstom događaja, statusom, načinom detektiranja, prioritetom/ozbiljnošću, datumom, razinom povjerljivosti, klasifikacijom (incident ili samo događaj), osobom ili entitetom koja je nadležna za upravljanje ako je događaj prepoznat kao incident, planom upravljanja i dodatnom oznakom važnosti incidenta. Prema terminologiji FIR-a, incident je onaj događaj nad kojim je potrebno poduzeti akciju (faze upravljanja incidentom). Događaji i incidenti mogu se pregledavati u zajedničkim ili odvojenim sekcijama web sučelja.
- Uređivanje događaja: Uz prijašnje navedena svojstva, događajima se nakon stvaranja mogu još definirati i komentari, datoteke (iz kojih nastaju artefakti), zadaci (eng. „to-do“), te pronalasci vezani uz događaj (eng. „nuggets“). Svaki događaj može se pregledati kao izvješće. Na stranici uređivanja pojedinog događaja nalaze se prečaci za slanje maila (preko konfiguriranog dodatka).
- Statistika: Rezultati rada mogu biti predočeni tablicama i grafovima unutar tromjesečnog ili godišnjeg razmaka, zatim postoje grafovi za usporedbu tekuće i prošle godine te tablice za prikaz velikih incidenata. Tu si i visoko prilagodljivi statistički prikazi događaja (po mjesecima, određenim kategorijama, detekciji, prioritetu i entitetima) te grafički prikaz incidenata prema atributima.
- Upravljanje korisničkom pločom (svi): Svaki prijavljeni korisnik može vidjeti svoje korisničko ime te vidjeti i mijenjati pravo ime, e-mail adresu, paginaciju (broj incidenata po stranici na radnoj ploči) i API ključ . Može se mijenjati i lozinka, gdje se komunikacija sa poslužiteljem automatski prebacuje na protokol https (ako nije već prije u postavkama postavljeno da je svaka komunikacija/radnja kriptirana).
- Upravljanje korisničkom pločom (admin): Administrator ima pregršt opcija stvaranja i mijenjanja gotovo svih elemenata (koji se vežu uz korisnike FIR-a i događaje). Opcije su abecedno sortirane i kategorizirane prema uređivanju autorizacijskih elemenata, događaja, dodataka i stranica. Također, postoji sekcija u kojem su prikazani događaji nedavnih uređivanja na administratorskoj stranici. Prema početnim postavkama FIR-a,

na administratorskoj ploči mogu se stvarati i mijenjati sljedeći elementi: API tokeni, grupe, korisnici, predlošci kategorija i primalaca (kod slanja maila), artefakti (uglavnom su to sažeci datoteka), datoteke, pronalasci vezani uz događaj, zadaci i njihovi predlošci, atributi (brojčani, slični metrikama, vezani uz pojedini događaj), entiteti, komentari, kategorije i predlošci događaja, oznake, datoteke zapisa, profili i stranice.

FIR je pisan u Python skriptnom jeziku te koristi Django web okvir. Aplikacija je smještena na Nginx web poslužitelju. Korisničko sučelje koristi Javascript, Ajax, d3.js i Bootstrap. Opisi platforme, instalacije i rukovanja FIR platformom, nalaze se unutar GitHub službene stranice FIR korisničkog priručnika: https://github.com/certsocietegenerale/FIR/wiki/User-Manual

10.5.2. Sandia Cyber Omni Tracker

Sljedeća alternativa TheHive platformi je Sandia Cyber Omni Tracker (skraćeno SCOT). Ova alternativa također dijeli funkcionalnosti sa TheHive platformom, uz pojačanu komponentu slijedne strukture rada. Otvorenog je programskog koda, pod licencom „Apache License 2.0“, a platformu je razvila „Sandia Corporation“. Naime, alarmi dolaze u SCOT sa raznih izvora (najčešće SIEM) preko e-maila ili REST API-ja. Analizirajući alarme, sigurnosni stručnjaci mogu promovirati alarm u događaj, ako smatraju da je alarm potrebno dodatno analizirati. Događajima se mogu dodavati bilješke, datoteke i razni metapodaci (poput TLP-a). Bilješke se mogu pretvoriti u zadatke za određene korisnike, gdje ujedno i počinje kolaboracija ljudi na događaju. Ako je procjeni potrebnim, događaj se može promovirati u incident. Incidentu se može dodati više metapodataka (poput vrste, ozbiljnosti, datuma otkrivanja, itd.) te se mogu definirati detalji kompromitiranog sustava, akcije potrebne za oporavak od incidenta, detalji detekcije, itd. Unutar svake faze (alarm, događaj, incident), SCOT automatski analizira svaku upisanu domenu, naziv datoteke sa ekstenzijom, sažetke, IP adrese, e-mail adrese, zemljopisne duljine i širine, ranjivosti (CVE) te CIDR zapise. Na njih se može kliknuti te tako dobiti informacija o tome je li taj element pronađen u „Intel“ sekciji SCOT-a (jer SCOT dopušta praćenje raznih javnih izvora o indikatorima kompromisa) zatim unutar integriranih sigurnosnih web usluga te ostalim bilješkama unutar cijele platforme. Krivulja učenja korištenja SCOT platforme veća je nego kod TheHive platforme, ne samo jer postoji više opcija i elemenata platforme sa kojima se radi, nego i sučelje ne izgleda intuitivno za rad. Ukratko, funkcionalnosti:

- Sakupljanje i upravljanje alarmima.
- Upravljanje događajima.
- Upravljanje incidentima.
- Dodavanje i praćenje izvora o indikatorima kompromisa.
- Upravljanje zadacima.
- Stvaranje vodiča za upravljanje i odziv specifičnih incidenata.
- Integracija s IDS platformama u svrhu migracije generiranih potpisa.

- Vizualizacije statistika.
- Automatsko ekstrahiranje datoteka, IP i e-mail adresa, ranjivosti te ostalih entiteta uz mogućnost njihovog pregleda.
- Stvaranje izvještaja na temelju vizualizacija.
- Vođenje rada sigurnosnih stručnjaka kroz ugrađeni kalendar.
- Administratorska ploča: upravljanje korisnicima, grupama, datoteka zapisa, API ključevima te izbrisanim/blokiranim objektima i entitetima.

SCOT je pisan u Perl skriptnom jeziku. Za svoj rad koristi MongoDB i Elasticsearch kao baze podataka te ActiveMQ kao broker za poruke. Web sučelje pisano je u ReactJS okviru. Dokumentacija detaljno opisuje svaki element SCOT platforme, od instalacije do korištenja svake funkcionalnosti, a nalazi se na Read the Docs službenim stranicama SCOT platforme: http://scot.readthedocs.io/en/latest/index.html

10.5.3. Request Tracker for Incident Response

Sustav za upravljanje ulaznicama (eng. „tickets“) sa dodanim funkcionalnostima za upravljanje sigurnosnim incidentima nazvan Request Tracker for Incident Response (skraćeno RTIR), još je jedan od alternativa TheHive platforme. Otvorenog je programskog koda (postoji i komercijalna inačica) pod licencom „GNU General Public License v2.0“, a proširenje Request Tracker platforme, kao i samu platformu, razvio je Best Practical Solutions razvojni tim. Nadalje u tekstu, RTIR će se neće spominjati kao proširenje, već kao nezavisna platforma. Fleksibilan tok rada i jednostavno, ali detaljno web sučelje, glavni su razlozi zašto je ovaj alat dobra alternativa za platforme koje orkestriraju SOC, CSIRT ili CERT timovima. Ipak, zbog svoje komplicirane instalacije i nadogradnje, nije uzet kao praktična komponenta ovog diplomskog rada. Tok rada započinje od strane korisnika (najčešće običnog zaposlenika) koji je otkrio naizgled sumnjivu radnju. Taj korisnik stvara ulaznicu definiranjem naslova, opcionalnim priloženim datotekama te opisom anomalije koju je primijetio. Slanjem ulaznice u red čekanja, netko iz sigurnosnog tima (eng. „staff user“, u nastavku osoblje) može vidjeti poslanu ulaznicu i preuzeti je na sebe, dobivajući time pravo pogleda na sve informacije ulaznice te opciju stvaranja incidenta. Odluči li osoblje stvoriti incident temeljem ulaznice, otvaraju se nove funkcionalnosti poput vođenja istrage ili definiranja protumjera. Obje funkcionalnosti su još jedno slanje ulaznice, no ovaj puta slanje određenoj osobi koja može pomoći u rješavanju istrage, odnosno u provođenju protumjera. Kod svakog stvaranja ulaznice, pošiljalac može stvoriti nova svojstva i popuniti ih informacijama za koje smatra da su od pomoći primaocu ulaznice. Glavna zadaća osoblja je ažuriranje statusa ulaznice, odnosno incidenta, temeljem aktivnosti koje se provode ili su već gotove. Ukratko, funkcionalnosti RTIR platforme bile bi:

- Stvaranje i pregled početne ulaznice.
- Korelacija podataka između svih ulaznica.

- Upravljanje ulaznicom od strane osoblja (preuzimanje ulaznice kao vlastitog zadatka, ažuriranje statusa, dodavanje novih informacija).
- Stvaranje i upravljanje incidentom.
- Stvaranje i pregled ulaznice istrage.
- Stvaranje i pregled ulaznice protumjere.
- Ažuriranje statusa i informacija incidenta.
- Globalna pretraga informacija unutar ulaznica.

RTIR je pisan u Perl skriptnom jeziku. Kao web poslužitelj, može koristiti bilo koji poslužitelj sa FastCGI komponentom (Apache, Lighttpd, Nginx. . .), a kao bazu podataka, može koristiti MySQL, MariaDB, PostgreSQL te OracleDB. Uz ovakvu fleksibilnost dolazi i cijena, a to je kompleksna instalacija ovisnih paketa (Perl moduli) sa vlastitim konfiguracijama za web poslužitelj/bazu podataka. Dokumentacija za instalaciju, u usporedbi sa dokumentacijom za upotrebu platforme, jako je skromna. Općenito, dokumentacija za RTIR samo je mali dio dokumentacije za RT, koja je preopširna za početnika. Web adrese dokumentacija:
https://docs.bestpractical.com/rt/4.4.3/index.html i
https://docs.bestpractical.com/rtir/4.0.1/index.html

10.5.4. CrowdStrike Falcon Orchestrator

Radi li se o Windows Server platformama, još jedna zanimljiva alternativa je CrowdStrike Falcon Orchestrator, koji je proširiva Windows aplikacija sa funkcionalnostima poput automatizacije tijeka rada, upravljanje „whitelistama“, softverskom imovinom te praćenjem alarma. Otvorenog je programskog koda, pod licencom „Affero GPLv3 license“, a aplikaciju je razvio Evan Burns. CrowdStrike upozorava da ovaj projekt nije unutar njihove izravne nadležnosti, nego je vlasništvo zajednice koja usko surađuje sa njima. Ipak, CrowdStrike Falcon Orchestrator koristi API za integriranje CrowdStrike podataka unutar opreme koje se pokreće na Windows OS-u. Funkcionalnosti:

- Upravljanje alarmima: CrowdStrike Falcon Orchestrator koristi CrowdStrike Falcon Endpoint Protection (komercijalan softver) API za dobivanje alarma. Svaki alarm ima varijabilan broj podataka koji su podijeljeni u dvije skupine: podaci o pronalasku i podaci o imovini/opremi gdje je anomalija detektirana. Moguće je svakom alarmu dodijeliti osobu (eng. „responder“), status, IP adresu, ozbiljnost, oznake i komentare.
- Automatizirano upravljanje odgovornostima: CrowdStrike Falcon Orchestrator će osobama prema zadanom redoslijedu i postupku dodjeljivati zadatke/odgovornosti (npr. odziv na specifičan alarm).
- Upravljanje taksonomijama i njihova vizualizacija. Moguće je postavljati pravila prema kojima će se alarmi prepoznati na vizualizacijskoj ploči. Pravila su određena poljem i njezinom vrijednošću.

- „Whitelistanje“: postupak koji definira određene događaje kao „šum“, tj. ono što se prikazuje i pojavljuje kao alarm, no u svojoj suštini zapravo nije nikakva anomalija.
- Blokiranje domenskih računa: Ako alarm ima podatak o korisniku (iz Active Directory baze podataka), mijenjanjem statusa alarma radi sprječavanja daljnjeg rizika/štete (eng. „containment“) može se pokrenuti proces blokiranja tog korisničkog računa i/ili prisilnog mijenjanja lozinke.
- Obavljanje forenzike: Ekstrakcija datoteka, preglednik datotečnog sustava, instalirani softverski paketi, itd.

CrowdStrike Falcon Orchestrator koristi tehnologije poput .NET 4.5, C#, ASP.NET MVC 4, Entity Okivra i PowerShell skriptnog jezika. Za svoje potrebe koristi MS SQL bazu podataka. Dokumentacija detaljno opisuje instalaciju, ali samo nekoliko funkcionalnosti. Dokumentacija se nalazi za GitHub Wiki stranici CrowdStrike Falcon Orchestrator aplikacije na adresi: https://github.com/CrowdStrike/falcon-orchestrator/wiki

11. Kolide Fleet + Osquery

Osquery je instrumentacijski okvir za operacijske sustave Windows, Linux, FreeBSD i OS X (macOS). Nema grafičko sučelje, nego se koristi kroz komandnu liniju (interaktivno) ili se konfigurira kao pozadinski proces (eng. „daemon“). Otvorenog je programskog koda pod licencama „Apache 2.0“ i „GPLv2“, a okvir je razvio Facebook Open Source razvojni tim. Osquery omogućuje svojem korisniku slanje upita u operacijski sustav (u kojem djeluje instalirana Osquery instanca), gdje se operacijski sustav „ponaša“ kao relacijska baza podataka. Upiti su SQL formata, koristeći klauzule SELECT-FROM-WHERE.

Kolide Fleet jedna je od platformi za upravljanje Osquery instanci kao „flote“ distribuiranih agenata. Omogućuje, kroz grafičko sučelje, grupiranje agenata, pojedinačno ili grupno pisanje upita prema agentima te stvaranje ponavljajućih upita. Otvorenog je programskog koda pod „MIT License“ licencom, a platformu je razvio Kolide razvojni tim.

11.1. Osquery funkcionalnosti

Unutar ovog poglavlja slijedi opis svih funkcionalnosti koje je autor prepoznao unutar rada Osquery-a (prikazane će biti autorove slike, primjeri upita...) te uz pomoć službene dokumentacije sa Read the Docs stranice Osquery okvira na adresi:
https://osquery.readthedocs.io/en/stable/

11.1.1. Interaktivno pisanje i pokretanje upita

Osquery sa instalacijom pruža korištenje interaktivne konzole za pisanje upita – „osqueryi“. Pokretanjem osqueryi konzole korisniku se na izbor, uz pisanje i izvršavanje upita, daju opcije promjene načina ispisa, skraćenice, zamjene praznih vrijednosti i delimitera, itd. Rezultati upita se ne spremaju (ni u bazu ni u datoteke zapisa) nego se odmah prikazuju korisniku u konzoli, kao što je primjer na slici 44.

