



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INFORMAČNÍCH SYSTÉMŮ
DEPARTMENT OF INFORMATION SYSTEMS

MODUL NA ZPRACOVÁNÍ DAT ZE SYSTÉMU SURICATA PRO IBM QRADAR

SURICATA MODULE FOR IBM QRADAR

BAKALÁRSKA PRÁCA
TERM PROJECT

AUTOR PRÁCE
AUTHOR

MARTIN KOZÁK

VEDÚCI PRÁCE
SUPERVISOR

Ing. RADEK HRANICKÝ, Ph.D.

BRNO 2023

Zadání bakalářské práce



147208

Ústav: Ústav informačních systémů (UIFS)
Student: **Kozák Martin**
Program: Informační technologie
Specializace: Informační technologie
Název: **Modul na zpracování dat ze systému Suricata pro IBM QRadar**
Kategorie: Bezpečnost
Akademický rok: 2022/23

Zadání:

1. Seznamte se se systémy IBM QRadar SIEM a Suricata IDS.
2. Nasadíte systémy IBM QRadar ve virtualizovaném prostředí
3. Po domluvě s vedoucím navrhněte modul typu DSM pro zpracování zpráv ze systému Suricata.
Zaměřte se na identifikaci jednotlivých typů zpráv a informace, které jsou zajímavé z hlediska detekce bezpečnostních incidentů a monitoringu provozu na síti.
4. Navrhněte také jednoduché rozšíření typu Add-on pro vizualizaci informací získaných z vašeho DSM.
5. Navržené subsystémy implementujte.
6. Demonstrujte použitelnost vašeho řešení při zpracování vybraných typů síťového provozu.
7. Zhodnoťte dosažené výsledky.

Literatura:

- Chakrabarty, B., Patil, S. R., Shingornikar, S., Kothekar, A., Mujumdar, P., Raut, S., & Ukirde, D. (2021). *Securing Data on Threat Detection by Using IBM Spectrum Scale and IBM QRadar: An Enhanced Cyber Resiliency Solution*. IBM Redbooks.
- S. Gupta, B. S. Chaudhari and B. Chakrabarty, "Vulnerable network analysis using war driving and security intelligence," *2016 International Conference on Inventive Computation Technologies (ICICT)*, 2016, pp. 1-5, doi: 10.1109/INVENTIVE.2016.7830165.
- Eldow, O., Chauhan, P., Lalwani, P., & Potdar, M. (2016). Computer network security ids tools and techniques (snort/suricata). *Int. J. Sci. Res. Publ*, 6(1), 593.

Při obhajobě semestrální části projektu je požadováno:

Body 1 až 4

Podrobné závazné pokyny pro vypracování práce viz <https://www.fit.vut.cz/study/theses/>

Vedoucí práce: **Hranický Radek, Ing., Ph.D.**

Vedoucí ústavu: Kolář Dušan, doc. Dr. Ing.

Datum zadání: 1.11.2022

Termín pro odevzdání: 10.5.2023

Datum schválení: 31.10.2022

Abstrakt

Táto práca integruje program Suricata do systému QRadar. Hlavným cieľom práce je navrhnuť a implementovať modul DSM v systéme QRadar, ktorý dokáže analyzovať záznamy z programu Suricata. Udalosti je možné následne vyšetrovať v prostredí systému QRadar. Ďalším cieľom je navrhnúť aplikáciu, ktorá zobrazí dátá detegované programom Suricata. Aplikáciu je možné nainštalovať do prostredia QRadar a použiť ju aj s ďalšími vstavanými komponentami. Aplikácia je naprogramovaná v knižnici Flask s použitím šablón Jinja2. Súčasťou aplikácie sú dve karty, ktoré zobrazujú udalosti v rôznych grafoch a tabuľkách.

Abstract

This work integrates the Suricata program into the QRadar system. The main objective of this work is to design and implement a DSM module in QRadar system that can analyze the records from Suricata. The events can then be investigated in the QRadar system environment. Another objective is to design an application that displays the data detected by the Suricata program. The application can be installed in the QRadar environment and used with other built-in components. The application is programmed in the Flask library using Jinja2 templates. The application includes two tabs that display events in different graphs and table

Kľúčové slová

QRadar, SIEM, DSM, Suricata

Keywords

QRadar, SIEM, DSM, Suricata

Citácia

KOZÁK, Martin. *Modul na zpracování dat ze systému Suricata pro IBM QRadar*. Brno, 2023. Bakalárská práca. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedúci práce Ing. Radek Hranický, Ph.D.

Modul na zpracování dat ze systému Suricata pro IBM QRadar

Prehlásenie

Prehlasujem, že som túto bakalársku prácu vypracoval samostatne pod vedením pána Ing. Radka Hranického Ph.D. V práci som uviedol všetky literárne pramene, publikácie a iné zdroje, z ktorých som čerpal.

.....
Martin Kozák
10. mája 2023

Poděkování

Veľmi rád by som sa poděkoval môjmu vedúcemu bakalárskej práce Ing. Radekovi Hranickému Ph.D za pomoc v tejto práci.

Obsah

1	Úvod	3
2	Systémy IDS a program Suricata	4
2.1	Typy systémov podľa metodík detekcie narušenia	4
2.2	Typy systémov IDS	5
2.3	Suricata	5
3	Qradar – systém SIEM	8
3.1	IBM QRadar – Všeobecné informácie	9
3.2	Priebeh spracovania udalostí	9
3.3	Konfigurácia DSM	10
3.4	Mapovanie udalostí na kategórie nízkej úrovne	11
3.5	Ukladanie informácií – Ariel Databáza a REST API	12
4	Návrh modulu DSM pre program Suricata	15
4.1	Základné údaje	16
4.2	Vlastné údaje	19
4.3	Kategorizácia v rámci QRadaru	24
5	Návrh addonu – vizualizácia dát z modulu DSM	28
5.1	Použité technológie – webová knihovna Flask a nástroj Jinja2	29
5.2	Vizualizácia dát	29
6	Implementácia DSM modulu a aplikácie pre systém Qradar	34
6.1	Virtualizácia systému Qradar a nastavenie programu Suricata	34
6.2	Implementácia modulu DSM	34
6.3	Implementácia aplikácie	36
6.4	Exportovanie DSM modulu a nasadenie Addonu	43
7	Demonštrácia použiteľnosti pri vybraných sieťových prevádzkach	44
7.1	Demonštrácia bezpečnej obvykľej prevádzky	45
7.2	Demonštrácia analýzy nebezpečnej prevádzky	45
7.3	Zhodnotenie	49
8	Záver	51
Literatúra		52
Zoznam príloh		54

A Obsah priloženého pamäťového média	55
B Manuál na inštaláciu modulu DSM a aplikácie pre systém QRadar	56
B.1 Inštalácia modulu DSM	56
B.2 Inštalácia rozšírenia pre QRadar	56
C Snímky obrazovky z aplikácie	58
C.1 Snímky obrazovky z karty hrozby	58
C.2 Snímky obrazovky z karty správy	62

Kapitola 1

Úvod

Internet, ktorý vznikol v minulom storočí, priniesol so sebou množstvo výhod, ktoré zlepšili úroveň života ľudstva. Avšak vznikom nového virtuálneho priestoru sa objavili aj nové hrozby, ako napríklad útoky na citlivé informácie a narušenie rozsiahlych zdravotníckych alebo štátnych systémov. V takýchto veľkých štruktúrach je potrebná implementácia rôznych bezpečnostných systémov. Táto práca sa venuje prepojeniu systému na detegovanie hrozieb a systému, ktorý tieto dátu dokáže ukladať a viesť nad nimi hlbšiu analýzu.

Na snímanie sietí sa v momentálnej dobe najviac využívajú systémy IDS (angl. Instruction Detection System). Medzi najznámejšie open-source programy patrí program Suricata. Účelom nasadenia tohto systému je skenovať a ďalej preposielat získané informácie o možných hrozbách. Na monitorovanie a správu bezpečnostných udalostí v sieťach sa využíva systém SIEM (Security Information and Event Management), medzi ktorými je najznámejší QRadar od spoločnosti IBM. Program je veľmi obsiahly a zahŕňa mnoho súčasťí na monitoring.

Systém QRadar poskytuje možnosť analyzovať záznamy z externých zdrojov, medzi ktorými sa však nenachádza program Suricata. Jedným z hlavných cieľov tejto bakalárskej práce je vytvoriť návrh nového modulu DSM pre program Suricata. Modul by mal dokázať zanalyzovať prichádzajúce hrozby, ktoré systém následne uloží. Ďalším cieľom je vytvoriť aplikáciu na vizualizáciu dát z programu Suricata uložených v systéme QRadar.

V tejto práci sa najprv podarilo navrhnúť a implementovať modul DSM. Boli námavané základné pravidlá IDS systému Suricata slúžiace na kategorizáciu udalostí v rámci systému QRadar. Pre zobrazenie dát bola vyvinutá aplikácia – dashboard, ktorá slúži ako rozšírenie do systému QRadar a môže byť spolu s modulom DSM voľne stiahnutá. Funkčnosť aplikácie aj modulu DSM bola overená nad nebezpečným tokom dát. Modul DSM aj aplikácia je prínosná pre užívateľov, ktorí pracujú so systémom QRadar a v sieti majú uložený monitorovací systém IDS Suricata.

Práca je členená na 6 kapitol. Po úvode nasleduje opis systémov IDS a Suricaty v kapitole 2. V kapitole 3 je opísaný systém QRadar a jeho rozšírenie DSM. Nasleduje návrh DSM systému a pre program Suricata v kapitole 4. V kapitole 5 je popísaný návrh rozšírenia, ktoré umožní vizualizáciu spomínaných dát. V ďalšej kapitole 6 je popísaná implementácia modulu DSM a aplikácie na vizualizáciu. V predposlednej kapitole 7 je uvedená demonštrácia aplikácie. Poslednou kapitolou je záver 8.

Kapitola 2

Systémy IDS a program Suricata

Pre pochopenie riešenej problematiky a získanie potrebných teoretických východísk, je potrebné vysvetliť, ako funguje program Suricata. V komplexe bezpečnostných systémov a prvkov existujú systémy, ktoré sa zaoberajú detekciou útokov a hrozieb, hrozby však následnej neanalyzujú. Slúžia ako základný prvok pri identifikácii hrozieb a získavaní informácií o nich. Sú to systémy detekcie prienikov (angl. Intrusion Detection System – IDS). Princípom systémov IDS je monitorovať a analyzovať správanie v sieti. Pri nasadení nepriamo slúžia ako ochranný prvok, keďže odosielajú informácie o hrozbách, ale samé o sebe nedokážu zabráňovať týmto útokom.

2.1 Typy systémov podľa metodík detekcie narušenia

Hlavným zmyslom IDS systémov je detegovať rôzne typy hrozieb a možných narušení. V priebehu vývoja vzniklo viacero spôsobov, ktoré aplikujú dva hlavné prístupy pri detekcii hrozieb [6].

Systém založený na detekcii signatúr

Signatúra je dopredu pripravený vzor, ktorý korešponduje so známou hrozbou. V procese zisťovania hrozieb sa predpripravené signatúry porovnávajú s pozorovanými udalosťami za účelom detekcie nechcených hrozieb. Táto technika je rýchla a ľahko konfigurovateľná. Zároveň efektívne deteguje možné hrozby, ktoré sú dopredu známe. Ak však ide o neznámu hrobu, je takmer nemožné ju týmto spôsobom odhaliť. Z tohto dôvodu možno využiť tejto techniky pokladáť za obmedzené. Techniku založenú na detekcii signatúr využíva práve spomínaný program Suricata.

Systém založený na detekcii anomálií

Nedostatky, ktorými disponuje predošlá metóda, rieši proces detekcie anomálií. V tomto procese sa porovnávajú definície normalizovaných aktivít s prichádzajúcimi aktivitami s cieľom získať signifikantný rozdiel. Takýto systém má predpripravené profily reprezentujúce normálne správanie užívateľov, hostí, internetových pripojení alebo aplikácií. Profily vznikajú časom na základe typického chovania jednotlivých prvkov v sieti. Hlavnou výhodou tejto metódy je možná detekcia hrozieb, ktoré nemusia byť známe alebo sa čiastočne odchyľujú od očakávaného správania. V porovnaní s detekciou signatúr, dokáže táto technika

zistiť hrozby, na ktoré by inak systém pripravený neboli. Napríklad dokáže detegovať hrozbu, ktorá priamo nemusí byť hrozbou, ale nesie v sebe prvky, ktoré nepôsobia štandardne.

2.2 Typy systémov IDS

IDS systémy je možné deliť aj podľa umiestnenia v sieti. Dvoma základnými typmi sú systémy nasadené na hostiteľoch a systémy nasadené priamo v sieti. Tieto dva typy systémov sú zobrazené na obrázku 2.1 a ich princíp fungovanie je vysvetlený nižšie [24].

Systém založený na hostiteľoch

Typ systému – HIDS (angl. Host-Based IDS) bol vynájdený ako prvý. Hlavná myšlienka spočíva v ochrane počítača, na ktorom je nainštalovaný. Systém dokáže chrániť len koncového užívateľa analyzovaním internetového rozhrania a systémových správ. Výhody tohto systému sú využité pri ochrane kritických prvkov v sieti, ako napríklad serverov a smerovačov. To, že musí byť nasadený na koncovom prvku v sieti, sa dá považovať za hlavnú nevýhodu tohto typu systému.

Systém založený na sieťových informáciach

Druhý základný typ – NIDS (angl. Network-based IDS) monitoruje sieťový ruch v sieti a analyzuje internetové a aplikačné protokoly. Najčastejšie je nasadený na hraniciach sietí, ktoré sú prepojené prvkami ako smerovače alebo virtuálne súkromné siete. Hlavnou nevýhodou systému NIDS je fakt, že nedokáže správne chrániť jednotlivých hostiteľov. Monitoruje však celú sieť a sieťový ruch cieľových hostiteľov je bez zníženia výkonu.

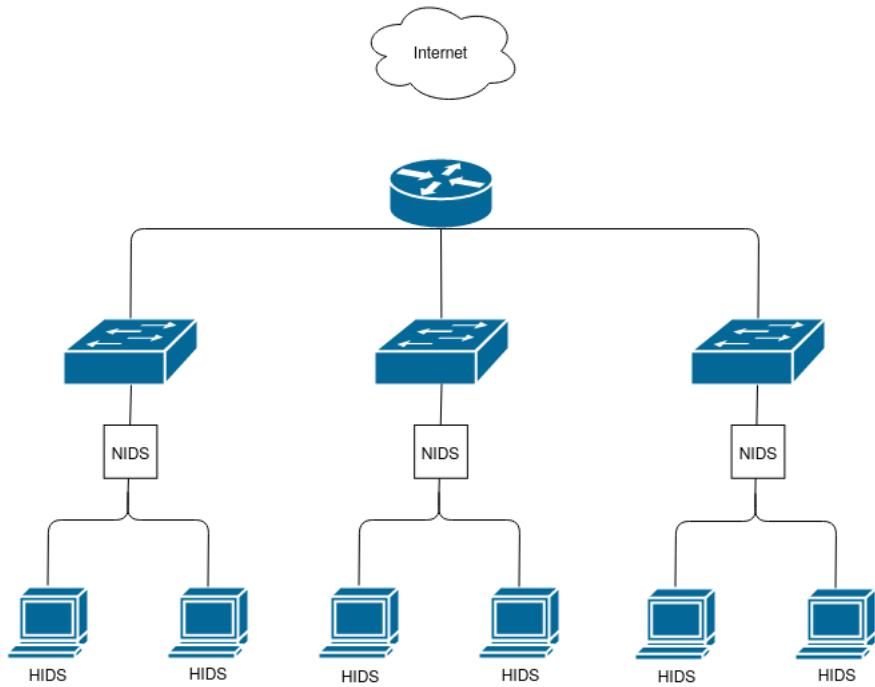
Fakt, že systémy IDS nedokážu zabráňovať útokom, priniesol obmenu systému. Systém na prevenciu prieniku (angl. Intrusion Prevention Systems – IPS) dokáže detegovaným hrozbám aj zabrániť a eliminovať útok. Môže byť považovaný za rozšírenie pre systémy IDS s funkcionalitami, ktoré dokážu chrániť počítače a zariadenia v sieti. V podstate ide o obdobu firewallu, rozdielom je spôsob zabránenia útoku. Firewally blokujú všetok sieťový prenos okrem toho, ktorý bol povolený v konfigurácii. Systémy IPS prepúšťajú všetku prevádzku, ale blokujú nebezpečnú prevádzku na základe pravidiel a signatúr, ktoré používa aj systém IDS [15].

2.3 Suricata

Program Suricata je jeden z novších systémov IPS/IDS, ktorého zámerom bolo vytvoriť systém IDS novej generácie s rozšíreniami podporujúcimi IPS. Ide o open-source program, ktorý využíva detekciu na základe signatúr. Bol založený na systéme Snort¹, preto s ním zdieľa rovnaké alebo podobné signatúry. Výhodou programu Suricata je, že umožňuje podporu multivláknového spracovania sieťového ruchu. Konfigurácia umožňuje nasadenie tohto systému ako NIDS alebo ako HIDS. Príklad signatúry – pravidla je zobrazený na obrázku 2.2.