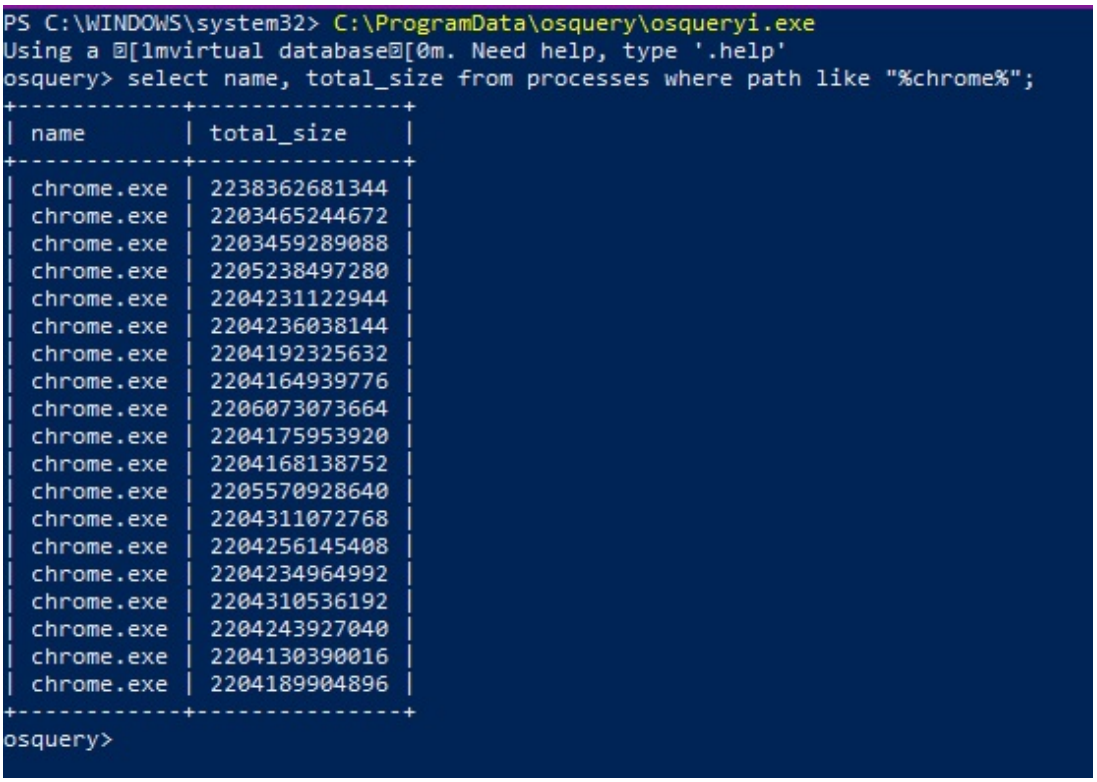
Također, moguće je poslati upit kao tekst u osqueryi, odnosno poslati tekst prema cijevi (eng. „pipe“) standardnog ulaza (eng. „stdin“) za osqueryi. Primjer naredbe:

```
echo "SELECT * FROM routes WHERE destination = '::1';" | osqueryi
```

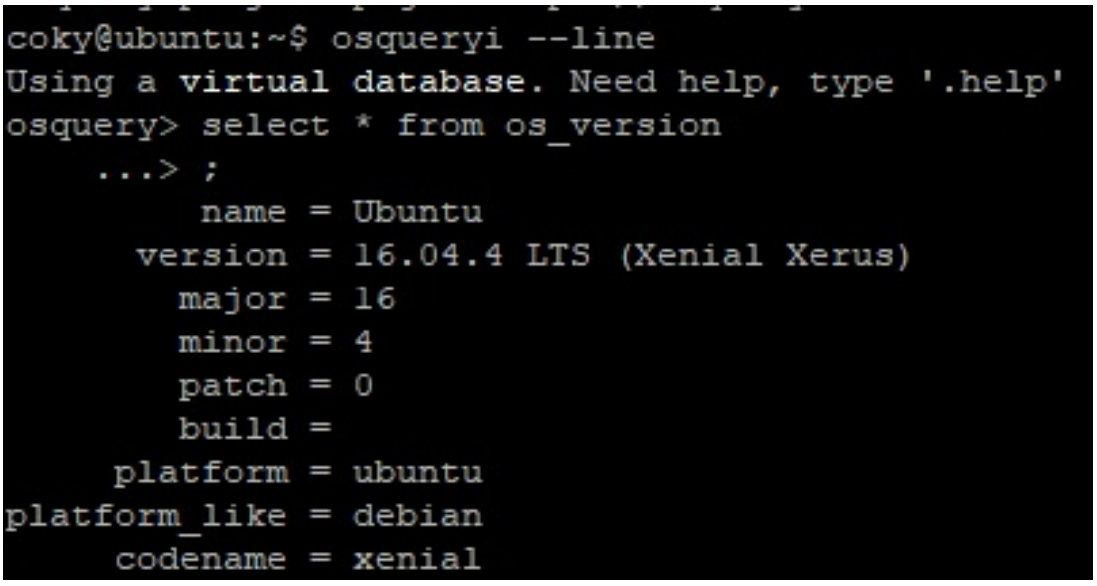
Sljedeća zanimljiva značajka, koja je prije spomenuta, je način ispisa rezultata. Slika 45 prikazuje primjer linijskog ispisa rezultata upita, gdje je način ispisa definiran već kod poziva osqueryi konzole (a ne unutar nje, što je također moguće).

11.1.2. Konfiguriranje automatskog pokretanja upita

Pozadinski proces, za nadgledanje računala na kojem je pokrenut, naziva se osqueryd i omogućuje planiranje izvršavanja upita i zapisivanje promjena stanja nakon izvršavanja upita. Taj pozadinski proces sakuplja rezultate upita kroz vrijeme i generira zapise unutar određene



Slika 44: Primjer pokretanja osqueryi konzole na Windows platformi i izvršavanje upita nad tablicom "processes" [autorski rad]



Slika 45: Primjer pokretanja osqueryi konzole (sa načinom rada linijskog ispisa rezultata) na Linux platformi i izvršavanje upita nad tablicom "os_version" [autorski rad]

datoteke koji predstavljaju promjenu stanja. Ono što omogućuje ovakvo ponašanje osqueryd-a je korištenje API-ja operacijskog sustava za slanje događaja (promjena stanja) i perzistentne baze podataka RocksDB (također proizvod Facebook razvojnog tima) temeljene na zapisima ključ-vrijednost koje ostaju sve do isteka vremenskog ciklusa intervala, jer se tada sve promjene upisuju u datoteku zapisa.

Osqueryd pruža puno više opcija za konfiguraciju, odnosno njih čak stotninjak. Općenito, to su opcije za upravljanje konfiguracijskom datotekom, nastavcima, TLS klijentska i poslužiteljska konfiguracija, dodatni audit, AWS konfiguracija, zatim Kafka, Syslog, Watchdog konfiguracija... Određene opcije koristi Kolide Fleet, o kojem će biti spomena kasnije. Konfiguracijska datoteka Osquery okvira (naziva osquery.conf) temelj je osqueryd pozadinskog procesa kod pokretanja, jer navodi sljedeće opcije:

Sekcija „options“ koja predstavlja generalne opcije za pozadinski proces:

- „config_plugin“: početna vrijednost je „filesystem“, što znači doslovno čitanje podataka konfiguracije čistog tekstualnog formata. Druga opcija može biti „tls“, koja uključuje HTTP/TLS zahtjeve kod čitanja konfiguracije. „Plugin“ i podaci za odgovor moraju biti JSON formata.
- „logger_plugin“: početna vrijednost je „filesystem“, što znači doslovno pisanje rezultata upita kao čisti tekstualni format. Druge opcije mogu biti tls, syslog, windows_event_log, kinesis, firehose, kafka_producer.
- „logger_path“: mapa (cijela putanja) u koju se spremaju informacije, upozorenja i greške tokom rada pozadinskog procesa. Ako je „logger_plugin“ jednak „filesystem“ vrijednosti, tada se u toj istoj mapi i spremaju rezultati automatskih upita.
- „disable_logging“: ako je „true“, tada se ne spremaju informacije, upozorenja i greške tokom rada pozadinskog procesa.
- „schedule_splay_percent“: minimalni postotak vremenskog odstupanja između svakog automatskog upita. Korisno ako postoji puno upita koji imaju isti vremenski interval okidanja.
- „database_path“: putanja na bazu podataka koja služi za spremanje događaja i razlike u rezultatima upita.
- „disable_tables“: tablice nad kojima se ne smiju postavljati upiti.

Sekcija „schedule“ koja predstavlja upite čiji se rezultati upisuju nakon određenog vremenskog intervala te se spremaju u „logger_path“. Svaki upit zasebna je sekcija, gdje je potrebno definirati:

- „query“: upit koji će se pokretati.
- „interval“: broj koji predstavlja trajanje jednog vremenskog ciklusa, odnosno koliko sekundi mora proći do ponovnog upisa u datoteku zapisa.

Zašto „interval“ ne predstavlja vrijeme koje je potrebno do ponovnog okidanja i spremanja rezultata upita? Recimo da postoji scenarij u kojem osqueryd izvršava upit o svim priključenim vanjskim podatkovnim uređajima svakih 60 sekundi. Prvi puta, upit pronalazi 2 uređaja (broj 2 je uzet kao primjer). Ako unutar 60 sekundi korisnik računala priključi USB te ga unutar istih 60 sekundi razdvoji, sljedeći rezultat upita bit će ponovo ona 2 ista uređaja, što nije ispravno.

U stvarnosti, kako bi se „registrirao“ ovaj treći USB, osqueryd koristi API za događaje unutar operacijskog sustava te svaki događaj vezan uz upit evidentira kao promjenu i sprema ju u bazu podataka osquery-a (RocksDB). Tako će, dok prođe 60 sekundi, datoteka zapisa imati 3, umjesto samo 2 zapisa.

Sekcija „decorators“ definira upite koji se okidaju kod svakog drugog upita te se podaci (redovi i stupci unutar SELECT) pridjeljuju rezultatima tih drugih upita.

Sekcija „packs“ definira kolekciju upita. Moguće je definirati „packs“ preko putanje (kao što je na slici 46) ili preko JSON formata unutar kojeg se nalaze upiti. Kolekcije upita stvaraju se u svrhu određene istraživačke logike. Na primjer, pregled „svih“ malicioznih radnji može biti zapisan u jednoj posebnoj .conf datoteci i pozvan od strane osquery.conf datoteke kao „pack“. Jedna zanimljiva značajka kolekcije upita je definiranje uvjetnih upita. U tom slučaju dodaje se novo JSON polje „discovery“ te se unutar njega definiraju upiti koji, ako rezultat njihova izvršavanja postoji, okidaju „normalne“ upite unutar „queries“ sekcije. Primjer se nalazi na slici 47.

```
59  // Linux:        /usr/share/osquery/packs
60  // OS X:         /var/osquery/packs
61  // Homebrew:     /usr/local/share/osquery/packs
62  // make install: {PREFIX}/share/osquery/packs
63  //
64  "packs": {
65    // "osquery-monitoring": "/usr/share/osquery/packs/osquery-monitoring.conf",
66    // "incident-response": "/usr/share/osquery/packs/incident-response.conf",
67    // "it-compliance": "/usr/share/osquery/packs/it-compliance.conf",
68    // "osx-attacks": "/usr/share/osquery/packs/osx-attacks.conf",
69    // "vuln-management": "/usr/share/osquery/packs/vuln-management.conf",
70    // "hardware-monitoring": "/usr/share/osquery/packs/hardware-monitoring.conf",
71    // "ossec-rootkit": "/usr/share/osquery/packs/ossec-rootkit.conf",
72     "windows-hardening": "C:\\ProgramData\\osquery\\packs\\windows-hardening.conf",
73     "windows-attacks": "C:\\ProgramData\\osquery\\packs\\windows-attacks.conf"
74  },
75
76  // Provides feature vectors for osquery to leverage in simple statistical
```

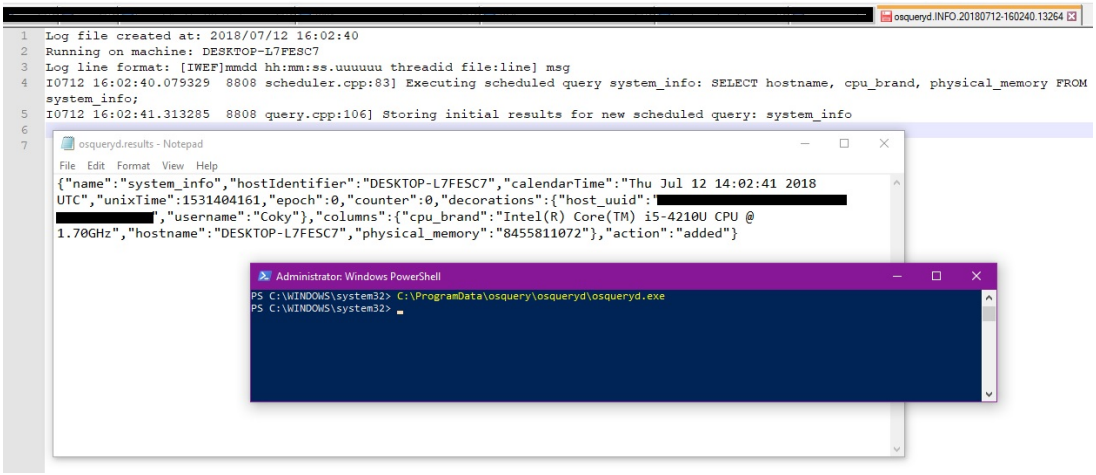
Slika 46: Dio konfiguracijske datoteke Osquery okvira za kolekcije upita. [autorski rad]

Kod ispunjenih konfiguracijskih parametara „logger_plugin“ i „logger_path“ rezultati automatskih upita zapisani su na određeni medij. Pa tako sa „filesystem“ kao „logger_plugin“ korisnik može, na računalu gdje je pokrenut osqueryd, pronaći datoteku zapisa na putanji koja je definirana u „logger_path“. Na slici 48 je primjer rezultata izvršenog upita iz „scheduled“ sekcije konfiguracijske datoteke, tj. upita čiji se rezultati pojavljuju unutar datoteke prema određenom vremenskom intervalu. Vidljivo je da su u datoteku „osqueryd.results“ upisani rezultati JSON formata jednog izvršavanja upita, dok su u automatski generiranu datoteku (zbog vrijednosti "filesystem" unutar „logger_plugin“ parametra) upisane informacije i upozorenja tokom rada pozadniskog procesa, koji je u ovom primjeru pokrenut unutar Powershell konzole.

```
"packs": {
  // "osquery-monitoring": "/usr/share/osquery/packs/osquery-monitoring.conf",
  // "incident-response": "/usr/share/osquery/packs/incident-response.conf",
  // "it-compliance": "/usr/share/osquery/packs/it-compliance.conf",
  // "osx-attacks": "/usr/share/osquery/packs/osx-attacks.conf",
  // "vuln-management": "/usr/share/osquery/packs/vuln-management.conf",
  // "hardware-monitoring": "/usr/share/osquery/packs/hardware-monitoring.conf",
  // "ossec-rootkit": "/usr/share/osquery/packs/ossec-rootkit.conf",
  // "windows-hardening": "C:\\ProgramData\\osquery\\packs\\windows-hardening.conf",
  // "windows-attacks": "C:\\ProgramData\\osquery\\packs\\windows-attacks.conf"
  "chrome-discovery": {
      "discovery": [
          "SELECT pid FROM processes WHERE name = 'chrome.exe';"
      ],
      "queries": {
          "active_directory": {
            "query": "SELECT cmdline FROM processes WHERE name = 'chrome.exe';",
            "interval": "10",
            "description": "Check arguments given on chrome.exe run."
          }
      }
  }
},

// Provides feature vectors for osquery to leverage in simple statistical
// analysis of results data.
//
```

Slika 47: Dio konfiguracijske datoteke Osquery okvira za kolekcije upita sa primjerom kolekcije koja ima uvjetni "discovery" upit. [autorski rad]



Slika 48: Primjer zapisanog rezultata automatskog upita te poruka o radu pozadniskog procesa osqueryd [autorski rad]

11.2. Kolide Fleet funkcionalnosti

Unutar ovog poglavlja slijedi opis svih funkcionalnosti koje je autor prepoznao unutar rada Kolide Fleet-a (prikazane će biti autorove slike, primjeri upita...) te uz pomoć dokumentacije sa Github stranice Kolide Fleet platforme na adresi: https://github.com/kolide/fleet#using-fleet

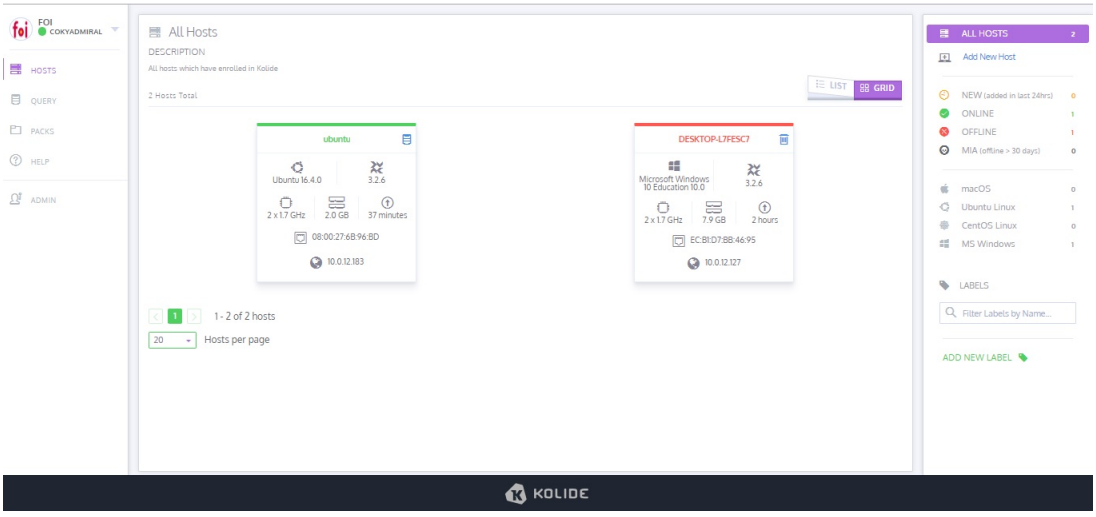
11.2.1. Upravljanje agentima

Kolide Fleet (u nastavku KF) vidi računala sa instaliranim Osquery okvirom kao agente. Želi li računalo postati agent pod kontrolom KF-a, ono mora pokrenuti pozadinski proces osqu-

eryd sa „tls“ vrijednostima određenih konfiguracijskih parametara, zatim korištenjem certifikata izdanog od strane KF-a te upisom dodatne „tajne“ koju također generira sam KF. Nakon uspješne prijave korisnika koji smije dodavati agente, potrebno je sa desne strane početnog prozora odabrati „Add New Host“. Potrebno je preuzeti certifikat klikom na „Fetch Kolide Certificate“ te kopirati „Osquery Enroll Secret“ klikom na ikonu odmah ispod „Reveal Secret“ opcije. Više o korištenju certifikata i „tajne“ kod pokretanja agenta u poglavlju od instalaciji i konfiguraciji KF platforme.