V tomto príklade 2.2 má akcia červenú farbu, hlavička zelenú a modrou sú napísané možnosti daného pravidla. Signatúra je rozdelená do troch častí:

¹<https://www.snort.org/>



Obr. 2.1: Rozloženie HIDS a NIDS v sieti [7]

```
drop tcp $HOME_NET any -> $EXTERNAL_NET any (msg:"ET TROJAN Likely Bot Nick in IRC (USA
+..)"; flow:established,to_server; flowbits:isset,is_proto_irc; content:"NICK "; pcre:"/NICK
.*USA.*[0-9]{3,}/i"; reference:url,doc.emergingthreats.net/2008124; classtype:trojan-
activity; sid:2008124; rev:2;)
```

Obr. 2.2: Príklad signatúry v Suricata

- **Akcia** znamená, čo sa má stať s daným tokom/paketom ak sa zhoduje s pravidlom. Existujú dva hlavné typy akcie. Akcia typu **alert** používateľa iba informuje o hrozbe v sieti a akcia typu **drop** navyše nebezpečný paket aj zahodí. Táto možnosť existuje len v móde ISP.
- V **hlavičke** sú obsiahnuté informácie o zdrojových a cielových IP adresách a portoch, na ktorých je sledovaný tok dát.
- **Možnosti pravidla** definujú bližšie špecifikácie pre filtrovanie paketov. Definujú sa ako vlastnosti a sú oddelené bodkočiarkou. Medzi vlastnosťami je možné definovať meno pravidla, kategóriu alebo napríklad čas aktualizácie.

Program Suricata má preddefinované pravidlá, ktoré sú stiahnuté spolu s inštaláciou balíka. Rozdelené sú podľa toho, aké typy protokolov chránia. Program Suricata umožňuje definíciu aj vlastných pravidiel, no manuálna definícia môže byť časovo náročná. Preto program Suricata ponúka stiahnutie aktualizovaných balíkov pravidiel. Skript **suricata-update** vykonáva správu nad balíkmi pravidiel. Umožňuje stiahnuť a aktualizovať pravidlá aj z treťich strán.

Program Suricata generuje rôzne typy správ a upozornení, ktoré používateľom poskytujú informácie o tom, čo sa v sieti deje. Najdôležitejšimi správami sú informácie o hrozbách, ktoré sú generované na základe pravidiel. Ďalej dokáže generovať správy o anomaliách a informovať užívateľa o stave siete. Generuje tiež správy o rôznych použitých aplikačných protokoloch ako napríklad: DNS, HTTP alebo TLS.

Na samotnú konfiguráciu systému je použitý súbor vo formáte YAML, ktorý je preddefinovaný v `/etc/suricata/suricata.yaml`. Po stiahnutí obsahuje základnú konfiguráciu systému, ktorú je možné bližšie špecifikovať. Ponúka širokú paletu konfiguračných nastavení. Umožňuje nastaviť, aké typy správ sa budú v programe Suricata objavovať, na akých rozhraniach a s akými tipmi IP adres. V tomto súbore je ďalej možné nastaviť ako a čo sa bude nachádzať v záznamoch generovaných programom Suricata.

Program Suricata a Snort sú veľmi podobné systémy IDS. Vedia rovnako detegovať hrozby ale program Suricata má jednú výhodu. Dokáže pracovať v multivláknom móde a byť rýchlejší ako program Snort [12]. Program Suricata pozostáva z niekolkých blokov nazvaných moduly, ktoré sú spustené paralelne v odlišných procesoch. Jeden modul slúži napríklad na dekódovanie paketu, ďalší zas na detekciu hrozieb alebo na generovanie výstupu. Práve modul na detekciu hrozieb je najviac vyťažený spomedzi všetkých modulov. To ako budú pracovať jednotlivé moduly a posielat si navzájom pakety, definuje beh aplikácie.

V režime behu **single** existuje len jedno vlákno na spracovávanie paketov. Ďalším behom je **autofp**, v ktorom sa vytvoria samostatné vlákna pre dve skupiny modulov. Posledným variantom je **workers**, kde sa vytvárajú vlákna, ktoré ale spracovávajú celý paket samostatne a nepredávajú informácie do iných. Práve možnosť rozdelenia tokov do samostatných vlákiem, umožňuje veľmi rýchle spracovanie paketov [23].

Vzhľadom na cieľ, pre ktorý nasadzujeme program Suricata, je potrebné definovať, ktorý typ záznamového súboru sa využije. Systém ponúka možnosť voľby výstupu vo forme Eve, ktorý je vo formáte JSON. Následne je možné si zvoliť, kam sa bude výstup posielat a kde sa budú ukladať všetky záznamy programu Suricata (v základnom nastavení je to záznamový súbor `fast.log`). Na to, aby sme mohli spracovávať záznamy systémom QRadar, ktorý je popísaný v ďalšej kapitole, použijeme štandardné spracovávanie záznamov `syslog`. Program Suricata sa tak priamo nastaví na odosielanie dát na toto rozhranie, ktoré vykoná ďalšiu analýzu.

Kapitola 3

Qradar – systém SIEM

V oblasti kyberbezpečnosti a analyzovania dát existujú systémy, ktoré zbierajú a vyhodnocujú možné nebezpečné útoky a hrozby. Systémy bezpečnosti informácií a manažmentu udalostí (ang. Security Information Event Management – SIEM) sú produkty, ktoré napomáhajú bezpečnostným analytikom zhromažďovať, analyzovať a prezentovať informácie zo siete a z rôznych zariadení. Systém SIEM vznikol kombináciou systémov manažmentu bezpečnostných informácií (angl. Security Information System – SIM) a manažmentu bezpečnostných udalostí (angl. Security Event Management – SEM) [8].

Prvá skupina – systémy SIM poskytujú zbieranie, ukladanie a analýzu bezpečnostných záznamov získaných z rôznych zariadení v rámci siete. Z týchto údajov je možné napríklad identifikovať hrozby a sledovať aktivity v systéme.

Druhá skupina – systémy SEM narozenie od systémov SIM dokážu pracovať v reálnom čase. Zbierajú a analyzujú údaje z rôznych zdrojov, aby identifikovali potencionálne bezpečnostné hrozby a anomálie. Dokážu reagovať na incidenty a hlásenia na obmedzenie bezpečnostných rizík.

Spojením týchto dvoch systémov s rôznymi podielmi na funkciách vznikol systém SIEM, ktorý umožňuje analytikom monitorovanie v reálnom čase a zároveň dlhodobú analýzu bezpečnostných udalostí. Systémy SIEM zahŕňajú aj nástroje na vizualizáciu údajov, ktoré pomáhajú organizáciám identifikovať trendy a vzory v bezpečnostných udalostiach. Hlavné prvky a nástroje, ktoré by mal systém SIEM zahŕňať sú [8]:

- korelácia udalostí,
- detekcia nežiadúcich a nečakaných situácií pomocou pravidiel,
- mapovanie IP adres na adresy MAC,
- kontrola bezpečnosti chráneného systému,
- aplikačné rozhranie na integrovanie systémami tretích strán,
- centrálny pohľad na všetky udalosti systému.

Nie všetky vlastnosti sú dostupné v každom systéme SIEM v rovnakej miere. Okrem týchto vlastností ponúkajú systémy SIEM aj rôzne rozšírenia a nadstavby, ktoré umožňujú použitie umelej inteligencie a presnejšiu detekciu bezpečnostných rizik a udalostí. Systém SIEM je komplexný nástroj, ktorý je možné integrovať do chránenej siete akejkoľvek organizácie a napomôcť tak bezpečnostnému tímu lepšie chrániť systém pred nežiadúcimi hrozbami a útokmi [16].

V minulej kapitole boli opísané systémy IDS a ISP. Ich hlavný rozdiel oproti systémom SIEM je, že spravidla len detegujú hrozby a slúžia na prevenciu útokov, zatiaľ čo SIEM systémy údaje zhromažďujú a poskytujú širšiu analýzu.

3.1 IBM QRadar – Všeobecné informácie

Existuje veľmi veľa systémov SIEM dostupných na trhu, medzi ktorými je momentálne najpopulárnejší systém QRadar, vyvinutý spoločnosťou IBM. Systém QRadar, tak ako iné systémy SIEM, dokáže pracovať v kombinácii so získavaním dát v reálnom čase a s uloženými historickými dátami. Používa sa v mnohých malých i veľkých organizáciach na zbieranie, normalizáciu a korelaciu sietových dát. Dokáže zbierať rôzne udalosti z rôznych systémov ako napríklad: virtuálne privátne siete, systémy IDS a IPS, firewally a smerovače.

Základnými funkciami systému QRadar je správa zabezpečenia siete prostredníctvom monitorovania **tokov** a **udalostí**. Podstatný rozdiel medzi údajmi o udalostiach a tokoch spočíva v tom, že udalosť je záznamom konkrétnej akcie, ako je napríklad prihlásenie používateľa. Tok je záznam sietovej aktivity, ktorá môže trvať určitú dobu, v závislosti od aktivity v rámci relácie. Tok dát je záznam v sietovej relácii, ktorá trvá až niekoľko hodín a je medzi dvoma hostiteľmi [17].

V prípade tejto práce je systém QRadar využitý na spracovanie udalostí z prichádzajúceho programu Suricata. Ten odosiela len záznamy o udalostiach a preto zachytávanie tokov je pre tento projekt irrelevantné. Základná architektúra systému QRadar na zachytávanie dát sa skladá z troch vrstiev [26]: zbieranie, spracovanie a vyhľadávanie dát. Pre predstavu je na obrázku 3.1 zobrazený priebeh spracovania udalostí a tokov. V nasledujúcich kapitolách sa zameriame na to ako a akým spôsobom prebieha spracovanie udalostí.

3.2 Priebeh spracovania udalostí

Spracovanie udalostí sa vykonáva v troch fázach. Najprv za zozbierajú dátu z rôznych zdrojov a analyzujú pomocou určených modulov. Následne sa uložia a sú dostupné v konzolovej časti.

Zbieranie dát z udalostí a Modul podpory zariadení – DSM

Zbieranie dát je prvou vrstvou v architektúre systému, kde sa zbierajú dva základné typy dát: udalosti a toky. Na zbieranie dát systém QRadar využíva kolektory, ktoré ich najprv zozbierajú v surovej (angl. raw) forme. Potom sú tieto dátá zanalyzované a normalizované tak, aby mohli byť prezentované v štrukturalizovanom formáte pre užívateľa. Túto analýzu má na starosti komponent zberač udalostí.

Keď zberač udalostí príjme udalosti zo zdrojov, udalosti sa umiestnia do vstupných front na spracovanie. Z týchto front sa udalosti ďalej analyzujú pomocou modulu podpory zariadení – DSM [11]. Na základe pravidiel definovaných v module DSM sa pomocou analýzy získajú dátu z udalostí pre ďalšie spracovanie. Ide o informácie ako napríklad: IP adresy, miera nebezpečia, meno používateľa, porty a rôzne iné. Podľa týchto informácií sa vytvoria záznamy, ktorým už systém QRadar rozumie, sú teda zanalyzované a poskytnuté na ďalšie spracovanie.

Systém QRadar rozpozná známe zdroje udalostí podľa zdrojovej IP adresy alebo názvu hostiteľa, ktorý je obsiahnutý v hlavičke správy. Na základe tejto informácie sa vyberie

príslušný modul DSM [10]. Systém QRadar môže prijímať záznamy o hrozbách zo systémov a zariadení pomocou rôznych štandardizovaných protokolov ako syslog, NetFlow alebo SNMP. DSM sa konfiguruje podľa toho, aký protokol z akého zariadenia je prijatý. V prípade tejto práce komunikácia prebieha pomocou protokolu syslog. Navrhovaný modul DSM je určený pre zdroje správ, ktorými sú programy Suricata. Systém Qradar všetky nazbierané informácie ukladá do svojej vlastnej centralizovanej databázy. Jedná sa o databázu Ariel, vyvinutú pre účely systému Qradar. Každá vlastnosť, ktorú zanalyzuje modul DSM, je uložená do tabuľky udalostí – **events**.

3.2.1 Spracovanie udalostí

Po ukončení analýzy sa dátá odošlú do komponenty procesoru udalostí, ktorý ich spracováva podľa vlastných pravidiel, ktoré sú definované užívateľom. Vyhodnocovanie pravidiel prebieha za pomoci modulu **Custom Rules Engine** – CRE. Nástroj je zodpovedný za spracovanie udalostí prijatých systémom QRadar, ich porovnávanie s definovanými pravidlami a generovanie oznámení pre používateľov. Keď udalosti zodpovedajú pravidlu, z procesoru udalostí sa odošle informácia na magistrát systému QRadaru o tom, že konkrétna udalosť spustila pravidlo. Komponent magistrát v konzole QRadar vytvára a spravuje priestupky.

3.2.2 Vyhľadávanie dát z udalostí – magistrát

V najvyššej vrstve sú dátá dostupné používateľom a je im umožnené s nimi interagovať. Interakcia prebieha formou vyhľadávania, analýzy, vytváranie hlásení a investigácie detegovaných hrozien a priestupkov. Hrozby sú definované ako nebezpečné udalosti, zatiaľ čo priestupky sú generované na základe výskytu nebezpečných hrozien. Priestupok sa tak môže objaviť napríklad po získaní nebezpečnej udalosti z programu Suricata.

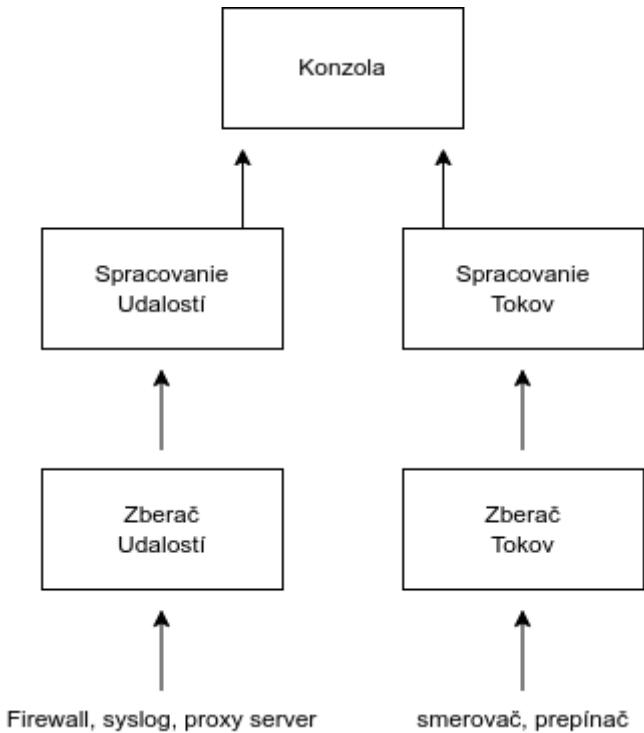
Spracovanie útokov

Jedným z hlavných prínosov systémov SIEM je detekcia útokov na základe hrozien, ktoré je možné identifikovať pomocou modulov DSM. Na správu priestupkov slúži komponent magistrát, ktorý dokáže zanalyzovať prichádzajúce priestupky na základe prijatých udalostí. Deteguje ich modul CRE, ktorý vytvára priestupky na základe definovaným pravidiel. V priestupkoch je možné sledovať rôzne informácie ako IP adresy, lokácie, informácie o tokoch a mieru nebezpečia. Pravidlá, ktoré dokážu detegovať, či sa jedná o priestupok alebo nie si môže užívateľ navoliť sám. Karta priestupky (angl. offenses) v systéme QRadar slúži na vypísanie zoznamu hrozien, ktoré sa momentálne nachádzajú v systéme [26].

3.3 Konfigurácia DSM

Na konfiguráciu prijímania správ z programov tretích strán, ktorým je práve program Suricata, je potrebné nastaviť dve hlavné súčasti a to zdroj záznamu a samotný modul DSM. Ak systém Qradar nerozozná zdroj automaticky, je potrebné ho nastaviť ručne voľbou sietového zariadenia a analyzačného modulu DSM. Ako zdroj môžu byť použité rôzne zariadenia a programy.

Na samotnú konfiguráciu a implementáciu modulu DSM, je možné využiť modálové okno. V ňom sa definujú pravidlá, podľa ktorých sa budú získavať atribúty zo záznamu. Väčšinou sa jedná o regulárne výrazy alebo vyberanie prvkov z textu vo formáte JSON. Podľa toho, aké pravidlá sa vytvoria sa budú zaznamenávať jednotlivé atribúty. Modálové



Obr. 3.1: Základná architektúra programu IBM QRadar

okno ponúka využitie/modifikáciu stávajúcich atribútov, ale ponúka aj možnosť vytvoriť úplne nové atribúty [5].

3.4 Mapovanie udalostí na kategórie nízkej úrovne

Dôležitou súčasťou systému QRadar je kategorizácia udalostí externých zariadení na vlastné kategórie. Výhoda tohto spôsobu je, že systém QRadar môže zbierať udalosti z ľubovoľného počtu zariadení a ich spoločne kategorizovať [14].

Dvoma hlavnými atribútmi, podľa ktorých QRadar rozoznáva a kategorizuje udalosti, sú kategória udalosti – **Event Category** a identifikátor udalosti – **Event ID**. Identifikátor udalosti predstavuje primárny identifikátor každej udalosti. Kategória udalosti, sekundárny identifikátor udalosti, vzniká na základe typu aktivity alebo správania, ktorú udalosť predstavuje. Tieto dva údaje jednoznačne identifikujú vzniknutú udalosť a sú potrebné k mapovaniu udalosti.

Ďalšou hodnotou potrebnou k mapovaniu je záznam QID (Qradar identifikátor). Záznam QID číselne reprezentuje konkrétnu udalosť externého zariadenia v rámci systému QRadar. Každý záznam QID obsahuje názov, popis, závažnosť a kategóriu nízkej úrovne.

Vytvorením záznamu QID sa získa údaj, ktorý je možné priradiť k udalosti analyzovanej modulom DSM. Tento proces sa nazýva mapovanie udalostí (angl. event mapping). Vezme sa kategória udalosti (angl. Event Category), špecifický identifikátor udalosti (angl. Event ID) a záznam QID. Na základe týchto troch údajov sa vytvorí mapovanie udalosti (angl. event mapping). Prakticky to znamená, že systém QRadar príjme udalosť, z ktorej modul DSM získa identifikátor udalosti (napr. MAC adresa) a kategóriu udalosti (napr. System

Restart). K týmto dvom údajom sa asociouje záznam QID s danou kategóriou nízkej úrovne (napr. System Boot).