Na alatnoj traci s lijeve strane, ispod padajućeg izbornika za korisnika, prva opcija je „HOSTS“ što označuje početni prozor, tj. prozor za upravljanje agentima. Ono što je vidljivo kod tog početnog prozora KF-a su prozori agenata u obliku sličica ili redaka (ovisno o načinu prikaza agenta, tj. „list“ ili „grid“). Svaki taj prozor agenta prikazuje naziv i inačicu operacijskog sustava, inačicu instaliranog Osquery okvira, broj jezgri i frekvenciju rada procesora, RAM kapacitet, vremenski period rada, MAC adresu te IP adresu računala. Ako je agent „podignut“, tj. ako je pozadinski proces na tom računalu pokrenut, tada je vidljiva plava ikona za prebacivanje prozora za stvaranje i izvršavanje upita. Ako pak to nije slučaj, tada je vidljiva ikona brisanja podataka o agentu.

S desne strane nalazi se svojevrsna statusna traka. Iznad nje se nalazi spomenuta opcija za dodavanje novog agenta, dok se ispod nalazi opcija za stvaranje nove oznake, tj. pojma prema kojem se mogu grupirati određeni agenti. Statusna traka sastoji se od prikaza koliko je agenata dodano u vremenskom periodu od 24 sata, koliko agenata radi, a koliko ne, zatim koliko je agenata označeno kao MIA (agenti koji ne rade duže od 30 dana) te broj agenata kategoriziranih po operacijskom sustavu. Klikom na jedan od statusnih redaka, automatski se filtrira prikaz prozora agenata u središtu početnog prozora. Također, moguće je filtrirati upisom željenih oznaka. Sve ovo vidljivo je na slici 49.



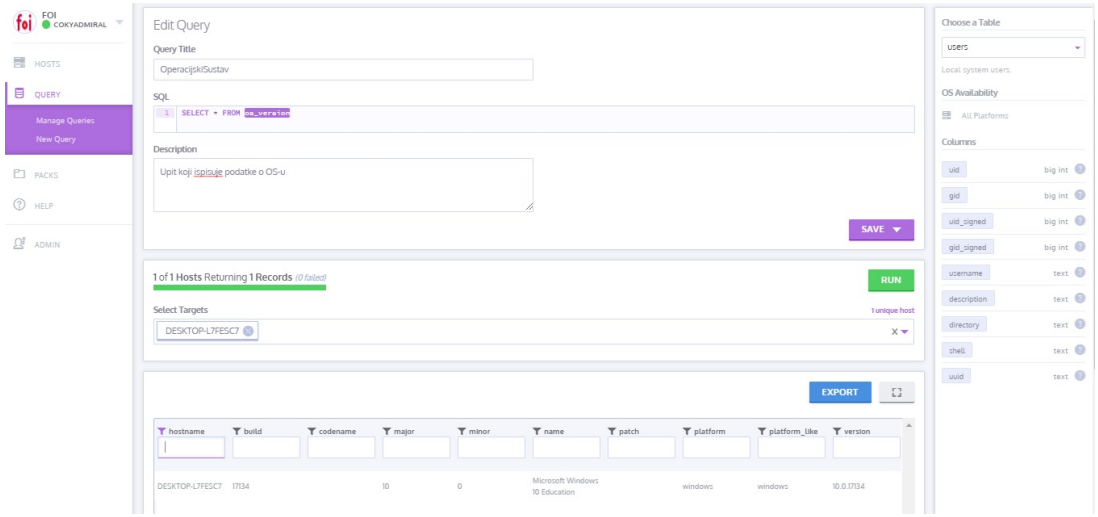
Slika 49: Primjer počenog prozora Kolide Fleet platforme sa dva "registrirana" agenta [autorski rad]

11.2.2. Postavljanje upita prema agentima

Klikom na „QUERY“ na alatnoj traci, otvara se prozor sa popisom svih napravljenih upita. Svaki upit unutar liste ima svoj naziv, autora i datum zadnje izmjene. Klikom na upit, s desne strane se pokazuje traka unutar koje se taj upit može uređivati ili pokrenuti, vidjeti kako izgledaju upit i njegov opis te vidjeti kojim sve kolekcijama upita pripada (ako one postoje). Iznad liste nalaze se opcije sa pretragu upita po nazivu te otvaranje prozora za stvaranje novog upita.

Kako bi se stvorio upit, potrebno je definirati njegov naziv, zatim napraviti upit (KF pomaže korisniku u izgradnji upita sa svojom „autocomplete“ funkcionalnošću), njegov opis te odabrati agente ili grupu agenata nad kojima će se upit izvršiti. Ako ni jedan agent nije naveden, upit se može samo spremiti u listu. Ako je barem jedan agent, nad kojim je moguće provesti napisani upit, naveden tada se upit može izvršiti. Rezultati upisa biti će prikazani na dnu istog prozora. Slika 50 prikazuje primjer nastanka i izvršavanja upita.

Kao dodatna pomoć pri izgradnji upita, s desne strane prozora nalazi se padajući izbornik sa cijelom „Osquery Schemom“. Korisnik može pronaći željenu tablicu te vidjeti opis tablice, koje sve stupce tablica ima (te kojeg su tipa) i nad kojim operacijskim sustavima je moguće postaviti upit.



Slika 50: Primjer nastanka i izvršavanja upita o operacijskom sustavu agenta [autorski rad]

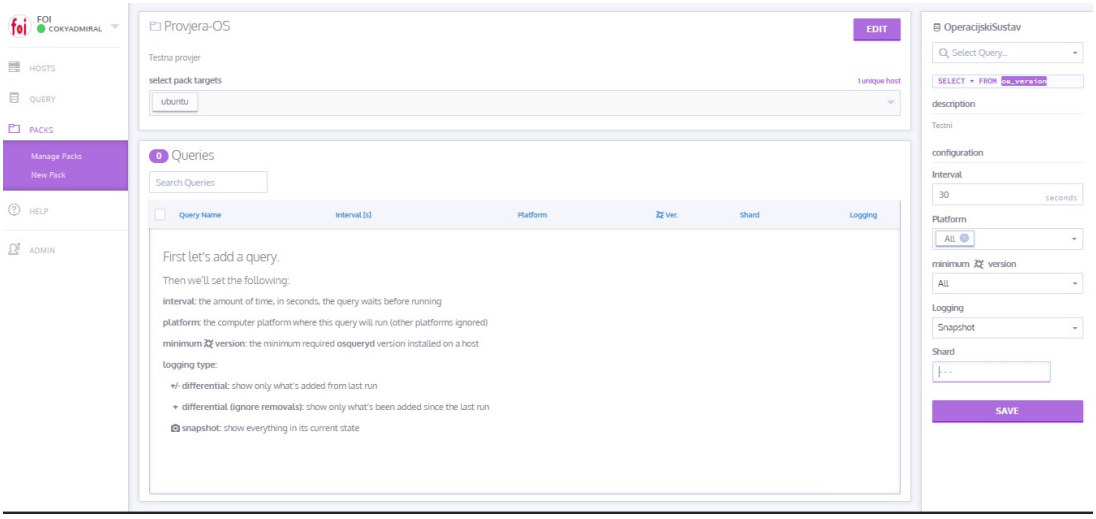
11.2.3. Upravljanje kolekcijama upita nad agentima

Klikom na „PACKS“ unutar alatne trake, otvara se prozor sa popisom svih kolekcija upita. Svaka kolekcija unutar liste ima svoj naziv, broj upita, status, broj agenata koji kolekcija zahvaća te datum zadnje izmjene. Klikom na kolekciju, s desne strane pokazuje se traka unutar koje se ta kolekcija može uključiti ili isključiti (promjena statusa), zatim uređivati, vidjeti opis kolekcije te vidjeti koji sve upiti postoje u kolekciji. Iznad liste nalaze se opcije za pretragu kolekcija po nazivu i otvaranje prozora za stvaranje nove kolekcije.

Kako bi se stvorila kolekcija, potrebno je definirati njezin naziv, opis i agente ili grupe agenata koje će kolekcija upita zahvaćati. Nakon toga potrebno je odabrati „Save query pack“, kako bi se otvorio novi dio prozora s odabirom upita (primjer na slici 51). Mogu se odabrati samo oni upiti koji se nalaze na popisu upita (prozor „QUERY“) i to sa desne strane trenutnog prozora, gdje je ponuđen padajući izbornik. Odabirom upita sa padajućeg izbornika, prikazuju se sljedeći parametri koje je potrebno popuniti:

- „Interval“: vremenski period u sekundama koji mora proći do ažuriranja datoteke zapisa sa rezultatima upita.
- „Platform“: padajući izbornik unutar kojeg se mogu odabrati operacijski sustavi (jedan ili svi) kojima će kolekcija upita biti namijenjena. Iako je upit taj koji određuje moguće operacijske sustave za svoje izvršavanje, unutar ove opcije moguće je dodatno suziti izbor agenata.
- „Minimum Osquery version“: padajući izbornik unutar kojeg se može odabrati najmanja dopuštena inačica Osquery agenta. Ako agent zadovoljava uvjet upita i „Platform“ izbora, ali ne i ovaj parametar, upit se neće izvršiti nad tim agentom.
- „Logging“: Upiti mogu biti pokrenuti preko dva načina: diferencijalno (samo se promjene zapisuju u datoteku zapisa) ili u cijelosti (svaki puta nakon vremenskog intervala zapisuje se puni rezultat).
- „Shard“: Broj koji označuje postotak izvršavanja upita u smislu „koliko će posto agenata od ukupnog broja zahvaćenih agenata izvršiti upite unutar kolekcije“.

Napravljena kolekcija upita automatski je uključena (status „Enabled“). Rezultati kolekcije upita mogu se vidjeti u datoteci zapisa koja je određena tokom pokretanja Kolide Fleet platforme.



Slika 51: Primjer stvaranja kolekcije upita [autorski rad]

11.2.4. Administracija platforme i postavke korisničkog računa

Nakon instalacije i prvog pokretanja KF platforme, potrebno je napraviti administratorski račun unutar web sučelja. Potrebno je navesti puno ime, korisničko ime, e-mail adresu, lozinku, naziv organizacije, avatar organizacije, web adresu i port KF-a. Tek tada moguće je prijaviti se u KF.

Administrator ima svoj posebni prozor na koji se može doći preko alatne trake sa lijeve strane, odabirom na „ADMIN“. Tu se mogu vidjeti postojeći korisnici, njihov kratak info (uloga, status, korisničko ime, e-mail adresa), zatim može se upravljati korisnicima (onesposobiti korisnički račun, promijeniti uloga, poslati „reset password“ e-mail te promijeniti detalji korisnika) i stvarati novi korisnik. KF dopušta stvaranje i korištenje SSO (eng. „Single Sign On“) korisničkih računa.

Administratorski prozor uz sekciju „Manage Users“, koja je upravo objašnjena, sadrži i sekciju „App Settings“, koja se tiče općenitih postavki vezanih uz KF. Postavke su podijeljene na podsekcije:

- „Organization Info“: Unutar ove podsekcije mogu se promijeniti podaci o nazivu organizacije i avatatru (URL na sliku).
- „Kolide Web Address“: Web adresa KF platforme koju će agenti koristiti kod svojeg pokretanja kako bi se spojili na platformu.
- „SAML Single Sign On Options“: Podsekcija za definiranje entiteta koji će potvrditi SSO prijavu korisnika.
- „SMTP Options“: Podsekcija unutar koje se definira e-mail adresa sa koje će se slati KF poruke (npr. e-mail za ponovno postavljanje lozinke), SMTP poslužitelj koji šalje poruke sa definirane e-mail adrese, SMTP port te podaci potrebni za autentikaciju na SMTP poslužitelj.
- „Osquery Emrollment Secret“: Ovdje se može promjeniti definirana „tajna“ koju koriste agenti kako bi se autentificirali kod KF platforme.
- „Advanced Options“: Napredne opcije poput definiranja HELO domene, upravljanja potencijalno nesigurnih certifikata te upravljanje STARTTLS uslugom kod SMTP poslužitelja.

11.3. Instalacija i konfiguracija

Osquery pisan je u C i C++ programskom jeziku te koristi RocksDB u pozadini. Kolide Fleet web je aplikacija čiji je frontend rađen u Redux-u (pisan u React JS okviru), a backend je rađen u Go programskom jeziku. Za bazu podataka Kolide Fleet koristi MySQL i Redis.

Prvi korak je instalacija Osquery okvira. Na Linux platformi potrebno je unutar konzole izvršiti sljedeće naredbe (zadnja inačica Osquery okvira u trenutku pisanja je 3.2.6.):

```
export OSQUERY_KEY=1484120AC4E9F8A1A577AEEE97A80C63C9D8B80B
sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys \$OSQUERY_KEY
sudo add-apt-repository 'deb [arch=amd64] https://pkg.osquery.io/deb deb main'
sudo apt-get update
sudo apt-get install osquery
```

Nakon instalacije, potrebno je samo upisati „osqueryi“ te početi sa pisanjem upita ili odabirom opcija za upravljanje trenutne osquery konzole.

Na Windows platformi potrebno je sa službenih stranica za preuzimanje izvršne Osquery datoteke preuzeti .msi paket te ga instalirati; tokom instalacije nije potrebno podešavati dodatne opcije. Zatim, kako bi se otvorila osqueryi konzola, potrebno je otvoriti cmd.exe ili Powershell konzolu te upisati putanju do osqueryi konzole. Standardno, ta putanja je „C:\ProgramData\osquery\osqueryi.exe“.

U opisanom postupku instalacije Kolide Fleet platforme, koristiti će se Linux Ubuntu Server 16.04. Potrebno je sa službene Ubuntu web stranice preuzeti ISO datoteku Ubuntu Server-a te ju instalirati unutar virtualnog okruženja ili na fizički poslužitelj. Primjer jedne konfiguracije virtualnog stroja nalazi se na slici 40. Primjer postupka instalacije Ubuntu Server 16.04. operacijskog sustava može se pronaći na web stranici https://www.tecmint.com/installation-of-ubuntu-16-04-server-edition/

Za potrebe korištenja Kolide Fleet platforme potrebno je prvo instalirati MySQL bazu podataka i Redis bazu podataka. Za instalaciju i konfiguriranje MySQL baze podataka za rad sa Kolide Fleet platformom potrebno je izvršiti sljedeće naredbe:

```
sudo apt-get install mysql-server -y
echo 'CREATE DATABASE kolide;' | mysql -u root -p
Nakon toga, potrebno je preuzeti, instalirati i konfigurirati Redis:
sudo apt-get install build-essential tcl
cd /tmp
curl -O http://download.redis.io/redis-stable.tar.gz
tar xzvf redis-stable.tar.gz
cd redis-stable
make
make test
sudo make install
sudo mkdir /etc/redis
sudo cp /tmp/redis-stable/redis.conf /etc/redis
```

Sljedeće linije potrebno je napisati unutar datoteke /etc/redis/redis.conf:

```
supervised systemd
dir /var/lib/redis
```

Zatim sljedeće linije potrebno je napisati unutar datoteke /etc/systemd/system/redis.service:

```
[Unit]
Description=Redis In-Memory Data Store
After=network.target

[Service]
User=redis
```

```
Group=redis
ExecStart=/usr/local/bin/redis-server /etc/redis/redis.conf
ExecStop=/usr/local/bin/redis-cli shutdown
Restart=always

[Install]
WantedBy=multi-user.target
```

Potrebno je nastaviti sa konfiguriranjem Redis baze podataka, izvršavajući sljedeće naredbe:

```
sudo adduser --system --group --no-create-home redis
sudo mkdir /var/lib/redis
sudo chown redis:redis /var/lib/redis
sudo chmod 770 /var/lib/redis
sudo systemctl enable redis
sudo systemctl start redis
```

Konačno, potrebno je preuzeti, otpakirati i konfigurirati Kolide Fleet.

```
sudo apt install unzip
wget https://dl.kolide.co/bin/fleet_latest.zip
unzip fleet_latest.zip 'linux/*' -d fleet
sudo cp fleet/linux/fleet* /usr/bin
sudo /usr/bin/fleet prepare db --mysql_address=127.0.0.1:3306 --mysql_database=
    kolide  --mysql_username=root --mysql_password=lozinka
sudo mkdir /var/log/kolide
```

Prije pokretanja platforme potrebno je generirati certifikate koje će Kolide moći koristiti kod upravljanja agenata, ali koje će agenti moći koristiti za autentifikaciju kod prijave na Kolide Fleet platformu.

```
sudo mkdir /etc/pki
sudo mkdir /etc/pki/tls
sudo mkdir /etc/pki/tls/certs
sudo mkdir /etc/pki/tls/private
sudo openssl genrsa -out /etc/pki/tls/private/server.key 4096
sudo openssl req -new -key /etc/pki/tls/private/server.key -out /etc/pki/tls/certs/
    server.csr
sudo openssl x509 -req -days 366 -in /etc/pki/tls/certs/server.csr -signkey /etc/pki
    /tls/private/server.key -out /etc/pki/tls/certs/server.cert
```

Pokretanje Kolide Fleet platforme radi se izvršavanjem sljedeće naredbe...