Udalosti je možné rozdeliť do rôznych kategórii vyššej úrovne. Systém QRadar disponuje až 21 kategóriami vyššej úrovne. Jedná sa o závažné kategórie typu Malware, ale aj obyčajné kategórie, ktoré slúžia na charakteristiku bežných správ. Každá takáto kategória vyššej úrovne obsahuje niekoľko kategórii nižšej úrovne. Tie slúžia priamo na vytvorenie QID záznamu. Záznam QID predstavuje udalosť externého zdroja v "jazyku" systému QRadar.

3.5 Ukladanie informácií – Ariel Databáza a REST API

V systéme QRadar sa väčšina údajov ukladá do databázy Ariel. Z tohto zdroja je možné získať údaje prostredníctvom aplikačného rozhrania – REST API.

3.5.1 Ariel databáza

Ariel databáza je špeciálna databáza, ktorú používa systém QRadar na ukladanie a spracovanie bezpečnostných dát a informácií. Databáza bola vyvinutá spoločnosťou IBM pre ich vlastné účely. bola navrhnutá tak, aby dokázala efektívne zhromažďovať a ukladať veľké množstvo dát z rôznych zdrojov v sieti. Používa vlastný dopytovací jazyk podobný SQL, ktorý sa volá Ariel Query Language – AQL. Základnou databázou v systéme QRadar pre prácu s dátami je databáza Ariel.

Systém QRadar pracuje s dvoma tipmi dát: udalosti a toky, ktoré majú vlastné tabuľky v databáze Ariel. V tejto práci sa využíva tabuľka na udalosti, do ktorej sa ukladajú záznamy generované modulmi DSM. Jej štruktúra spočíva v tom, ako sú nakonfigurované moduly DSM. Ak sa pridá nová vlastnosť do modulu, vytvorí sa aj nový stĺpec v tabuľke.

3.5.2 REST API

Representational state transfer (REST) API je architektonický štýl na získavanie dát pomocou HTTP dotazov na koncové body aplikačného rozhrania. Pomocou tohto štýlu je možné získať, meniť, vytvárať a mazať zdroje v rámci aplikačného rozhrania alebo systému. Slúži napríklad na integrovanie systémov tretích strán do vlastných aplikácií na rozšírenie funkcionality. Systém QRadar pomocou tohto rozhrania ponúka možnosť, ako komunikovať priamo so systémom. K dispozícii na komunikáciu je široká škála koncových bodov, ktoré je možné získať alebo zmeniť hromadu informácií [2]. V tejto práci sa využijú hlavne koncové body na získavanie dát z databázy Ariel a informácie o priestupkoch. Na obrázku 3.2 je snímok obrazovky z dokumentácie, ktorú je možné nájsť v systéme QRadar. Dotazy je možné vykonávať priamo v dokumentácii.

3.5.3 Komunikácia s databázou Ariel a použité koncové body REST API

Na získanie dát z databázy Ariel je potrebné vykonať dotaz na REST API s dotazom v jazyku AQL. Dopytuje sa na tabuľku udalostí s výberom jednotlivých polí. Tak ako jazyk MySQL, aj jazyk AQL využíva agregačné funkcie a logické a porovnávacie operátory. V tomto jazyku však neexistujú žiadne relácie medzi tabuľkami. Nižšie je uvedený príklad dotazu v jazyku AQL [3]:

The screenshot shows the QRadar REST API Documentation interface. On the left, there's a sidebar with a tree view of API endpoints under 'API Version: 12.0'. The 'searches' endpoint under the 'ariel' category is selected. On the right, there's a detailed view for the 'GET /ariel/searches' endpoint. It includes a 'Description' section stating 'Retrieves the list of Ariel searches. This includes search_ids for completed and active searches.', a 'Response Description' section indicating 'A list of search IDs.', and a 'Success & Error Responses' section with a table mapping response codes to descriptions and status codes.

Obr. 3.2: Screenshot z dokumentácie QRadar REST API

```
SELECT sourceip, COUNT (*) as number
FROM events
WHERE logsourceid = '122' AND event_type='dns'
ORDER BY number DESC
LIMIT 5
```

Ďalej je možné získať a upravovať rôzne informácie, ktoré nesúvisia s databázou Ariel ale budú potrebné v tomto projekte. Na komunikáciu boli použité koncové body z tohto zoznamu:

- Ak chce užívateľ získať dátá z databázy Ariel musí najprv vytvoriť dotaz zavolením koncového bodu REST API. V hlavičke správy sa posielá zvolený dotaz v jazyku AQL.

POST -- /ariel/searches

- Následne sa cyklicky volá koncový bod na získanie informácií o tomto požiadavku na vyhľadávanie. Jedná sa o kontrolu, či daný dotaz je dokončený.

GET -- /ariel/searches/{search_id}

- Po úspešnom dokončení dotazu sa zavolá koncový bod na získanie výsledkov.

GET -- /ariel/searches/{search_id}/results

- Ďalej sa využije v práci koncový bod na získanie informácií ohľadom priestupkov, ktoré boli získané na základe pravidiel.

GET -- /siem/offenses

- Na to, aby mohli byť získané informácie z daného zdroja, je potrebné získať identifikátory zdrojov, ktoré sú špecifické pre moduly DSM.

```
GET -- /config/event_sources/log_source_management/log_sources
```

Kapitola 4

Návrh modulu DSM pre program Suricata

Hlavným cieľom tejto práce je navrhnúť modul DSM, ktorý dokáže spracovať zachytené záznamy z programu Suricata. Modul DSM analyzuje dátu, ktoré sú štandardizované protokolom syslog. Nižšie je uvedená syslog správa, ktorú produkuje program Suricata:

```
Dec 28 18:28:05 martin-HP-ZBook suricata[223314]: {"timestamp": "2023...
```

Jedná sa o začiatok syslog správy, kde za dátumom nasleduje meno zdroja, ktoré sa nastaví ako zdroj pre modul DSM a telo správy vo formáte JSON. Tým, že program Suricata produkuje správy v tomto formáte, konfigurácia modulu DSM je oveľa jednoduchšia a nie sú potrebné žiadne iné typy analýz, len manipulácia s JSON objektom. V následujúcej ukážke je príklad tela analyzovanej správy:

```
{
  "timestamp": "2022-12-28T18:28:05.865077+0100",
  "flow_id": 876600621377764,
  "in_iface": "wlo1",
  "event_type": "alert",
  "src_ip": "2600:9000:206e:f800:0018:30b3:e400:93a1",
  "src_port": 80,
  "dest_ip": "2a01:c844:205a:5400:153a:bbf3:9db1:e682",
  "dest_port": 60598,
  "proto": "TCP",
  "alert": {
    "action": "allowed",
    "gid": 1,
    "signature_id": 2100498,
    "rev": 7,
    "signature": "GPL ATTACK_RESPONSE id check returned root",
    "category": "Potentially Bad Traffic",
    "severity": 2,
    "metadata": {
      "created_at": [
        "2022-12-28T18:28:05.865077+0100"
      ]
    }
  }
}
```

```

        "2010_09_23"
    ],
    "updated_at": [
        "2010_09_23"
    ]
}
},
.
.
.

}

```

Pri analýze tela správy sa využíva vlastnosť JSON objektov. Na analyzovanie vlastností je potrebné vytvoriť v module DSM filtre. Jednou z možností, ako vytvoriť filtro, je využitie regulárnych výrazov. Druhou možnosťou je využitie spomínanej vlastnosti JSON objektov. Na implementáciu sa použije rozhranie v konzovlovej aplikácii systému QRadar [11].

4.1 Základné údaje

Najdôležitejšie zaznamenávané údaje sú tvorené z polí, ktoré sú spoločné pre všetky záznamy analyzované pomocou modulu DSM. Jedná sa o základné polia v databáze Ariel. Sú predpripravené systémom Qradar a dôležité pre základnú analýzu. Jedná sa o polia:

- zdrojová IP adresa – Source IP,
- cielová IP adresa – Destination IP,
- zdrojový port – Source Port,
- cielový port – Destination Port,
- protokol – Protocol,
- kategória udalosti – Event Category,
- identifikátor udalosti – Event ID.

Modul DSM je nastavený tak, že vlastnosti prichádzajúcej správy mapuje na základe základných vlastností modulu. Klasifikačným kritériom je meno vlastnosti indikujúce typ prenášanej informácie, napríklad "src_ip" je mapované na Source IP. Vlastnosti Event Category a Event ID sa priradzujú na typ udalosti – "event_type" a číslo signatúry – "signature_id". V ukážke záznamu sú zobrazené vlastnosti z prichádzajúcej správy:

```
{
    "src_ip": "192.168.1.2",
    "src_port": 80,
    "dest_ip": "192.154.100.3",
    "dest_port": 60598,
    "proto": "TCP",
    "event_type": "alert",
    "alert": {
        "signature_id": "230547",
        ...
    }
}
```

V takmer všetkých prípadoch mapovania sa priradí hodnota vlastnosti prichádzajúcej správy k prislúchajúcej vlastnosti modulu DSM. Ak sa jedná o Event Category alebo Event ID, tak sa nepriradzujú konkrétnie hodnoty vlastností. V prípade Event Category sa priradí názov buď Suricata Messages alebo Suricata Alert. Pre Event Category boli v tomto prípade využité dva regulárne výrazy, ktoré je možno vidieť na ľavom obrázku 4.1. V prvom prípade regulárny výraz na základe typu udalosti vyhodnotí, či sa jedná o hrozbu – "event_type": "alert" a priradí sa názov Suricata Alert. V druhom scenári sa priradí názov Suricata Messages, do ktorej patria udalosti okrem udalosti typu hrozba.

Pre Event ID boli vytvorené tiež dve skupiny. V tej prvej sa odchytia čísla pravidiel. V druhom prípade sa uloží názov konkrétnej udalosti, teda napr. dns alebo ssh, okrem udalosti alert. Oba prípady je možné vidieť na pravom obrázku 4.1. Do identifikátora udalosti – Event ID sa teda buď uloží číslo alebo názov udalosti – okrem hrozieb.

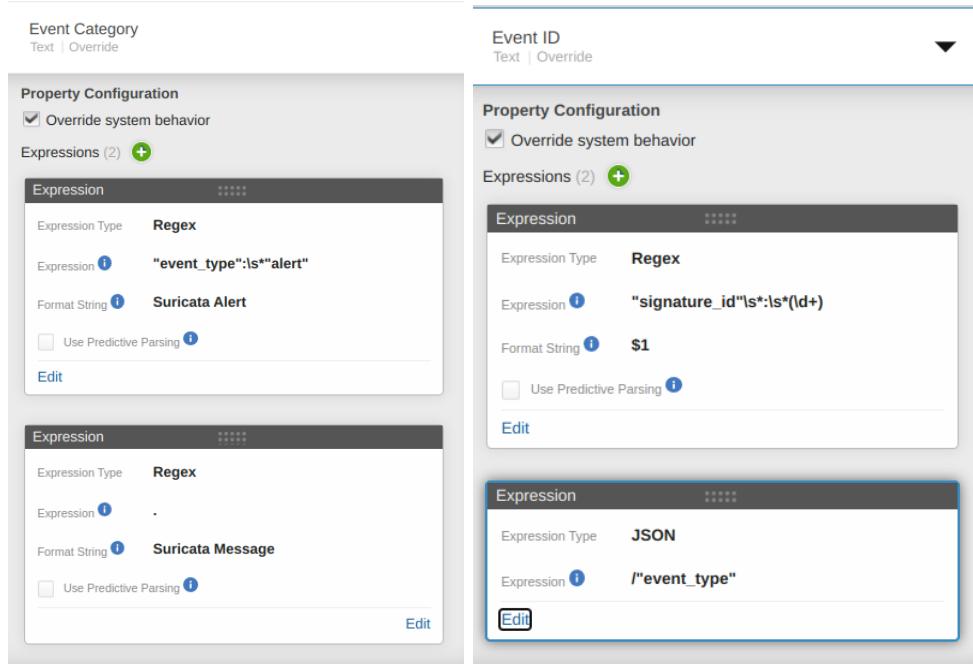
Event Category a Event ID spolu identifikujú konkrétnu udalosť, ktorú je možné na-mapovať ako je popísané v sekcii 3.4. Proces mapovania udalostí programu Suricata je opísaný v sekcii 4.3.

V tabuľke 4.1 sa nachádza priradovanie Event Category a Event ID pre všetky typ udalostí:

Typ udalosti	Event ID – identifikátor udalosti	Event Category - kategória
alert – hrozba	číslo pravidla	Suricata Alert
DNS, SSH, flow, .. ostatné typy	typ udalosti: flow, dns, ssh...	Suricata Message

Tabuľka 4.1: Prehľad vytvárania identifikátora udalosti a kategórie udalostí

Význam jednotlivých polí pre základné vlastnosti je v tabuľke 4.2, kde naľavo je údaj, ktorý chcem zaznamenať, v strede je názov pola v zázname pochádzajúceho z programu Suricata a napravo je názov záznamu, ktorý sa nachádza v DSM module a s ktorým pracuje systém QRadar.



Obr. 4.1: Snímka regulárnych výrazov kategórie a identifikátora udalostí

Údaj	Názov pola v zázname	QRadar základné údaje
zdrojová ip adresa	src_ip	Source IP
cieľová ip adresa	dst_ip	Destination IP
zdrojový port	src_port	Source Port
cieľový port	dst_port	Destination Port
typ protokolu	proto	protocol
typ správy	event_type	Event Category a Event ID

Tabuľka 4.2: Základné údaje a k nim určené názvy polí v JSON objekte

4.2 Vlastné údaje

Tvorba DSM modulov ponúka možnosť vytvorenia vlastných atribútov pre analýzu jednotlivých prichádzajúcich správ z programu Suricata. Základné nastavenie modulu DSM ponúka len niekoľko základných polí, ktoré nestacia na hlbšiu analýzu prichádzajúcich správ. Pre rôzne typy udalostí budú zaznamenané modulom DSM iné atribúty.

4.2.1 Alert

Hrozba – **Alert** je najdôležitejší záznam, ktorý je možné získať. Jedná sa o záznam, ktorý bol získaný na základe nejakého pravidla. Tieto pravidlá majú svoje jedinečné vlastnosti, ktoré používa program Suricata na analýzu hrozieb. Skúmajú sa hodnoty dôležité na určenie hrozby, jej pôvodu a miery nebezpečenstva. Každé pravidlo, ktoré bolo vytvorené pre program Suricata obsahuje číslo – **signature_id**, ktorým je jednoznačne identifikované. Meno pravidla je vo vlastnosti **signature** a pochádza z tretej časti signatúry programu – možnosti pravidla, viď obrázok 2.2 a popis k nemu. Vlastnosť **severity**, teda závažnosť hrozby, určuje mieru nebezpečenstva danej udalosti¹. Program Suricata rozlišuje tri stupne hrozby. Najzávažnejšie sú stupňa 1 - aktivity tróskeho koňa, stupeň 2 predstavuje možné útoky a stupňom 3 sú ohodnotené potencionálne hrozby. Na základe tohto čísla sa pri mapovaní udalostí priradí závažnosť hrozby. Nižšie je možné vidieť záznam hrozby s vyznačenými dôležitými poliami:

```
"alert": {  
    "action": "allowed",  
    "gid": 1,  
    "signature_id": 2024056,  
    "rev": 1,  
    "app_proto": "http",  
    "signature": "HTTP body talking about corruption",  
    "severity": 3,  
}
```

V tabuľke 4.3 sú zobrazené vlastnosti s popiskami.

Údaj	Názov pola
identifikátor pravidla	signature_id
meno pravidla	signature
miera hrozby	severity

Tabuľka 4.3: Údaje o hrozbách

¹Túto informáciu v oficiálnej dokumentácii nie je možné nájsť. V zdroji <https://docs.siemonster.com/current/optimizing-alerts> sú uvedené 3 mier hrozby, ale nejedná sa o oficiálnu stránku. Informácie boli nakoniec overené v konfiguračnom súbore – **classification**, v ktorom sú uvedené všetky kategórie pravidiel aj s číslom miery hrozby.

4.2.2 DNS

V prípade DNS záznamov² sú zaznamenávané údaje, ktoré sa nachádzajú v hlavičke paketu DNS [4]. Tento paket má dve verzie a to **request** a **response**, ktoré určujú o aký typ DNS správy sa jedná. Ďalej v hlavičke paketu nasledujú polia, ktoré sú dôležité len pre niektoré typy správ. Kedže nie každá správa ich môže obsahovať, nepoužijú grafickú analýzu, ale stanú sa súčasťou tabuľky. Pre grafickú analýzu sa použijú polia **rrtype** – typ záznamu zdroja, **rrname** – doménové meno zdroja a **rdata** – IP adresa, na ktorú sa názov domény prekladá. Nižšie je uvedený príklad záznamu správy DNS:

```
"dns":{  
    "version":2,  
    "type": "answer",  
    "id":50968,  
    "flags": "8583",  
    "qr": true,  
    "aa": true,  
    "rd": true,  
    "ra": true,  
    "z": false,  
    "tc": false,  
    "rrname": "2.100.168.192.in-addr.arpa",  
    "rrtype": "PTR",  
    "rcode": "NXDOMAIN",  
    "rdata": 192.168.100.5  
    "authorities": [  
        {  
            ...  
        }  
    ]  
}
```

V tabuľke 4.4 sú údaje zo záznamu s popisom.