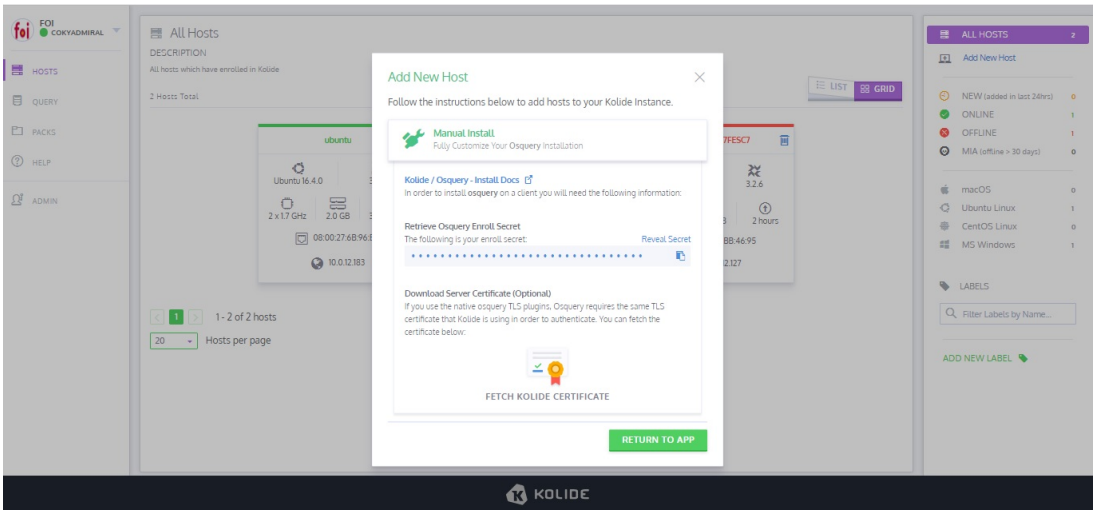
```
sudo /home/korisnik/fleet/linux/fleet serve --mysql_address=127.0.0.1:3306 --
    mysql_database=kolide --mysql_username=root --mysql_password=lozinka --
    redis_address=127.0.0.1:6379 --server_cert=/etc/pki/tls/certs/server.cert --
    server_key=/etc/pki/tls/private/server.key --osquery_result_log_file=/home/coky/
    osquery_result --osquery_status_log_file=/var/log/kolide/osquery_status --
    auth_jwt_key=nasumican_string_koji_predstavlja_tajnu
```

...pod pretpostavkom da je:

- skripta za pokretanje „fleet“ u mapi /home/korisnik/fleet/linux/
- adresa MySQL poslužitelja na „localhost“ adresi i portu 3306

- naziv baze podataka kojom će se Kolide Fleet služiti tokom rada „kolide“
- korisničko ime za prijavu na MySQL bazu podataka jednako „root“
- lozinka za prijavu na MySQL bazu podataka jednaka „lozinka“
- adresa Redis poslužitelja na „localhost“ adresi i portu 6379

Sljedeći korak je pokretanje Osquery okvira na računalima koje će biti agenti pokrenute Kolide Fleet platforme. Certifikat potreban za autentifikaciju agenta moguće je preuzeti sa Kolide Fleet web sučelja (slika 52). Također sa istog prozora na web sučelju moguće je kopirati „tajnu“ te ju spremiti u običnu tekstualnu datoteku. Nakon prijenosa obaju artefakata do računala sa Osquery okvirom, moguće je pokrenuti pozadinski proces...



Slika 52: Prozor za preuzimanje cetrifikata i "tajne" za buduće agente [autorski rad]

```
sudo osqueryd --enroll_secret_path=/home/coky/osquery/var/osquery/enroll_secret --
    tls_server_certs=/home/coky/osquery/var/osquery/10.0.12.183_8080.pem --
    tls_hostname=10.0.12.183:8080 --host_identifier=uuid --enroll_tls_endpoint=/api/
    v1/osquery/enroll --config_plugin=tls --config_tls_endpoint=/api/v1/osquery/
    config --config_tls_refresh=10 --disable_distributed=false --distributed_plugin=
    tls --distributed_interval=3 --distributed_tls_max_attempts=3 --
    distributed_tls_read_endpoint=/api/v1/osquery/distributed/read --
    distributed_tls_write_endpoint=/api/v1/osquery/distributed/write --logger_plugin
    =tls --logger_tls_endpoint=/api/v1/osquery/log --logger_tls_period=10
```

...pod pretpostavkom da je:

- računalo, koji će postati agent, koristi operacijski sustav Linux
- „tajna“ spremljena u datoteci „enroll_secret“ na putanji /home/coky/osquery/var/osquery/enroll_secret
- certifikat naziva „10.0.12.183_8080.pem“ smješten na putanji /home/coky/osquery/var/osquery/10.0.12.183_8080.pem
- IP adresa Kolide Fleet platforme 10.0.12.183:8080

11.4. Kolide Fleet + Osquery recenzija

Većina alata koja je istraživana u korist ovog diplomskog rada ima korisničko sučelje. Ovisno o osobini i navici stručnjaka koji se bave odzivom na sigurnosne incidente, to može biti prednost (uređenost, prikaz kao slika ili animacija, dodatna prilagodba...), ali i nedostatak (gubi se tražena minimalnost, trošenje dodatnih resursa na prikaz sučelja, vremenski odziv). Osquery ne koristi grafičko sučelje – štoviše, ono nije ni potrebno, budući da Osquery ima samo par funkcionalnosti koje je vrlo lako koristiti. Osobe koje su naučene na CLI (eng. „Command Line Interface“) raznih sustava za upravljanje bazama podataka, odmah će shvatiti princip rada i rukovanja ovog alata. Iako se ne koriste prave tablice, apstrakcija podataka operacijskog sustava kao zamišljene baze podataka te način njihova saznanja putem svojevrsnih SQL upita, genijalna je ideja. Forenzika putem interaktivnog rada ili praćenje rada putem pozadinskog procesa, samo su neki od odgovarajućih „parova“.

Instalacija Osquery okvira jako je jednostavna, neovisno o operacijskom sustavu. Autori ovog okvira aktivni su na službenim GitHub stanicama Osquery okvira te odgovaraju na upite korisnika u kratkom roku. Također, izdaju i stalna poboljšanja u performansama, proširenjima i sigurnosti.

Ako ne brojimo svaku tablicu kao funkcionalnost, tada možemo reći da Osquery sadrži samo dvije funkcionalnosti. Interaktivni rad je dostupan kroz pokretanje programa „osqueryi“. Pozadinski proces potrebno je konfigurirati. Datoteka za konfiguraciju, uz svaku opciju, sadrži i objašnjenje. Ako ono nije dovoljno, potrebno je konzultirati se sa službenom dokumentacijom koja je također uredno pisana. Naravno, budući da je Osquery otvorenog programskog koda, dokumentacija nudi razvojnim inženjerima API za proširenje ili unutarnju konfiguraciju okvira za svoje potrebe.

Kolide Fleet jedna je od platformi koja koristi Osquery u svojem radu te je navedena na službenim stranicama Osquery okvira. Korištenje Kolide Fleet platforme također je jednostavno, budući da postoji malen broj funkcionalnosti te moderno i uredno korisničko sučelje. Glavna funkcionalnost ove platforme je upravljanje sa više Osquery-a, što je dodatni „plus“ za sigurnosne stručnjake, jer olakšava forenziku. Bez Kolide Fleet-a, stručnjak bi morao fizički na svakom računalu pokrenuti Osquery u željenom načinu rada, gubeći tako na vremenu i vlastitoj učinkovitosti.

Instalacija platforme je relativno jednostavna, no svakako nije za početnika. Dokumentacija traži od korisnika da zna generirati certifikate, dok jedan od mogućih postupaka instalacije MySQL i Redis baze podataka navodi u jednoj od tekstualnih datoteka na GitHub službenoj stranici. Dokumentacija je pomalo nespretno napisana i teže je snaći se za osobu koja je npr. Linux početnik, jer postupak nije „straight-forward“.

Funkcionalnosti koje Kolide Fleet nudi mogu se gledati kao nadogradnja na postojeće funkcionalnosti Osquery okvira. Ono što platforma nudi je jedna točka upravljanja i prikaza, dok ostale funkcionalnosti poput izvršavanja upita i stvaranja/pokretanja kolekcije upita su zapravo moguće i unutar samog Osquery-a. Uz postojeće funkcionalnosti, ništa više ne fali, osim možda integracije sa Cloud uslugom koju organizacija Kolide također pruža kao jednu od

svojih komercijalnih usluga, ali je van teme budući da nije otvorenog programskog koda.

Od nedostataka koji su primijećeni, bio bi onaj vezan uz pokretanje Osquery okriva kao agenta Kolide Fleet platforme. Postoji pregršt opcija koje su skromno dokumentirane, poput one za korištenje TLS protokola kao proširenja. Štoviše, lako je dovesti pokretanje Osquery-a u konfliktno stanje istodobnim korištenjem definiranih opcija unutar posebne datoteke i definiranih opcija kao argumenata tokom pokretanja. Kod Linux operacijskog sustava postoji problem registriranja kolekcija upita za „osqueryd“ instance. Potrebno je koristiti Osquery CLI, kako bi se kolekcije ručno registrirale, no postoji vjerojatnost nevažećih certifikata za autentifikaciju. Kod Windows operacijskog sustava postoji problem neispravno postavljenih prava pristupa na RocksDB koji Osquery koristi za kolekcije upita. Ovaj problem može se riješiti izvršavanjem sljedećih Powershell naredbi:

```
$path = "c:\PUTANJA\DO\OSQUERY\MAPE\osquery.db"
TAKEOWN /F $path /A /R /D "Y"
icacls \$path /remove:d "NT␣AUTHORITY\SYSTEM" /T
```

Kako god, Osquery i Kolide Fleet odlično se slažu te čine učinkoviti par alata kod forenzike, ali i praćenja rada računala, pa čak i lova na prijetnje (eng. Threat Hunting).

11.5. Kolide Fleet + Osquery alternative

Kolide Fleet i Osquery u simbiozi čine alat klijent-server arhitekture, usmjerenih isključivo na obavljanje forenzike odmah/interaktivno ili periodično/automatizirano, s naglasnom na distribuiranost (jedan poslužitelj može istovremeno komunicirati sa većim brojem klijenata). Sigurnosni stručnjaci tako mogu koristiti alat tokom bilo koje faze napada, istrage ili incidenta, ali i za svrhe obogaćivanja SIEM-a. Pronađene su dvije alternative za alat poput Kolide Fleet + Osquery, a to su Doorman + Osquery te GRR Rapid Response.

11.5.1. Doorman + Osquery

Kao zamjena za Kolide Fleet platformu, Doorman se pokazuje dobrom alternativom. Otvorenog je programskog koda pod „MIT License“ licencom, a platformu je razvio Marcin Wielgoszewski sa velikim doprinosom Andrew Dunhama. Velik broj sličnosti u funkcionalnostima čini odabir težim, no neki od razloga zašto je Kolide Fleet izabran za praktičnu dio u ovom diplomskom radu su sljedeći:

- Ažurnost, nadogradnje i odziv na korisnička pitanja – Doorman je zadnji puta ažuriran prije godinu dana, dok se na odzive za neodgovorena korisnička pitanja čeka po nekoliko mjeseci.
- Pogreška u produkciji (tokom pisanja ovog rada) – zadnja inačica Doorman-a sadrži pogreške, Nginx javlja „502 Bad Gateway“. Ručna instalacija i Docker kontejner se ne mogu koristiti. Inačica prije ne stvara ove probleme.

- Doorman nije kolaboracijski alat – nema registracija unutar web sučelja, već se jedan jedini račun stvara unutar konfiguracijske skripte. Prijava je još moguća uz pomoć LDAP-a i Google OAuth 2.0.

Doorman, od svojih funkcionalnosti, korisnicima pruža:

- Praćenje (ne)aktivnih čvorova i njihovo grupiranje: čvorovi su zapravo registrirane Osquery instance na računalima (analogno agentima kod Kolide Fleet platforme), a registracija se provodi isto kao i kod registracije na Kolide Fleet. Od podataka koje Doorman prikazuje o svojem čvorovima tu su identifikatori čvora, naziv, proizvođač i model računala, serijski broj, proizvođač i model procesora, broj jezgara procesora, količina radne memorije (u bajtovima), zadnja registrirana IP adresa, datum prve registracije, datum zadnje registracije i oznake.
- Detalji čvora: klikom za detaljan prikaz čvora iz liste, korisniku se pojavljuju dodatne informacije poput koje sve kolekcije upita zahvaćaju ovaj čvor, koji su sve spremljeni upiti povezani sa čvorom, kojim se datotekama unutar čvora provjerava integritet, a moguće je i uređivati oznake za trenutno prikazani čvor.
- Izvršavanje upita: Doorman omogućava pisanje i izvršavanje „ad-hoc“ upita te praćenje njihove povijesti izvršavanja i rezultate koje je Osquery vratio.
- Pisanje pravila i slanje alarma: Za razliku od Kolide Fleet platforme, Doorman omogućuje korisniku pisanje uvjeta za okidanje alarma. Pravila se sastoje od naziva, mehanizma slanja alarma te uvjetnog upita - vrijednosti stupaca Osquery tablica te operatori AND, OR, NOT sa mogućnošću gniježđenja.
- Usluga agregacije i slanja podataka/rezultata u vlastito spremište (Elasticsearch, rsyslog, itd.). Kolide Fleet trenutno pruža uslugu spremanja rezultata upita u posebnu datoteku.

Doorman je pisan u Python programskom jeziku, koristi Bower za instalaciju jQuery, bootstrap i ostalih frontend komponenata, dok u pozadini koristi Nginx poslužitelj i PostgreSQL kao bazu podataka. Prijenos podataka između Osquery instanci je zaštićen TLS-om. Vlastita konfiguracija je moguća definiranjem njezine putanje kao vrijednosti DOORMAN_SETTINGS varijable okruženja. Dokumentacija o instalaciji, korištenju i opcijama konfiguriranja kvalitetno je napisana na službenim GitHub stranicama https://github.com/mwielgoszewski/doorman

11.5.2. GRR Rapid Response

Kao alternativa koja ne koristi Osquery, ali se i dalje oslanja na klijent-server arhitekturu, GRR Rapid Response (skraćeno GRR) omogućuje vođenje „udaljenje“ forenzike, pa tako i odziv na sigurnosne incidente, uz pomoć posebno napravljenih GRR klijenata i GRR poslužitelja. Otvorenog je programskog koda pod „Apache License 2.0“ licencom, a platformu je razvio Google razvojni tim. Ne postoji posebni razlog zašto GRR nije odabran za praktični dio ovog diplomskog rada, prema svojim funkcionalnostima i pristupu odličan je konkurent Osquery okviru

sa Kolide Fleet platformom. Nažalost, kao i Doorman, GRR ne podržava registraciju niti mehanizme višekorisničkog rada, jer za svoje potrebe i svrhu to smatra nepotrebnim. Za razliku od „push“ metode kojom Kolide Fleet ažurira i ispituje svoje agente, ovdje GRR klijenti šalju „pull“ zahtjev GRR poslužitelju kako bi saznali svoje zadatke. GRR klijente može se preuzeti direktno sa web sučelja GRR poslužitelja. Instalacija klijenta, kao i kod Osquery, moguća je na više operacijskih sustava: Linux, Windows i MacOS. Iako se ne koristi apstrakcija operacijskog sustava kao baze podataka, GRR korisiti moćan Sleuth Kit – kolekciju alata komandnog sučelja i biblioteka pisanih u C jeziku koje omogućuju analizu diskovnog prostora i ekstrahiranja datoteka. Razvijena je i podrška za YARA pravila (definirana od strane korisnika na GRR poslužitelju), omogućujući udaljenu analizu radne memorije u stvarnom vremenu. Budući da ovakav način rada može trošiti popriličan broj računalnih resursa, GRR klijenti imaju način nadgledavanja sustava na kojem su pokrenuti i ograničavanja potrošnje resursa. Od funkcionalnosti koje se nalaze na GRR poslužitelju, tu su:

- Povezivanje sa GRR klijentom: Pregled svih klijenata sa dohvaćenim informacijama poput identifikatora klijenta, operacijskim sustavom, MAC adresom, korisničkim imenom, prvim datumom registracije, inačicom GRR klijenta, oznaka i datuma zadnje istrage.
- Prikaz informacija o (fizičkom) poslužitelju na kojem se GRR poslužitelj nalazi. Velik broj informacija o hardveru, instaliranom operacijskom sustavu, korisnicima, radnoj memoriji, raznim datumima, mrežnim sučeljima, diskovnim prostorima, itd.
- Upravljanje tokovima: GRR uvodi pojam toka (eng. „Flow“) kao predefinirani zadatak za klijenta koji se sastoji od forenzike mreže računala (Netstat), diskovnog prostora, povijesti web preglednika, itd. Ova funkcionalnost analogna je Osquery periodičnim upitima. Svaki od tokova ima nekoliko opcija i polja za konfiguraciju koje je potrebno ispuniti, ne bi li klijent preuzeo tok, izvršio ga kao zadatak/zadatke unutar određenog vremenskog perioda i na kraju poslao rezultate toka natrag GRR poslužitelju. U alatnoj traci moguće je odabrati prozor za stvaranje tokova (eng. „Start new flows“) ili upravljanje pokrenutim tokovima (eng. „Manage launched flows“).
- Udaljen pristup datotečnom prostoru na kojem se nalazi klijent: ova funkcionalnost je moguća jer se klijent izvršava sa „root“ ili administrator pravima. Datotečni prostor vizualiziran je kao stablo, uz napomenu da to stablo nije inicijalno popunjeno. Stablo se puni informacijama o mapama i datotekama ručnim upitom prema klijentu. Datoteke je moguće preuzeti sa GRR poslužitelja za daljnju obradu izvanmrežno.
- Upravljanje Cron poslovima: Cron poslovi služe za periodično čišćenje i održavanje GRR poslužitelja i njegovih klijenata.
- Upravljanje lovom: tokovi nisu ograničeni za pokretanje na samo jednom klijentu. Lov (eng. „hunt“) omogućuje pokretanje tokova istovremeno nad više klijenata prema određenim uvjetima. Lov je analogan kolekciji upita kod Osquery okvira.
- Prikaz statistika vezanih uz forenziku.
- Prikaz vizualizacija opterećenja GRR poslužitelja.