Údaj	Názov pola
typ	type
dotaz/odpoved	qr
autoritatívna odpoved	aa
rekurzia	tc
rekurzia želaná	rd
rekurzia dostupná	ra
zdrojové meno záznamu	rrname
typ záznamu	rrtype
zdroj dát	rdata

Tabuľka 4.4: Údaje pre DNS

²<https://suricata.readthedocs.io/en/suricata-6.0.0/output/eve/eve-json-format.html#event-type-dns>

4.2.3 HTTP

V rámci protokolu HTTP³ je možné vypísať veľmi veľké množstvo dát, ale pre túto prácu je postačujúce poznať základné typy, ktoré jednoznačne určujú HTTP protokol [22]. Z tohto údaju používateľ môže zistiť odkiaľ, ako a akým spôsobom prebehol HTTP dotaz na určitý server. Parametrami sú meno servera, dotazovaná adresa URL, použitý prehliadač, metóda a status. Nižšie je zobrazený správy:

```
"http": {  
    "hostname": "test.co.uk",  
    "url": "\/test\/file.json",  
    "http_user_agent": "<User-Agent>",  
    "http_content_type": "application\/json",  
    "http_refer": "http:\/\www.test.com\/",  
    "http_method": "GET",  
    "protocol": "HTTP\1.1",  
    "status": 200,  
    "length": 310,  
    "request_headers": [  
        {...}]  
    ..  
}
```

Údaje sú uvedené v tabuľke 4.5 s popisom.

Údaj	Názov pola
meno hostiteľa	hostname
URL	url
typ prehliadača	http_user_agent
metóda	http_method
návratový kód	status
dĺžka	length

Tabuľka 4.5: Údaje na zaznamenanie HTTP

4.2.4 TLS

V rámci protokolu TLS⁴ je možné zobraziť informácie o poskytnutých certifikátoch. Z dát, ktoré je možné analyzovať, sa použije predmet – "subject" protokolu TLS. Pole subjektu je jednou z kľúčových zložiek certifikátu TLS a obsahuje identifikačné informácie o subjekte, ktorému je certifikát vydaný. Ďalším dôležitým poľom je "issuer", v ktorom je vydavateľ získaný z predošlého pola. Ďalej je analyzovaná hodnota "sni" označujúca názov servera, na ktorý sa pripája klient. Poslednou informáciou je verzia, ktorá bola použitá pri vytvorení spojenia TLS [21]. Nižšie je uvedený príklad záznamu TLS:

³<https://suricata.readthedocs.io/en/suricata-6.0.0/output/eve/eve-json-format.html#event-type-http>

⁴<https://suricata.readthedocs.io/en/suricata-6.0.0/output/eve/eve-json-format.html#event-type-tls>

```

"tls": {
    "subject": "C=US, ST=California, L=Mountain View, O=Google Inc,
    CN=*.google.com",
    "issuer": "C=US, O=Google Inc, CN=Google Internet Authority G2",
    "serial\char 34: "0C:00:99:B7:D7:54:C9:F6:77:26:31:7E:BA:EA:7C:1C",
    "fingerprint:"8f:51:12:06:a0:cc:4e:cd:e8...",
    "sni": "calendar.google.com",
    "version": "TLS 1.2",
    "notbefore": "2017-01-04T10:48:43",
    "notafter": "2017-03-29T10:18:00"
}

```

V tabuľke 4.6 sú údaje s popisom.

Údaj	Názov pola
subjekt – informácie o vydavateľovi	subject
vydavateľ	issuer
verzia	version
identifikátor servera	sni

Tabuľka 4.6: Údaje získané z TLS

4.2.5 SSH

V rámci protokolu SSH⁵ sa využili informácie, ktoré plynú z komunikácie medzi dvoma koncovými bodmi. Boli využité polia určujúce softvér a typ protokolu klienta a servera uskutočňujúcich pripojenie. Nižšie je uvedený typický záznam SSH.

```

"ssh": {
    "client": {
        "proto_version": "2.0",
        "software_version": "OpenSSH_6.7",
        "hash": {
            "hash": "ec7378c1a92f5a8dde7e8b7a1ddf33d1",
            "string": "curve25519-sha256,diffie-hellman-group..",
        }
    },
    "server": {
        "proto_version": "2.0",
        "software_version": "OpenSSH_6.7",
        "hash": {
            "hash": "ec7378c1a92f5a8dde7e8b7a1ddf33d1",
            "string": "curve25519-sha256,curve25519-sha256..",
        }
    }
}

```

⁵<https://suricata.readthedocs.io/en/suricata-6.0.0/output/eve/eve-json-format.html#event-type-ssh>

Údaje sú uvedené v tabuľke 4.7.

Údaj	Názov pola
protokol serveru	server proto_version
software serveru	server software_version
protokol klienta	client proto_version
software klienta	client software_version

Tabuľka 4.7: Údaje získané z SSH

4.2.6 DHCP

V rámci protokolu DHCP⁶, ktorý slúži na priradovanie adres DHCP, je možné zistiť informácie o serveri, na ktorý sa zasielajú žiadosti. V práci boli vybrané 3 informácie a to: typ správy, typ správy DHCP , a klientská IP adresa – `client_ip`, podľa ktorej je možné zistiť IP adresu klienta odosielajúceho žiadosť [9]. Nižšie je zobrazený výpis záznamu DHCP:

```
"dhcp": {
    "type": "reply",
    "id": 2787908432,
    "client_mac": "54:ee:75:51:e0:66",
    "assigned_ip": "192.168.1.120",
    "client_ip": "0.0.0.0",
    "relay_ip": "192.168.1.1",
    "next_server_ip": "0.0.0.0",
    "dhcp_type": "offer",
    "subnet_mask": "255.255.255.0",
    "routers": ["192.168.1.100"],
    "hostname": "test",
    ...
}
```

Údaje sú uvedené v tabuľke 4.8.

Údaj	Názov pola
typ	type
DHCP typ	dhcp_type
IP adresa klienta	client_ip

Tabuľka 4.8: Údaje získané z DHCP

4.2.7 FLOW

Udalosti Flow⁷ predstavujú toky dát v rámci nadviazaných spojení. Pre modul DSM sú zaujímavé dáta zaznamenávajúce počet paketov `pkts__` a počet prenesených bajtov

⁶<https://suricata.readthedocs.io/en/latest/output/eve/eve-json-format.html#event-type-dhcp>

⁷<https://suricata.readthedocs.io/en/suricata-6.0.0/output/eve/eve-json-format.html#event-type-flow>

`bytes_...` Spoločne so základnými údajmi je možné získať ucelený pohľad, koľko bajtov/paketov bolo prenesených z jedného miesta na druhé, kde sú najväčšie toky dát. Každý tok dát nesie so sebou informáciu či bola na ňom vyvolaná hrozba – "alerted". Poslednou informáciou pre modul DSM je dôvod ukončenia toku – `reason`. Nižšie je ukážka záznamu:

```
"flow": {
    "pkts_toserver": 23,
    "pkts_toclient": 21,
    "bytes_toserver": 4884,
    "bytes_toclient": 7392,
    "start": "2019-05-28T23:32:29.025256+0200",
    "end": "2019-05-28T23:35:28.071281+0200",
    "age": 179,
    "bypass": "capture",
    "state": "bypassed",
    "reason": "timeout",
    "alerted": false,
    ...
}
```

V tabuľke 4.9 sú uvedené záznamy s popisom.

Údaj	Názov pola
pakety odoslané serveru	pkts_toserver
pakety odoslané klientovi	pkts_toclient
bajty odoslané serveru	bytes_toserver
bajty odoslané klientovi	bytes_toclient
príznak hrozby	alerted
dôvod ukončenia toku	reason

Tabuľka 4.9: Údaje na zaznamenanie toku dát

4.3 Kategorizácia v rámci QRadaru

Dôležitou súčasťou integrácie nového systému do systému Qradar, je mapovanie udalostí generovaných týmto systémom na kategórie udalostí systému Qradar. Kategorizácia v rámci systému Qradar vzniká spojením kategórie udalosti a identifikátora udalosti so záznamom QID, ako už bolo spomenuté v sekcií 4.1.

Prvou navrhovanou kategóriou v rámci `Event_Category` je kategória `Suricata Messages`, do ktorej spadajú všetky správy, ktoré nie sú hrozbami. Identifikátor udalosti v tomto prípade je typ správy, teda: DNS, HTTP a iné. Podľa rôznych typov udalostí sa vytvoria záznamy QID, kde meno prislúcha k danej udalosti. Vzniknuté QID záznamy a mapovania udalostí sú zobrazené v tabuľke 4.10. V prvom stĺpci je možné vidieť typ udalosti, v druhom záznam QID a v treťom stĺpci je možné vidieť kategóriu nízkej úrovne Qradar. Kategória je priradená v rámci procesu mapovania udalostí.