- - Upravljanje binarnim datotekama: tu se mogu preuzeti dodijeljene Python skripte i ostale izvršne datoteke. Ovaj prozor je ujedno jedini način na koji je moguće preuzeti GRR klijente.
- Opcije i konfiguracija GRR poslužitelja (samo za čitanje).
- Upravljanje artefaktima: GRR poslužitelj daje veliku fleksibilnost korisniku u vođenu artekataka, odnosno dokaza koje je prikupio tokom forenzike. Artefakti, kao datoteka, mogu se prenijeti na poslužitelj te tamo grupirati i biti obogaćeni potrebnim metapodacima.

GRR je većinski pisan u Python skriptnom jeziku, a baza podataka koja se koristi za pohranu svih radnih podataka je MySQL. Prijenos podataka između GRR poslužitelja i GRR klijenata zaštićen je TLS-om. Instalacija, konfiguracija, održavanje i još mnogo toga opširno je napisano unutar službene dokumentacije na Read the Docs adresi https://grr-doc.readthedocs.io/en/latest/index.html

Jedan od načina instalacije GRR je preuzimanje DEB paketa (za Debian Linux distribucije, kao što je Ubuntu Server 16.04) preko naredbe:

```
wget https://storage.googleapis.com/releases.grr-response.com/grr-server_3.2.3-2
    _amd64.deb
```

Kako bi se instalirao i pokrenuo poslužitelj, potrebno je upotrijebiti preuzeti DEB paket:

```
sudo apt install -y ./grr-server_3.2.3-2_amd64.deb
```

Nakon što je instalacija gotova, GRR poslužitelj trebao bi biti instaliran i pokrenut. Ponovo pokretanje (ako je potrebno) moguće je izvršiti naredbom:

```
sudo systemctl restart grr-server
```

Web sučelje GRR poslužitelja dostupno je kroz IP adresu virtualne mašine na portu 8000

12. MozDef

MozDef, skraćeno od Mozilla Defense Platform, platforma je za upravljanje i odziv sigurnosnim incidentima sa naglašenom komponentom upravljanja sigurnosnim informacijama i događajima (eng. „Security Informaition and Event Management“, u nastavku SIEM). Otvorenog je programskog koda pod licencom „Mozilla Public License 2.0“, a platformu je razvila zajednica Mozilla. MozDef može djelovati kao SIEM, jer koristi tehnologije koje primaju, obrađuju, spremaju te vizualno prikazuju datoteke zapisa (eng. „Logs“). Budući da su SIEM platforme van opsega ovog diplomskog rada, objasniti će se funkcionalnosti MozDef-a kod upravljanja i odziva sigurnosnim incidentima: generiranje alarma putem softvera za obavljanje periodičnih zadataka nad podacima koji su smješteni u bazi podataka, stvaranje istraga ili lova na prijetnje, upravljanje incidentima putem VERIS taksonomije i vizualizacija navedenih komponenti. Dio Mozdef-a za upravljanje incidentima omogućuje kolaboraciju između više korisnika platforme, baš kao i prije opisani TheHive.

Spomenute funkcionalnosti, kao i njihov položaj i korištenje unutar samih alata, biti će opisani u sljedećim poglavljima.

12.1. MozDef funkcionalnosti

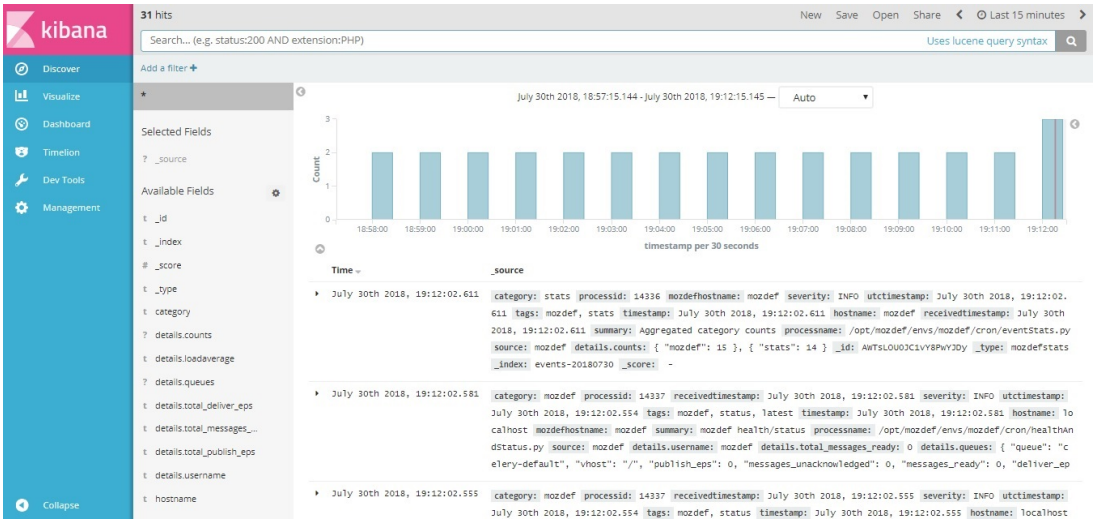
Unutar ovog poglavlja slijedi opis svih funkcionalnosti koje je autor prepoznao unutar rada MozDef-a (prikazane će biti autorove slike, testni podaci, položaj kontrola za obavljanje raznih funkcionalnosti...) te uz pomoć službene dokumentacije sa Read the Docs stranice MozDef platforme na adresi: http://mozdef.readthedocs.io/en/latest/

12.1.1. Usluga upravljanja sigurnosnim informacijama i događajima

MozDef koristi broker za poruke koji inicijalno prima datoteke zapisa sa raznih izvora (Nginx, Syslog, Logstash...) i to unutar JSON formata preko HTTP(S), AMQP(S) ili SQS protokola. Koristeći transformaciju podataka putem raznih proširenja, koja uključuju normiranje, dodavanje metapodataka itd. datoteke zapisa dolaze u bazu podataka koja je pogodna za brzo spremanje i indeksiranje te brzu i fleksibilnu pretragu spremljenih datoteka. Kako bi prikazao datoteke, MozDef koristi posebnu aplikaciju s bogatim grafičkim sučeljem za prikaz i filtriranje gotovo svakog polja spremljene datoteke, u raznim oblicima vizualizacije. Također, preko tog alata moguće je u stvarnom vremenu pratiti nove datoteke koje dolaze u bazu podataka, kao što je primjer na slici 53. Sve ove značajke osnovna su SIEM platforme, što MozDef i pruža sa svojim komponentama, koje su također otvorenog programskog koda.

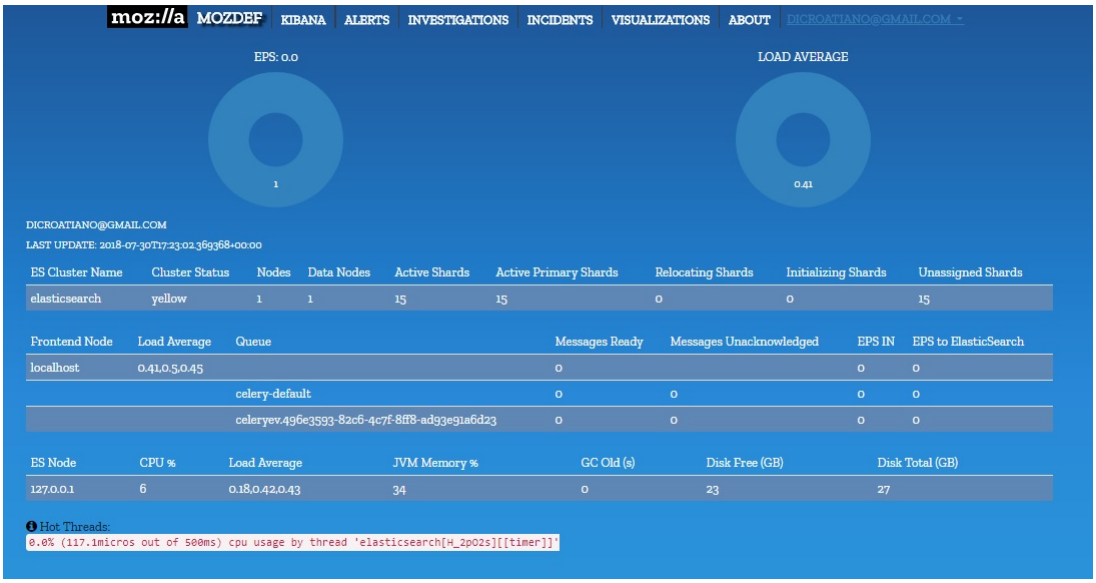
Svaki izvor podataka koji želi slati datoteke zapisa u MozDef, mora znati njegovu adresu i način na koji brokeri za poruke mogu primiti datoteke zapisa za daljnju obradu. Proširenja, koja „obogaćuju“ datoteke zapisa, također moraju biti pravilno konfigurirana kako baza

podataka ne bi dobivala neispravne podatke koji bi (u najgorem scenariju) srušili bazu podataka.



Slika 53: Primjer prozora aplikacije koja prati sve spremljene datoteke zapisa unutar baze podataka [autorski rad]

Potrebno je napomenuti da u kartici „About“ (slika 54) unutar alatne trake web sučelja MozDef postoji uvid u platformu, točnije u SIEM komponentu, gdje su evidentirani podaci o statusu baze podataka, poslužitelju i korištenim resursima za održavanje platforme.

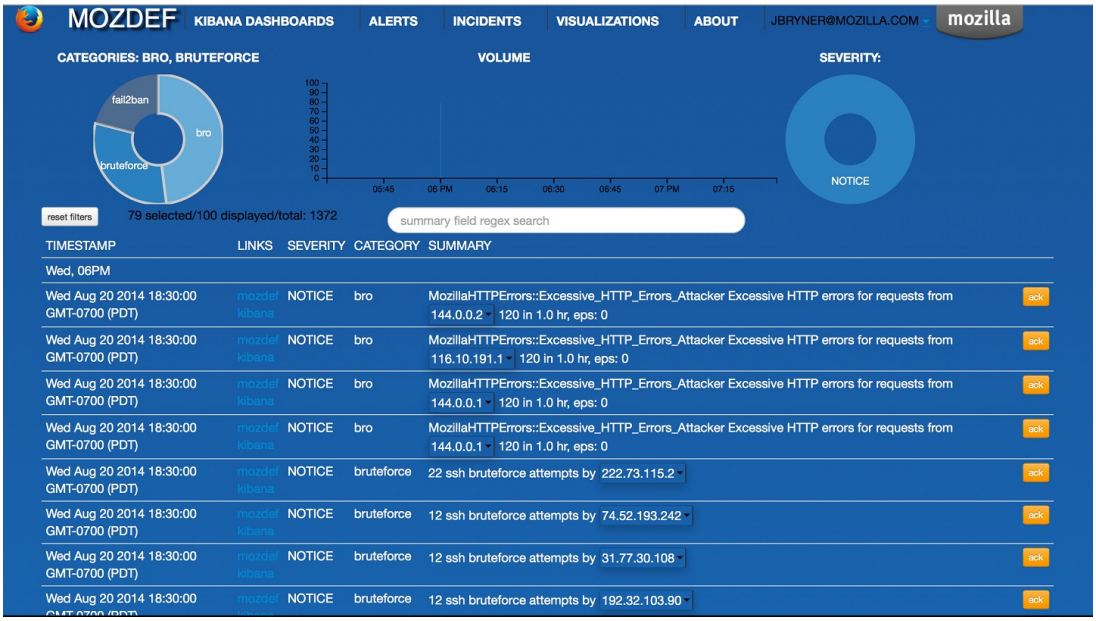


Slika 54: Prozor "About" [autorski rad]

12.1.2. Definiranje i pregled alarma

Alarmi su Python skripe pokrenute kao zadaci koji šalju upit bazi podataka u potrazi za individualnim događajima ili korelacijom više njih. Prozor sa alarmima (slika 55) pokazuje zadnjih 100 alarma i dozvoljava interaktivno filtriranje po kategoriji, ozbiljnosti alarma, datumu te formatu upita pisanom uz pomoć Regex predložaka. Kako u stvarnom vremenu dolaze novi

alarmi, tako se ovaj prozor ažurira. Pokraj svake prikazane IP adrese stoji padajući izbornik koji omogućuje slanje upita za tu adresu na web usluge poput whois, dshield, CIF, itd. Također, moguće je dodati IP adresu na listu zabrane pristupa poslužitelju, definirajući konkretno IP adresu ili podmrežu, trajanje zabrane, komentar, referencu te proširenja (VCP Blackhole, IPBlockList, Banhammer, Facebook ThreatExchange).



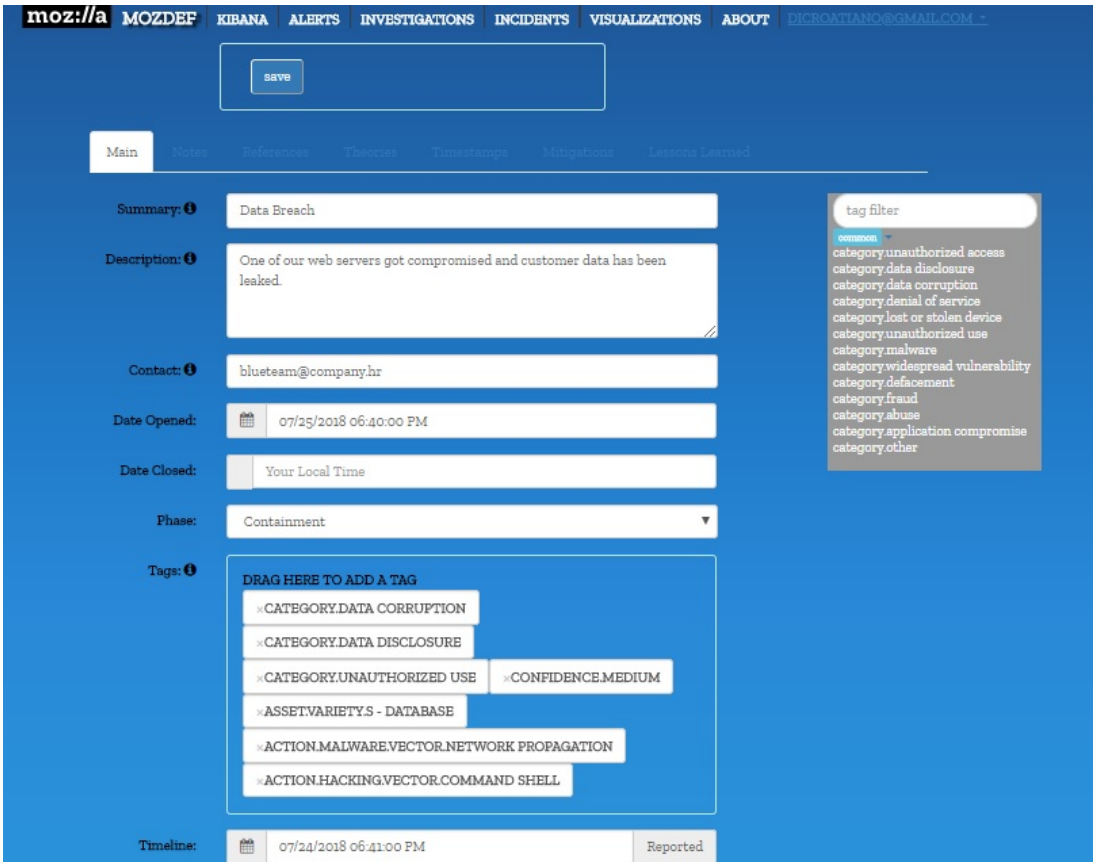
Slika 55: Prozor "Alarms" [35]

12.1.3. Upravljanje istragama i incidentima

Ono što čini MozDef pravim alatom za SOC je funkcionalnost stvaranja istrage ili incidenta uz pomoć faza opisanih unutar metodologija upravljanja i odziva na sigurnosne incidente te upotreba VERIS okvira. Kod stvaranja nove istrage, uz naziv i datum otvaranja, potrebno je odabrati inicijalnu fazu trenutne istrage: identifikacija, dokazivanje, držanje, eskalacija ili završetak. Kod stvaranja novog incidenta, faze mogu biti: identifikacija, ograničavanje, iskorjenjivanje, oporavak ili novo-naučeno. Oba prozora pokazuju zadnjih 100 istraga odnosno incidenata u obliku tablice. Svaki redak prikazuje opciju uređivanja i brisanja, zatim naziv, fazu, datum i vrijeme kad je istraga/incident stvoren te tko ga je stvorio. Odabirom opcije uređivanja, prikazuju se polja za promjenu naziva, opisa, e-mail adrese za kontakt, datuma otvaranja i zatvaranja istrage/incidenta te oznaka koje se temelje na VERIS okviru. Oznake je moguće pronaći sa desne strane prozora te ih je potrebno premjestiti u predviđeno polje. Spomenuta polja nalaze se u glavnoj sekciji (primjer na slici 56), a postoje još i sljedeće sekcije:

- „Notes“: Definiranje bilježaka, tj. svaka moguća informacija sa kojom bi se istraga ili incident mogli povezati, a da ta informacija nije službeno dokumentirana. Sastoji se od definiranja naziva bilješke i njenog sadržaja/opisa.
- „Reference“: poveznice koje pomažu u razumijevanju i rješavanju istrage/incidenta.

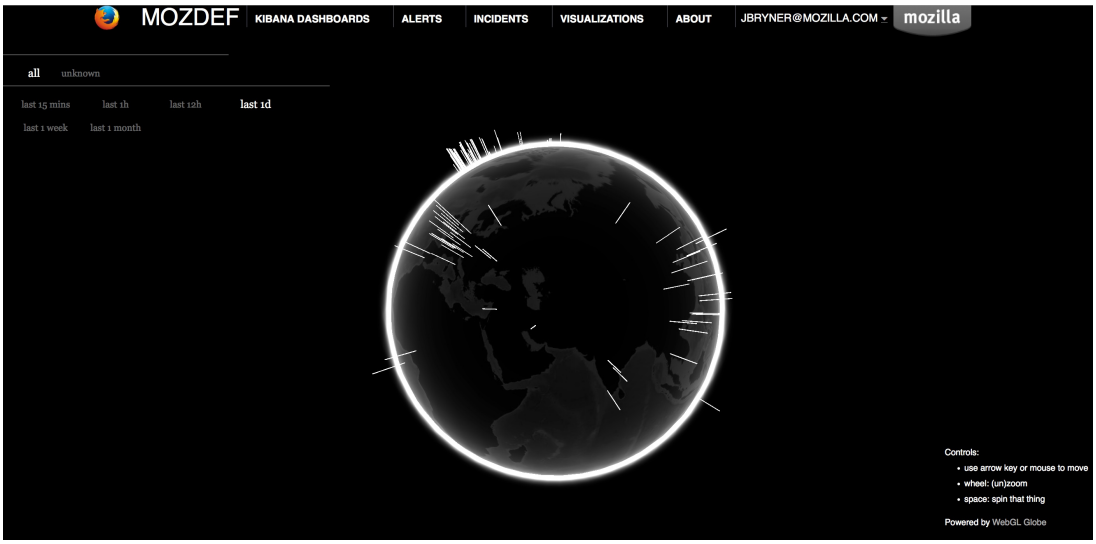
- „Theories“: definiranje mogućih teorija/sumnja oko istrage ili incidenta. Sastoji se od definiranja naziva teorije i njenog sadržaja/opisa te statusa (potvrđena ili nepotvrđena teorija).
- „Timestamps“: datumi i vremena koja pomažu u razumijevanju i rješavanju istrage/incidenta.
- „Mitigations“: definiranje akcija koje zbrinjavaju rizike ili ublažuju štetu. Sastoji se od naziva, opisa, statusa (implementirano ili ne) te zastavice je li akcija pomogla privremeno ili za stalno.
- „Lessons Learned“: zaključci i rezultati tehnika/metoda koje su korištene tokom vršenja istrage ili obrade incidenta. Koristi se u svrhu budućeg poboljšanja kod istih ili sličnih istraga/incidenata. Sastoji se od naziva i opisa naučenog.
- „Indicators“: Svaki alarm ili aktivnost koja baca sumnju te potiče razvoj istrage. Sastoji se od naziva indikatora i njegova opisa.
- „Evidence“: Svi dokazi koji oblikuju krajnji rezultat istrage. Sastoji se od naziva dokaza i njegova opisa.



Slika 56: Prozor "Incidents", sekcija "Main" primjer upravljanja incidentom [Autorski rad]

12.1.4. Vizualizacija prijava korisnika i lokacija napadača

Uz posebnu aplikaciju koja pokazuje podatke spremljene unutar baze podataka preko raznih vrsta vizualizacija, MozDef ima posebnu opciju u alatnoj traci „Visualizations“ koja prikazuje broj ukupnih prijava (omjer uspješnih i neuspješnih prijava) trenutno registriranih korisnika, zatim tri vrste vizualizacije aktivnosti napadača: globus (lokacije napadača, primjer na slici 57), antagonisti (korelacija alarma i događaja unutar 3D reprezentacije napadača) i ratna zona (područje koje vizualizira ozbiljnost napada i ostale njegove detalje).



Slika 57: Vizualizacija u obliku zemaljske kugle koja pokazuje lokacije napadača, bazirano na njihovim IP adresama [35]

12.2. Instalacija i konfiguracija

MozDef je složena platforma, budući da može istovremeno nadomjestiti nekoliko „obrambenih“ mehanizama u mreži. Za svoje web sučelje koristi MeteorJS sa integracijom MongoDB kao baze podataka i d3.js/dc.js/three.js za vizualizacije. Aplikacija je smještena na Nginx web poslužitelju preko uWSGI-a koji kontrolira Python skripte (bottle.py biblioteka) za upravljanje web zahtjevima. Za potrebe SIEM-a MozDef koristi RabbitMQ kao broker za datoteke zapisa koje dolaze sa raznih izvora podataka, Elasticsearch kao glavnu bazu podataka nad kojom MozDef izvršava upite/pretraživanja i posebnom Kibana instancom unutar koje korisnik može na vlastiti način oblikovati vizualizacije podataka iz Elasticsearch-a. Na portu 80 moguće je pristupiti web sučelju, zatim na portu 9090 nalazi se Kibana, na portu 9200 Elasticsearch, na port 8080 dolaze datoteke zapisa dok se na portu 8081 nalazi API.

U opisanom postupku instalacije MozDef platforme, koristiti će se Linux Ubuntu Server 16.04. Potrebno je sa službene Ubuntu web stranice preuzeti ISO datoteku Ubuntu Server-a te ju instalirati unutar virtualnog okruženja ili na fizički poslužitelj. Primjer jedne konfiguracije virtualnog stroja nalazi se na slici 40, no budući da se radi o većoj platformi negoli prijašnje dvije koje su opisane tokom ovog rada, preporuka je koristiti barem 6 GB radne memorije

unutar virtualne. Primjer postupka instalacije Ubuntu Server 16.04. operacijskog sustava može se pronaći na web stranici
https://www.tecmint.com/installation-of-ubuntu-16-04-server-edition/

Najlakši način instalacije te najbrži način korištenja MozDef platforme je putem Docker kontejnera, stoga je potrebno instalirati Docker:

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
sudo add-apt-repository "deb␣[arch=amd64]␣https://download.docker.com/linux/ubuntu␣\
    $(lsb_release␣-cs)␣stable"
sudo apt-get update
sudo apt-get install -y docker-ce
```

Nakon toga, potrebno je klonirati MozDef projekt sa službene Git stranice:

```
git clone https://github.com/mozilla/MozDef
```

Također, za izradu kontejnera i pokretanje platforme potreban je program „make“:

```
sudo apt-get install make
```

Potrebno je preuzeti i pripremiti programe o kojima ovisi MozDef te preuzeti kontejnere koji čine SIEM i ostatak platforme. Ovaj proces prvi puta može potrajati neko duže vrijeme, a stvoriti će jedan kontejner u kojima će se nalaziti cijela platforma:

```
sudo make single-build
```

Nakon što je proces završen, kontejner se može „podići“ naredbom:

```
sudo make single-run
```

MozDef je nakon nekoliko sekundi pristupačan na adresi VM-a. Web sučelju se može pristupiti bez eksplicitnog upisa porta.

12.3. MozDef recenzija

Tražeći alate otvorenog programskog koda za upravljanje i odziv na sigurnosne incidente, često se može naići na razne SIEM platforme, glomazna rješenja, arsenal za „sve i svašta“, stoga je potrebno dobro filtrirati odabir, tako da se dobije precizan alat sa što više funkcionalnosti, ali ne previše izvan opsega zadatka koji sam alat mora pokriti. MozDef je jedan od alata koji ima ovu karakteristiku. Uz SIEM koji sakuplja i pokazuje podatke koji se analiziraju unutar koraka upravljanja i odziva na sigurnosne incidente, MozDef pruža dodatno sučelje baš za tu svrhu. Na taj način stručni timovi ne trebaju koristiti više različitih alata, pa zbog toga nema dodatnog posla oko integriranja ili mogućih nekompatibilnosti.

Koristi li se Docker kontejner, instalacija MozDef platforme ne može biti lakša. Korisnicima je čak omogućeno da biraju žele li sve komponente platforme unutar jednog kontejnera ili više njih. Dokumentacija u potpunosti opisuje načine instalacije i pokretanja platforme preko kontejnera i ručno (preuzimanjem, instaliranjem i konfiguriranjem svake komponente). Ono što dokumentacija također jako dobro opisuje je što SIEM zapravo radi, sa kakvim podacima je

potrebno puniti SIEM, što je potrebno pratiti te kako MozDef iskorištava prednost SIEM-a „integriranog“ unutar same platforme. Ono što dokumentacija opisuje dosta skromno je komponenta koju bi koristili stručnjaci za odziv sigurnosnim incidentima.

Funkcionalnosti MozDef platforme, ako isključimo SIEM, dosta su skromne. Ako izdvojimo funkciju analize i blokiranja IP adresa te vizualizacije (čiji je izbor također skroman), ostaje popunjavanje istraga i incidenata tekstom u za to predviđena polja. Korisnika se navodi na preporučen način vođenja i rješavanja istraga i incidenata uz pomoć sučelja podijeljenog po metodologiji iz SANS-a. Dodatno olakšanje kod opisa incidenta stvara VERIS okvir, koji je unutar MozDef sučelja implementiran kao sustav oznaka.

Web sučelje MozDef platforme prosječno je loše dizajnirano. Iako intuitivno, nepotrebno je imati dvije opcije (istraga i incidenti) koje se razlikuju u tome što jedna ima dvije sekcije za tekst više od druge. Tema ne koristi usklađene boje. Sučelje nije responzivno. U polja za unos teskta ponekad nije moguće pisati. Funkcionalnost zaboravljene lozinke ne radi, pa e-mail korisniku nije poslan. Od funkcija još nedostaju administratorska ploča (definiranje e-mail poslužitelja, SSL certifikata i prijenosa podataka, upravljanje korisnicima i ulogama. . .), slanje e-maila za aktivaciju korisnika nakon registracije, ploča za izradu vizualizacija (tri od četiri moguće vizualizacije prikazuju istu stvar), korelacija između prozora „Alerts“ i Kibana aplikacije. Kibana može jasnije prikazati one podatke koje pokazuje „Alerts“ prozor, čineći time „Alerts“ prozor suvišnim. Kibana se trenutno koristi kao dodatni „nekonfigurirani“ alat za korisnika te ako nije potrebna u radu, ona bespotrebno troši poslužiteljske resurse. Uspoređujući sa ostalim platformama sa SIEM komponentnom, MozDef je jedina platforma otvorenog programskog koda koja ne treba više od 5,2GB radne memorije za normalan rad, čineći je tako „performance-friendly“ (Cyphon i Zentral, o kojima će kasnije biti riječi, ne mogu biti pokrenuti ispod 8 GB radne memorije).

12.4. MozDef alternative

Platforma sa već implementiranim SIEM-om i sučeljem koje omogućava kolaboraciju na domeni upravljanja i odziva sigurnosnih incidenata, moćan je alat svim SOC, CSIRT i ostalim „blue“ timovima. Također, ovakva vrsta platforme otklanja problem kompatibilnosti i troškove spajanja već postojećeg SIEM riješenja. Postoje dva alternativna proizvoda otvorenog koda, koja mogu poslužiti kao dobra zamjena za MozDef: Zentral (usko usmjeren na Apple uređaje) i Cyphon (velika fleksibilnost i zanimljiva terminologija, ali veliki "potrošač" resursa).

12.4.1. Zentral

Potreban li je SIEM sa integriranom podrškom za Osquery okvir te s Kolide Fleet funkcionalnostima? Poslužitelj koji je namijenjen ispitivanju bilo kojeg čvora unutar mreže, s naglaskom na opremu koja koristi MacOS? Platformu koja je izgrađena modularno te za svoj rad ne treba sve komponente već one koje treba korisnik? Zentral je odlično rješenje, jer je spoj alata i komponenata za forenziku, upravljanje softverom, praćenje događaja te organiziranje i grupiranje mrežnih čvorova, a još k tome i sadrži karakteristike SIEM platforme. Otvorenog je

programskog koda pod „Apache License 2.0“ licencom, a projekt je pokrenuo Apfelwerk, mreža konzultanata Apple organizacije unutar prostora Njemačke. Napravljen je isključivo za organizacije koje za krajnja računala te ostalu informatičku opremu koriste proizvode tvrtke Apple. Svaka komponenta ove platforme može sakupljati, analizirati, pratiti pa čak i mijenjati informacije i softversku opremu MacOS/OS X klijenata. Zašto je Zentral naveden kao alternativa MozDef-u? Zato što Zentral također sadrži SIEM komponentu (čak i isti tehnološki stog) te sadrži poslužitelj za praćenje i upravljanjem Osquery okvirom instaliranog na klijentima. Ipak, preopširna je platforma za opis unutar ovog diplomskog rada zbog povećeg broja funkcionalnosti i komponenata. Funkcionalnosti se ukratko mogu podijeliti na sljedeće:

- Upravljanje registriranim klijentima: moguće je pretražiti (uz pomoć filtera) sve registrirane klijente te tako pronaći spremljene podatke o njima, poput modela uređaja, procesora, broja jezgara, radne memorije, operacijskog sustava, instaliranih aplikacija, pripadajućih grupa te događaja (jedan stupac je vrsta događaja sa datumom, drugi stupac je sadržaj događaja u JSON formatu).
- Upravljanje aplikacijama na MacOS uređajima: Unutar ovoga prozora moguće je pronaći sve registrirane MacOS aplikacije registriranih klijenata.
- Instalacija i upravljanje sondama: sonde kao „klijenti“ (Osquery, Munki, Google Santa, events) za klijentske uređaje.
- Upravljanje izvorima otkrivenih indikatora kompromisa.
- Slanje alarma na razne usluge: Slack, Zendesk, GitHub, Email, JSS, itd.

Zentral je pisan u Python3 skriptnom jeziku te koristi Django 1.9 web okvir. Aplikacija je smještena na Nginx web poslužitelju. Od ostalih platformi, sa kojima surađuje, tu su: Elasticsearch (baza podataka svih uređaja, događaja, itd.), PostgreSQL (baza podataka za Zentral platformu – korisnici, oznake, itd.), RabbitMQ (broker poruka, služi kao agregator) te Kibana i Prometheus za statistike i vizualizacije. Podatkovni kanali koriste TLS kod komunikacije. Detaljan opis platforme, instalacija te videozapisi (njihove poveznice) koji prikazuju rad unutar platforme, nalaze se na službenoj GitHub Wiki stranici Zentral platforme na adresi https://github.com/zentralopensource/zentral/wiki

12.4.2. Cyphon

Jedno od najzanimljivijih rješenja za upravljanje i odziv sigurnosnim incidentima zasigurno je Cyphon. Otvorenog je programskog koda pod „GNU General Public License v3“ licencom, a platformu je razvio Dunbar Cybersecurity razvojni tim. Cyphon pruža uslugu SIEM-a, sustava za upravljanje ulaznicama i kolaboracije nad incidentima. Odlična je alternativa MozDef platformi, no za svoj rad iziskuje puno resursa (8 GB radne memorije minimalno), ali i krivulja učenja rada na Cyphon poslužiteljskoj strani znatno je veća nego na ostalim spomenutim platformama unutar ovog diplomskog rada. To je ujedno i ono što je zanimljivo kod ove platforme – širok odabir konfiguracije uz pojmove koji se tiču industrije alkoholnih pića. Komponente koje obavljaju primanje, filtriranje, obogaćivanje, grupiranje i skladištenje podataka imaju

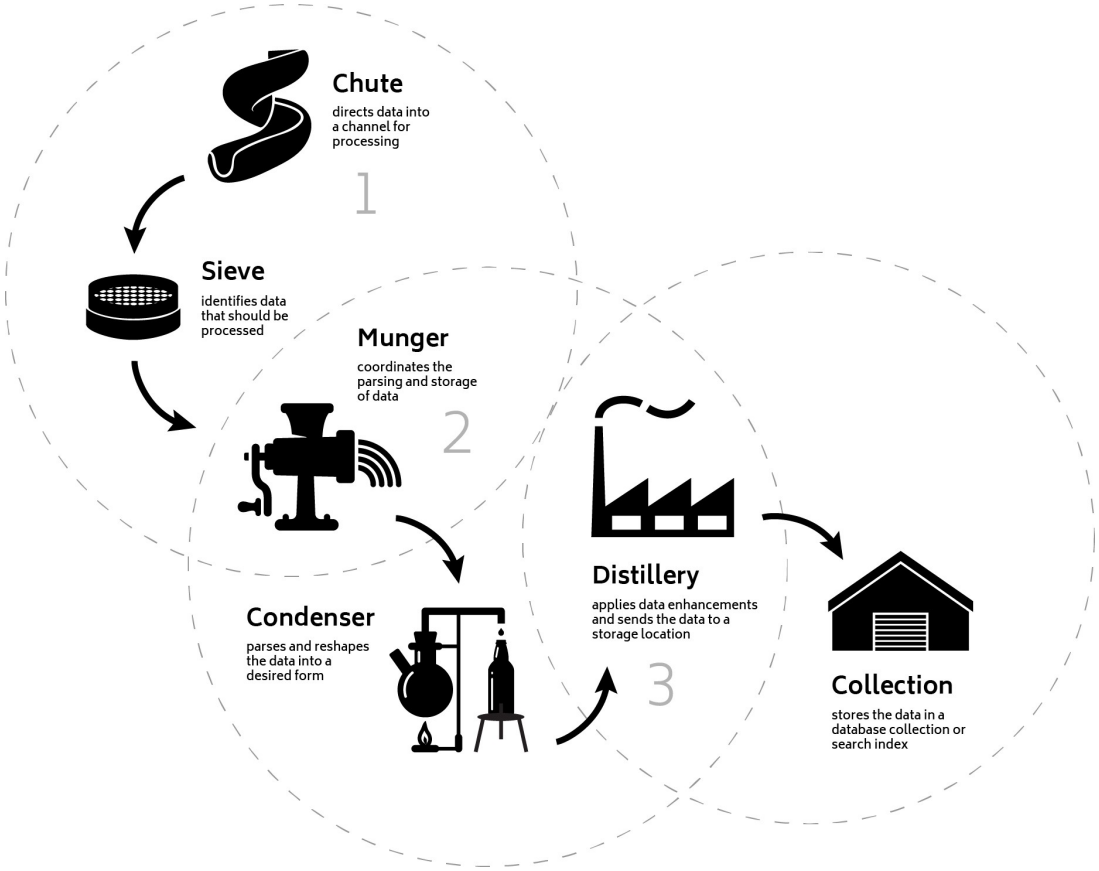
nazive poput boce, naljepnice, sita, lijevka, kašete, destilerije, itd. Rad Cyphon platforme može se opisati u četiri faze. Prva faza uključuje sakupljanje podataka: SIEM (ili direktno IDS) alarmi, društvene mreže, API-jevi, Skeneri ranjivosti, itd. Druga faza uključuje alarmiranje: nakon što se podaci sakupe, oni se analiziraju i obogaćivanju, filtriraju i grupiraju, a na kraju im se i odrede prioriteti koji će označiti kakav se alarm šalje korisniku na kontrolnu ploču. Treća faza je istraga, koja uključuje klijentsku stranu Cyphon-a nazvanu Cyclops, a sastoji se od vizualizacije alarma te kolaboracije korisnika platforme. Zadnja faza je odziv, a uključuje daljnje alarmiranje, izvršavanje procesa otklona štete, itd. Od funkcionalnosti koje pruža poslužiteljska strana, tu su:

- Konfiguriranje i oblikovanje podataka (eng. „shaping dana“): Stvaranje polja (naziv, tip, ciljana skupina...) za boce i stvaranje samih boca (eng. „bottle“) – elemenata koji predstavljaju model podataka. Stvaranje polja (naziv, tip, ciljana skupina, „inspekcija“...) za naljepnice i stvaranje samih naljepnica (eng. „label“) – elemenata koji dodatno opisuju podatke sačuvane u bocama. Namještanje kontejnera, koji se sastoje od boca i naljepnica te okusa (eng. „taste“) koji predstavlja generičke podatke o alarmima na temelju kojih se oni mogu pretražiti, filtrirati, itd.
- Konfiguriranje skladišta (eng. „warehouse“) i kolekcija (eng. „collection“): Cyphon fizički sprema podatke u skladište (naziv, vrsta baze) u obliku kolekcije (naziv kolekcije, naziv postojećeg skladišta). Podaci se nalaze unutar nekog kontejnera koji je povezan sa barem jednom kolekcijom (taj dio se konfigurira u destileriji).
- Konfiguriranje izvora podataka (eng. „data collection“): Cyphon može skupljati tri vrste podataka: datoteke zapisa, e-mailove i podatke sa socijalnih mreža (npr. „tweetovi“. Prikupljanje datoteka zapisa za analizu, kao potencijalne alarme, konfigurira se unutar posebnih usluga. Prikupljanje e-mailova konfigurira se unutar sekcije upravljanja e-mailovima – stvaranje novog poštanskog sandučića (naziv, URI, adresa maila), uređivanje e-mailova (poštanski sandučić, naslov, ID, zaglavlja...) i privitaka. Kod konfiguriranja socijalnih mreža kao izvora podataka (trenutno u toku pisanja, napravljena je samo podrška za Twitter), potrebno je prvo definirati parametre traženja. Moguće je definirati parametre za pratitelje (nadimak, puno ime i prezime, korisnički ID, korisničko ime), lokacije (naziv, geometrijska lokacija na karti) te pojmovi koji se traže (naziv pojma, negacija značenja). Nakon toga, moguće je napraviti filtere koji se baziraju na novonastalim parametrima i te filtere primijeniti kod kondenzatora koji spaja izvore podataka sa kontejnerom.
- Konfiguriranje procesa i analize nad podacima: Nakon što Cyphon dobije podatke kroz svoje izvore, njih je potrebno procesirati. U tome sudjeluju šaht (eng. „chute“), sito (eng. „sieve“), ušće (eng. „munger“), kondenzator (eng. „condenser“) i destilerija (eng. „distillery“), kao što je prikazano na slici 58. Šaht zahtjeva sito i ušće te predstavlja poveznicu između njih. Sito predstavlja filter podataka koji pristižu te ako zadovoljavaju određeni kriterij, šalje ih dalje. Sito se sastoji od jednog ili više pravila (eng. „rules“) koje predstavljaju uvjete. Ušće predstavlja poveznicu između destilerije i kondenzatora. Kondenzator pretvara pridošle podatke iz jednog podatkovnog modela u drugi. Sastoji se

od jednog ili više parsera koji mapira dijelove podatka iz jednog polja u drugo ili drugog kondenzatora. Destilerija specificira model podataka (preko kontejnera), lokaciju gdje će biti smješteni (preko kolekcije) i dodatne informacije za praćenje (preko kategorije).

- Poboljšavanje podataka (eng. „enhancing dana“): Naljepnice koriste inspekcije (eng. „inspections“) i procedure (eng. „procedures“) kako bi bolje opisale rafinirane podatke. Inspekcije se sastoje od koraka u kojem je svaki korak testiranje podataka preko postojećeg sita. Ako barem jedan od koraka vrati rezultat testiranja, rezultat se vraća i dodjeljuje naljepnici, a ako ne, tada se vraća „null“ vrijednost. Procedure su složeniji oblici testiranja i analize podataka, a sastoje se od protokola (eng. „protocols“). Svaki protokol sastoji se od naziva, paketa, modula i funkcije te mogu poslužiti za npr. utvrđivanje geolokacije IP adrese. Upravljanje rezultatom isto je kao i kod inspekcija.
- Konfiguriranje alarma (eng. „configuring alerts“): Postoje dvije vrste alarma: čuvari (eng. „watchdogs“) i monitori (eng. „monitori“). Čuvari generiraju notifikaciju ako podatak zadovoljava određeni skup uvjeta. Kada destilerija spremni rafinirane podatke, čuvari počinju sa analizom koja se sastoji od sita, prioriteta i razine opasnosti. Čuvari mogu imati i brnjicu (eng. „muzzle“) te na taj način neće generirati notifikaciju ako se utvrdi da je ona duplikat. Monitori generiraju notifikacije ako podaci nisu spremljeni prema očekivanoj vremenskoj stopi od strane destilerije.
- Istraživanje alarma (eng. „investigating data“): Istraživanje relacija između alarma radi se pomoću konteksta (eng. „context“). Prvo se definiraju dvije destilerije. Jedna je ona koja je spremila podatke zaslužne za notifikaciju alarma preko čuvara, a druga ukazuje na kolekciju unutar koje želimo naći tražene alarme. Uvjeti prema kojima će se naći alarmi su vremenski intervali te rezultati funkcija čiji će parametri biti podaci iz jedne i druge kolekcije.
- Upravljanje alarmima: Moguće je pregledati listu alarma, uz sortiranje prema poljima i traženje prema željenim vrijednostima polja, zatim promijeniti razinu, status, ishod i/ili korisnike koji su zaduženi za analizu alarma.
- Upravljanje akcijama nakon alarma: Akcije mogu pozvati REST API za obavljanje određene operacije, u odnosu na rezultate istraživanja nad alarmima. U trenutku pisanja ovog diplomskog rada, postoji samo JIRA integracija. Odgovor od API-ja naziva se otprema (eng. „dispatch“) koja sadrži referencu na alarm, podatke koji su vraćeni od API-a i pečat (eng. „stamp“) API poziva. Pečat sadrži podatak koji korisnik je pozvao API, vremenski početak i kraj API poziva i putovnicu (eng. „passport“) koja se koristila za autentifikaciju poziva.
- Dodavanja i uređivanje korisnika i grupa.
- Konfiguracija push notifikacija (u korelaciji sa Firebase servisom).
- Konfiguracija e-mail notifikacija.
- Konfiguracija autentifikacijiskih elemenata poput putovnice i vize.

- Konfiguracija automatskih mehanizama koji vrše interakciju sa izvorima podataka – kuriri i vodoinstalateri.



Slika 58: Izvor, analiza, mjenjanje i destinacija podataka [36]

Od funkcionalnosti koje pruža klijentska strana (Cyclops), tu su:

- Prikaz radne ploče i generalnih podataka: Prikazuje ukupni broj alarma, zatim broj alarma prema prioritetu, statusu, te pripadnosti u kolekciji. Grafički se prikazuje još i fluktuacija broja alarma kroz dane prema prioritetu. Na kraju tu je i karta svijeta koja pokazuje POI-je prema geopodacima (npr. iz IP adresa).
- Prikaz i upravljanje postojećim alarmima: Prikazuje listu svih alarma, omogućuje filtriranje liste prema kategoriji, prioritetu, statusu, izvoru podataka, datumu i pripadnosti korisniku. Za svaki označeni element liste, prikazuju se detalji alarma, rafinirani podaci pridruženi tom alarmu, a mogu se dodijeliti korisnici i akcije.

Cyphon je pisan u Python3 skriptnom jeziku te koristi Django web okvir. Aplikacija je smještena na Nginx web poslužitelju. Klijentski dio pisan je u ReactJS Typescript okviru i koristi d3.js. Od ostalih komponenti, tu su: Elasticsearch ili MongoDB te PostgreSQL kao baze podataka za potreban rad, RabbitMQ kao broker, a Logstash kao agregator poruka. Opis platforme, instalacija i korištenje detaljno su napisani na službenim Read the Docs stranicama Cyphon platforme na adresi: http://cyphon.readthedocs.io/en/latest/index.html

13. Pomoćni alati za upravljanje i odziv sigurnosnim incidentima

Uz nabrojane alate od kojih su neki bili više usmjereni na forenziku, neki na integraciju s SIEM-om, dok neki na čistu kolaboracijsku platformu, postoje i alati koje se ne može jednostavno svrstati u neku od navedenih kategorija, a mogli bi pomoći sigurnosnim timovima u upravljanju i odzivu na sigurnosne incidente. U nastavku će biti navedena još četiri takva alata: threat_note, nightHawk Response, CIRTKit, MISP

13.1. threat_note

Sigurnosni stručnjaci često koriste uredničke alate za vođenje istraga i rješavanje incidenata, gdje je potrebno voditi popriličan broj bilježaka, povezivati nove informacije sa starim, spremati takve informacije na strukturirani način kako ne bi nastao nered. Najčešći takvi alati: Notepad, Notepad++, OneNote, Evernote, itd. Spajanje sa postojećim sigurnosnim alatima ili čak pisanje njihovih proširenja kako bi ti alati podržavali funkcionalnost vođenja spomenutih bilježaka, često je kompliciranа te troši previše vremenskih (ponekad i novčanih) resursa. Ipak, postoji jednostavna web aplikacija koja omogućuje lagano i jasno vođenje bilježaka o domenama, IP adresama, napadačima (eng. „Threat Actor“) i kampanjama, a zove se threat_note. Otvorenog je programskog koda pod „Apache License 2.0“ licencom, a platformu je razvio Brian Warehime. Ovisno o spomenutim elementima, korisnik popunjava informacije i vodi bilješke koji su specifične baš za taj element. Postoji čak i integracija sa web uslugama poput whois, VirusTotal te OpenDNS Investigate, gdje pokretanjem usluge sa određeni element, threat_note prikazuje rezultate upita. Ova funkcionalnost može poboljšati kolaboraciju između stručnih timova. Alatna traka sastoji se od sljedećih prozora:

- „Dashboard“: prozor sa statistikama, vizualizacijama... uglavnom sveobuhvatni pogled na zapisane elemente.
- „Network Indicators“: prozor koji prikazuje tablicu sa stvorenim elementima poput IP adresa, domena i općenito mrežnih indikatora. Svaki indikator može se uređivati i brisati. Odabirom jednog elementa prikazuju se sve informacije o njemu (one koje je zapisao korisnik te one koje je otkrila pokrenuta web usluga za indikator). Prozor također sadrži opciju za stvaranje novog elementa.
- „Threat Actors“: prozor analogan „Network Indicators“, samo što prikazuje tablicu sa stvorenim elementima koji predstavljaju napadače ili prijetnje.
- „Campaigns“: Prozor sa listom kampanja, gdje svaka kampanja sadrži indikatore (mrežni indikatori te napadači i prijetnje) koji su povezani sa njom. Kampanje je u ovom prozoru moguće stvarati, mijenjati i brisati.
- „Settings“: Prozor unutar kojeg je moguće brisati sve podatke (reset baze podataka), definirati HTTP(S) Proxy adresu i konfigurirati web usluge poput njihova uključivanja i

isključivanja te API ključeva.

threat_note je većinski pisan u Python skriptnom jeziku, a baza podataka koja se koristi za pohranu svih radnih podataka je SQLite. Instalacija i primjeri korištenja napisani su na službenoj GitHub stranici koja se nalazi na adresi https://github.com/defpoint/threat_note

13.2. NightHawk Response

Još jedan specifičan alat za sve sigurnosne stručnjake koji se bave forenzikom i/ili odzivom na sigurnosne incidente. NightHawk Response je platforma za čitanje Mandiant Redline datoteka, tj. kolekcija linija dobivena trijažom operacijskog sustava uz pomoć Redline alata. Otvorenog je programskog koda, licenca nije poznata, a platformu su razvili Daniel Eden i Roshan Maskey. Iako instalacija ove platforme nije komplicirana, sa novom inačicom gdje je web sučelje većinski promijenjeno, pronađeno je dosta dizajnerskih grešaka koje čine platformu pomalo nepreglednom za rad. Popis funkcionalnosti koje nudi NightHawk Response:

- Stvaranje slučaja: kako bi se mogli čitati podaci .man datoteke, potrebno je stvoriti slučaj (eng. „case“) unutar kojega je potrebno prenjeti .man datoteku i imenovati slučaj.
- Prikaz sadržaja slučaja u obliku stabla: Nakon prijenosa .man datoteke, moguće je koristiti navigaciju u obliku stabla po datoteci. Svaka krajnja točka koja sadrži zapise prikazat će se u središtu prozora, u obliku tablice, a moguća je pretraga po željenim pojmovima, sortiranje te paginacija.
- Prikaz interaktivnog stabla procesa: ako se radi o procesima, prijašnje spomenute zapise moguće je prikazati interaktivnom mentalnom mapom. Na taj način sigurnosni stručnjak može dobiti drugačiji prikaz za lakše snalaženje, budući da se odmah ne prikazuju svi podaci o procesima, nego tek dok ih se označi (npr. prolazom miša).
- Prikaz nastanka/izvršavanja procesa u kronološkom nizu.
- Prikaz zapisa temeljem agregacije: za razliku od prikaza zapisa jedne krajnje točke, moguće je kombinirati više takvih točaka. Na taj način sučelje dodaje novi stupac „Count“ koji predstavlja ukupan broj prikazanih zapisa. Ovakav prikaz ne sadrži duplikate zapisa (ključni stupac po kojemu se određuju duplikati je „Attributes“).
- Globalna pretraga željenog zapisa.
- Obogaćivanje zapisa proizvoljnim podacima: korisnik platforme može nadoknadno obogatiti zapis novim podacima poput komentara, imena analitičara, datuma te oznakama poput „zlonamjerno“, „indikator kompromisa“, „lažni indikator“, itd. Postoji posebni prozor unutar kojeg je moguće vidjeti sve obogaćene zapise. Napomena: upisani podaci nisu spremljeni u prenešenu datoteku, već su svi zapisi sa prenesene datoteke + ručno upisani (novi) podaci spremljeni unutar baze podataka ove platforme.

NightHawk Response za poslužiteljsku stranu koristi Python (Django) i Go Lang, dok za klijentsku stranu koristi AngularJS i d3.js/jsTree. Od ostalih komponenata tu su Elasticsearch

kao baza podataka, Nginx sa uWSGi kao web poslužitelj te Kibana i RabbitMQ. Instalacija i primjeri korištenja napisani su na službenoj GitHub stranici koja se nalazi na adresi https://github.com/biggiesmallsAG/nightHawkResponse

13.3. CIRTKit

Uz Osquery, CIRTKit je još jedan alat za sigurnosne stručnjake koji pruža CLI za rad svojim korisnicima. Točnije, CIRTKit je skup alata predstavljenih u DFIR (eng. „Digital Forensics and Incident Response“) SANS tečaju, a domena alata je pružanje forenzičkih sposobnosti i provjere datoteka na web uslugama poput VirusTotal-a, Cuckoo Sandbox, Passive DNS, itd. Otvorenog je programskog koda pod MIT licencom, a platformu je razvio Bob "byt3smith" Argenbright. Ono što čini CIRTKit zanimljivim je minimalizam i središnja točka za provođenje istraga. Inspiracija za stvaranje CIRTKit-a je stvaranje ažuriranog i kvalitetnog arsenala za „Blue Team“, kao što je Metasploit okvir za „Red Team“. CIRTKit dijeli istu arhitekturu kao i Viper Binary Analysis okvir: moduli kao alati za rad unutar istrage, projekti (koristi se pojam „istraga“) kao glavni element i sesije kao objekti spremanja stanja. Pokretanje CIRTKit alata radi se izvršavanjem Python skripte sa zastavicom „-i“ i nazivom istrage. Moguće su sljedeće funkcionalnosti, a vidljive su upisom „help“ u konzolu CIRTKit-a:

- Upravljanje datotekom: pronalaženje, otvaranje, stvaranje, analiza, spremanje stanja i brisanje datoteke – sve to moguće je raznim kombinacijama komandi koje nudi CIRTKit.
- Upravljanje sesijama: rad unutar CIRTKit-a na istrazi ne mora biti „izgubljen“ nakon izlaza iz alata. Sesija kao mehanizam praćenja i spremanja trenutnog stanja postoji i može se koristiti uz pomoć ponuđenih komandi.
- Upravljanje bilješkama: pregled, stvaranje i uređivanje bilježaka koje se tiču trenutne istrage.
- Korištenje modula i integracija u istrazi: CIRTKit trenutno broji 27 modula i jednu integraciju (CarbonBlack Live Response). Moduli su glavna jedinica analize tokom rada nad istragom. Neki od modula su Cuckoo Sandbox, ELF, JAR parser, EML/MSG parser, MISP upload, PDF analyzer, Radare2, strings program, VirusTotal, YARA rules, itd.
- Pregled statistika: u obliku tablice mogu se vidjeti ukupna količina istraga, MIME tipova datoteka, oznaka, veličina statistika u bajtovima, itd.

CIRTKit je većinski pisan u Python skriptnom jeziku, a baza podataka koja se koristi za pohranu svih radnih podataka je SQLite ili PostgreSQL, ovisno o instalaciji i konfiguraciji od strane korisnika. Dokumentacija je jako skromna, kvalitetno su objašnjenja jedino instalacija i motivacija nastanka ovog alata. Službena GitHub stranica projekta nalazi se na adresi https://github.com/opensourcesec/CIRTKit

13.4. MISP Threat Sharing

Tokom prve cjeline spomenuo se pojam „Threat Sharing Platform“, kao dio „Security Intelligence“-a, odnosno radne komponente SOC odjela. „Malware Information Sharing Platform and Threat Sharing“, skraćeno MISP, glasi kao rješenje za sakupljanje, pohranjivanje, distribuciju i dijeljenje „Threat Intelligence“-a pronađenim tokom analize malicioznih programskih kodova i općenito incidenata diljem različitih organizacija. Otvorenog je programskog koda, pod licencom „GNU Affero General Public License version 3“ , a platformu je napravila šira zajednica: NATO/NCIRC, CIRCL, Belgian Defence, Christophe Vandeplas, Andras Iklody, Andreas Ziegler. Platforma je namijenjena svim sigurnosnim stručnjacima, SOC i CSIRT odjelima, kako bi sa njima dijelila indikatore kompromisa i informacije od svježim incidentima iz dana u dan. Funkcionalnosti MISP-a su:

- Pohrana indikatora kompromisa u bazu podataka na učinkoviti način, dajući korisnicima brzi i strukturiran pregled kada im je to potrebno.
- Automatska korelacija veza između pohranjenih indikatora kompromisa, radi dobivanja dodatnih informacija. Primjeri korelacija uključuju „ssdeep“ (povezivanje kriptografskih potpisa) i uklapanje CIDR blokova.
- Pružanje fleksibilnog modela podataka gdje se kompleksni objekti mogu jednostavnije prikazati, pa i međusobno povezati.
- Mogućnosti dijeljenja vlastitog „Threat Intelligence“-a kroz različite modele i politike dijeljenja.
- Izvoz podataka u raznim formatima poput običnog teksta, CSV, XML, JSON, OpenIOC, IDS, itd. To omogućuje integraciju MISP-a sa alatima poput IDS-a, STIX-a, itd.
- Uvoz podataka u paketima ili zasebno, s podrškom mnogih formata poput CSV, „Sandboxing“ formata, itd.
- Omogućena kolaboracija kroz web sučelje za korisnike MISP-a, u svrhu obogaćivanja ili ažuriranja postojećih podataka.
- Korištenje API-ja za razne integracije s drugim uslugama.
- „Threat Intelligence“ rječnici koji se sastoje od predefiniranih podataka o APT-ovima, malicioznom programskom kodu, zatim „Threat Intelligence“ sa MITRE ATT&CK, itd. Svi podaci se automatski koreliraju sa pristiglim događajima i tako se svojevrsno obogaćuju.

MISP je većinski pisan u Python skriptnom jeziku. Kao bazu podataka koristi MySQL, a moguće je koristiti Nginx poslužitelj kao proxy. Implementirana je enkripcija između platforme i svih izvora/odredišta podataka. Zajednica je aktivna, postoji nekoliko načina instalacije platforme čije skripte i dokumentaciju održavaju sami korisnici (kvaliteta pisanja i održavanja varira). Korisničke upute su detaljno pisane i dostupne u nekoliko digitalnih formata. Spomenute poveznice i opis platforme nalazi se na službenoj GitHub stranici MISP-a: https://github.com/MISP/MISP

14. Zaključak

Unutar ovog diplomskog rada objašnjeni su najbitniji pojmovi za razumijevanje teme upravljanja i odziva na sigurnosne incidente. Nakon toga opisana su i objašnjena dva glavna resursa – okvir za strukturu i format praćenja incidenata te okvir, odnosno preporučeni životni ciklus za izradu vlastitih planova i metodologija za upravljanje incidentima. Zatim su navedene i opisane najpoznatije javno dostupne metodologije i kolekcije vodiča koje implementiraju te metodologije, za sve one korisnike koji žele pratiti najbolje prakse. Ostatak ovog rada čine opisi i recenzije alata otvorenog programskog koda za upravljanje i odziv na sigurnosne incidente. Opisano je 15 alata, koji su kategorizirani prema njihovoj glavnoj namjeni, dok su tri odabrana alata detaljnije opisana, poput postupka instalacije i primjera konfiguracije.

Upravljanje i odziv na sigurnosne incidente, prema mišljenju autora ovog rada, najzanimljivija je domena unutar informacijske sigurnosti. Iako na prvi pogleda izgleda kao posao isključivo za „Blue Team“, odnosno stručne timove koji pojačavaju zaštitu, potrebna je široka koordinacija, ne samo sa „Red Team“-om (identifikacija ranjivosti, lov na napadača, testiranje sustava na propuste...) već i sa ostalim odjelima u organizaciji. Svaki zaposlenik može biti u istom trenutku i žrtva i asistent rješavanju incidenta. Sljedeće što na prvi pogled ne izgleda očito je važnost određenih faza unutar navedenih metodologija, odnosno pripreme i primjene naučenog (prva i zadnja faza). Priprema određuje mogućnost pojave incidenta, dok primjena naučenog tjera stručne timove da ažuriraju svoje procedure i pristup tokom svih faza upravljanje i odziva na incidente.

Iako kategorizirani prema svojoj glavnoj svrsi, alati se mogu podijeliti prema „pripadnosti“ SOC i CSIRT timovima. Od forenzički orijentiranih platformi, najviše će koristi izvući CSIRT, dok će s druge strane SOC puno više koristiti orkestracijske platforme i platforme sa SIEM-om. Alati otvorenog koda, koji su dostupni za odziv na incidente, većinom ne prate ili ne implementiraju metodologije, a iznimke su MozDef i FIR. Uglavnom se korisnicima takvih alata daje velika sloboda podesivosti, dok kod nekoliko alata postoji i velika krivulja učenja, poput Cyphon i RTIR. Što se tiče tehničkih detalja, gotovo svi alati imaju mogućnost instalacije i održavanja unutar kontejnera, a postoje i alati koji su osmišljeni za samo jedan operacijski sustav ili sigurnosni proizvod, poput CrowdStrike Falcon Orch., Zentral i NightHawk. Zaključno, na „tržištu“ alata za upravljanje i odziv na incidente postoji niz kvalitetnih rješenja koja su otvorenog koda i koja su prepoznata u mnogim sigurnosnim organizacijama te se koriste redovito u poslovanju.

Popis literature

[1] M. Borrelli, *Malware and Computer Security Incidents - Handling Guides*. Nova, 2013.

[2] L. J. Mike Kessler, *Computer incident response and forensics team management - conducting a successful incident response*. Syngress, 2014.

[3] K. M. Jason Luttgens Matthew Pepe, *Incident Response & Computer Forensics*. McGraw Hill, 2014.

[4] D. Rajnović, *Computer Incident Response and Product Security (Networking Technology and Security)*. McGraw Hill, 2010.

[5] M. Rouse. (). Security Incident, adresa: https://whatis.techtarget.com/definition/security-incident. (pristupano: 19.08.2018).

[6] (). Incident Management, adresa: https://en.wikipedia.org/wiki/Incident_management. (pristupano: 19.08.2018).

[7] (). Computer Security Incident Management, adresa: https://en.wikipedia.org/wiki/Computer_security_incident_management. (pristupano: 19.08.2018).

[8] R. B. Scott J. Roberts, *Intelligence-Driven Incident Response - Outwitting the Adversary*. O'Reilly, 2017.

[9] A. de Beaupre. (). Incident Response vs. Incident Handling, adresa: https://isc.sans.edu/forums/diary/Incident+Response+vs+Incident+Handling/6205/. (pristupano: 19.08.2018).

[10] M. Rouse. (). Incident Response, adresa: https://searchsecurity.techtarget.com/definition/incident-response. (pristupano: 19.08.2018).

[11] G. Messina. (). How to Develop an Incident Response Plan in 9 Simple Steps, adresa: https://resources.infosecinstitute.com/develop-incident-response-plan-9-simple-steps/. (pristupano: 19.08.2018).

[12] (). Security Awareness Training - A critical component of a healthy security program, adresa: https://www.rapid7.com/fundamentals/security-awareness-training/. (pristupano: 19.08.2018).

[13] N. Lord. (). What is a Security Operations Center (SOC)?, adresa: https://digitalguardian.com/blog/what-security-operations-center-soc. (pristupano: 19.08.2018).

[14] M. Rouse. (). Computer Security Incident Response Team (CSIRT), adresa: https://whatis.techtarget.com/definition/Computer-Security-Incident-Response-Team-CSIRT. (pristupano: 19.08.2018).

[15] ——, (). Security Intelligence, adresa: https://whatis.techtarget.com/definition/security-intelligence-SI. (pristupano: 19.08.2018).

[16] ——, (). Threat Intelligence (cyber threat intelligence), adresa: https://whatis.techtarget.com/definition/threat-intelligence-cyber-threat-intelligence. (pristupano: 19.08.2018).

[17] ——, (). Threat Intelligence Feed (TI feed), adresa: https://whatis.techtarget.com/definition/threat-intelligence-feed. (pristupano: 19.08.2018).

[18] (). What is a Threat Intelligence Platform (TIP)?, adresa: https://www.anomali.com/platform/what-is-a-tip. (pristupano: 19.08.2018).

[19] ——, (). Intrusion Prevention, adresa: https://searchsecurity.techtarget.com/definition/intrusion-prevention. (pristupano: 19.08.2018).

[20] T. Marden. (). Making Sense of Information Security Technologies: IDS/IPS, UTM, and SIEM, adresa: https://blog.cygilant.com/blog/making-sense-of-information-security-technologies-ids-ips-utm-and-siem. (pristupano: 19.08.2018).

[21] M. Rouse. (). Security Information and Event Management (SIEM), adresa: https://searchsecurity.techtarget.com/definition/security-information-and-event-management-SIEM/. (pristupano: 19.08.2018).

[22] M. V. Jeff Bollinger Brandon Enright, *Crafting the InfoSec Playbook - Security Monitoring and Incident Response Master Plan*. O'Reilly, 2015.

[23] M. Rouse. (). Forensic, adresa: https://whatis.techtarget.com/definition/forensic. (pristupano: 19.08.2018).

[24] ——, (). Network Forensics, adresa: https://searchsecurity.techtarget.com/definition/network-forensics. (pristupano: 19.08.2018).

[25] ——, (). Computer Forensics (cyber forensics), adresa: https://searchsecurity.techtarget.com/definition/computer-forensics. (pristupano: 19.08.2018).

[26] ——, (). Verizon VERIS, adresa: https://searchsecurity.techtarget.com/definition/Verizon-VERIS-Vocabulary-for-Event-Recording-and-Incident-Sharing-Framework. (pristupano: 25.08.2018).

[27] K. Johns. (). Consistency in Reporting Data Breaches, adresa: https://slideplayer.com/slide/8463282/. (pristupano: 25.08.2018).

[28] V. Community. (). VERIS - the vocabulary for event recording and incident sharing, adresa: http://veriscommunity.net/. (pristupano: 25.08.2018).

[29] J. Fritz. (). Incident Response Methodology: The OODA Loop, adresa: https://www.alienvault.com/blogs/security-essentials/incident-response-methodology-the-ooda-loop. (pristupano: 28.08.2018).

[30] H. enciklopedija. (). NIST, adresa: http://www.enciklopedija.hr/natuknica.aspx?ID=43895. (pristupano: 28.08.2018).

[31] S. F. A. Q. (faq). (). What is the SANS Institute?, adresa: http://www.sans.org/faq/#faq67. (pristupano: 28.08.2018).

[32] S. Institute. (). SANS Institute InfoSec Reading Room, adresa: https://www.sans.org/reading-room/whitepapers/incident/incident-handlers-handbook-33901. (pristupano: 28.08.2018).

[33] certsocietegenerale. (). Incident Response Methodologies, adresa: https://github.com/certsocietegenerale/IRM/. (pristupano: 28.08.2018).

[34] J. L. Thomas Franco Saâd Kadhi. (). TheHive, adresa: https://github.com/TheHive-Project/TheHive. (pristupano: 11.07.2018).

[35] J. Bryner. (). MozDef, adresa: http://mozdef.readthedocs.io/en/latest. (pristupano: 01.08.2018).

[36] D. Cybersecurity. (). Cyphon, adresa: http://cyphon.readthedocs.io/en/latest. (pristupano: 04.08.2018).

Popis slika