Druhou kategóriou sú všetky hrozby vyvolané programom Suricata. Mapovanie udalostí pre hrozby (angl. Alert) sa určuje z:

Typ udalosti	Meno záznamu QID/názov udalosti	QRadar kategória nízkej úrovne
DNS	Suricata DNS	Unknown Generic Log Event
HTTP	Suricata HTTP	Unknown Generic Log Event
Flow	Suricata Flow	Unknown Generic Log Event
SSH	Suricata SSH	Unknown Generic Log Event
DHCP	Suricata DHCP	Unknown Generic Log Event

Tabuľka 4.10: Kategorizácia v rámci bežných správ programu Suricata

- identifikátora udalosti – **Event ID**, kde v prvej skupine regulárnych výrazov sa ukladá číslo,
- z prvej kategórie udalosti – **Event Category**, ktorá sa volá **Suricata Alert**,
- príslušného záznamu QID, ktorý vznikne na základe mapovanej udalosti.

Program Suricata používa približne 30 000 pravidiel s troma stupňami hrozieb v základom nastavení. Plán bol najprv namapovať ručne iba 300 pravidiel od vývojárov programu Suricata. Tie však takmer všetky patria do jednej kategórie a neupozorňujú užívateľa na žiadne reálne hrozby. Nakoniec bolo všetkých 30 000 pravidiel rozdelených do kategórii, ktoré ponúka systém QRadar a s ktorými pracuje.

Pravidlá sa v rámci programu Suricata rozdeľujú na balíky. Vývojári programu Suricata vytvorili základný balík pravidiel, ktorý zväčša informuje o neplatných hlavičkách v paketoch a nesprávnom použití paketov. Tieto pravidlá, ako som už spomíнал, nedokážu reálne ochrániť žiadnen počítač, preto použijem skript **suricata-update**, ktorý umožňuje aktualizovať pravidlá do zložky s pravidlami.

Najrozšírenejší balík, ktorý sa používa je od firmy **proofpoint**⁸, ten obsahuje približne 30 000 pravidiel, ktoré sú rozdelené do kategórií. Každá táto kategória je opísaná v dokumente na ich oficiálnej stránke⁹. Jedná sa o kategórie, ktoré je možné namapovať na kategórie nízkej úrovne systému QRadar. Balíček je možné stiahnuť pomocou skriptu na aktualizovanie pravidiel. Jeho názov je **et/open**.

Na mapovanie týchto kategórií je zvolený postup, v ktorom sa najprv prechádzajú všetky kategórie vysokej úrovne systému QRadar a všetky kategórie balíka **et/open**. Každé jedno pravidlo nesie so sebou v mene pravidla aj kategóriu balíka **et/open**, podľa ktorej je možné filtrovať. Ďalej sa využijú kategórie programu Suricata, z ktorých sú niektoré podobné kategóriám QRadar. Ak existuje špecifická kategória nízkej úrovne, ktorá sa nachádza v pravidlach, tak sa filtruje podľa klíčového slova, napr. **backdoor**. Ak pravidlo nie je možné filtrovať tak je umiestnené do kategórie **Unknown Suspicious Event**.

Podľa týchto troch vstupov sa postupne vytvárajú najprv záznamy QID, ktoré sa v kombinácii s **Event Category** a **Event ID** použijú na vytvorenie mapovania danej udalosti.

⁸<https://www.proofpoint.com/us/products/advanced-threat-protection/et-intelligence>

⁹<https://tools.emergingthreats.net/docs/ETPro%20Rule%20Categories.pdf>

Každému záznamu QID sa priradí meno podľa názvu pravidla. Toto meno reprezentuje názov udalosti v systéme QRadar.

Použitý regulárny výraz.

```
grep -iE 'msg:"[nazov kategórie].*?";|classtype:[názvo kategórie]?;'  
suricata.rules > rules-for-qradar/test!
```

V tabuľke 4.11 je uvedené ako sú namapované jednotlivé kategórie pravidiel na kategórie nízkej úrovne systému QRadar. Niektoré pravidlá sa nachádzajú viackrát v dvoch kategóriach. Jedná sa o kategórie, ktoré nie sú špecifické, napr. Unknown Suspicious Event. Preto sa najprv budú vkladať pravidlá, ktoré špecifické sú, napr. Recon a až potom všeobecné kategórie.

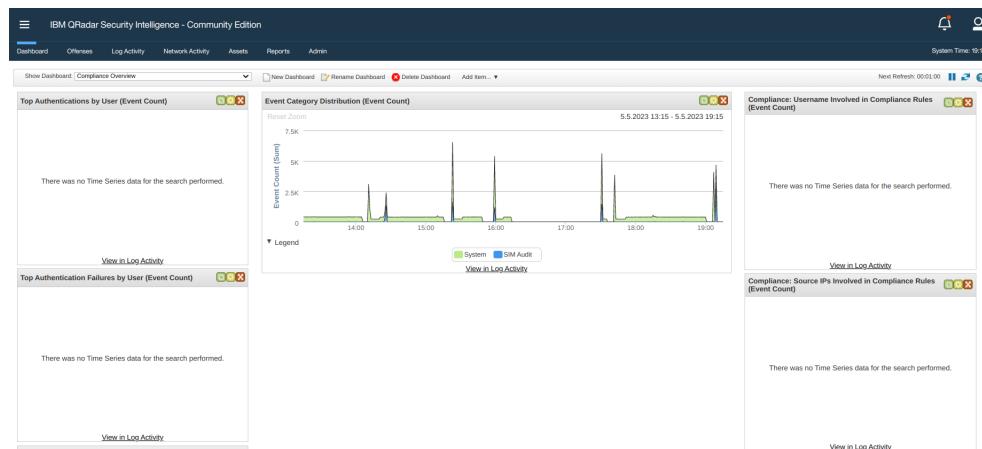
ET kategória	Suricata kategória	Kľúčové slovo	Poznámka	QRadar kategória nízkej úrovne
DoS	attempted-recon			Recon
	denial-of-service			Unknown DoS Attack
Attack-Response		SQL injection		SQL Injection
		záznamy o prevedených útokoch		Unknown Exploit Attack
Adware	pop-activity		adware	Adware
		backdoor		Backdoor
		keylogger		
	misc-attack			Misc Malware
Phishing				Phising
		Ransomware		Ransomware
				Spyware Detected
		spam		Spam
	trojan-activity			Trojan Detected
ET Malware				Unknown Malware
ET Policy				Unknown Policy
ET JA3 Hash, ET EXPLOIT				Unknown Potential Exploit
	DNS			DNS protocol anomaly
	misc-activity			Misc Suspicious Activity
	SQL			SQL anomaly
ET Web Specific Apps, ET WEB Client, ET WEB Server				Potential Web Vulnerability
		zvyšné pravidlá		Unknown Suspicious Event

Tabuľka 4.11: Tabuľka rozdelenia pravidiel do QRadar kategórií

Kapitola 5

Návrh addonu – vizualizácia dát z modulu DSM

Udalosti prúdiace do systému QRadar sa ukladajú do databázy Ariel. Tieto udalosti je možné zobraziť v konzolovej časti aplikácie. Na obrázku 5.1 je zobrazená ukážka základnej karty konzoly systému QRadar. Na tejto karte je možné nastavovať rôzne grafy zobrazujúce infomácie o udalostiach. Informácie sú ale obmedzené na všeobecné informácie spoločné pre všetky moduly DSM. Do kariet je možné pridať iba výsledky predpripravených dotazov vývojármu systému QRadar¹. Z toho vyplýva, že konkrétnie udalosti analyzované modulom DSM pre program Suricata nie je možné na tejto karte zobraziť. Cieľom tejto kapitoly je vytvoriť prehľadnú kartu na zobrazenie informácií z programu Suricata.



Obr. 5.1: Nástenka úvodnej stránky systému QRadar

Riešením je využiť možnosť vývoja a integrovania nových aplikácií do systému QRadar. Pre vývoj rozšírení je odporúčané použiť knižnicu Flask, napísanú v jazyku Python, ktorý je integrovaný v systéme QRadar. Vyvinutá aplikácia sa nainštaluje ako rozšírenie do systému QRadar a je jej poskytnuté vlastná karta v konzole systému QRadar [1].

¹<https://www.ibm.com/docs/en/qradar-on-cloud?topic=dashboards-log-activity>

5.1 Použité technológie – webová knihovňa Flask a nástroj Jinja2

Flask² je webový framework pre jazyk Python, ktorý umožňuje jednoducho vyvíjať webové aplikácie. Jedná sa o minimalistický framework so základnou funkcionalitou, ktorý je vhodný na rýchle používanie a je možné ho rozšíriť o množstvo funkcionalít. Navrhovaná aplikácia vo frameworku Flask slúži ako východiskový bod aplikácie a stará sa o zobrazenie získaných dát. K zobrazeniu pohľadov používa šablónovitý systém Jinja2. Umožňuje dynamicky vytvárať stránky. Navyše umožňuje dynamicky vkladať premenné, programové štruktúry, ktoré sú následne dynamicky spojené a vykreslené do HTML kódu. Na generovanie používa cykly, podmienky alebo aj makrás.

Aplikácia používa návrhový model – **Model - View - Template**, ktorý je typický pre frameworky využívajúce šablóny. Jedná sa o návrhový vzor, v ktorom sú využité tri dôležité komponenty: model, pohľad (angl. View) a šablóna (angl. Template). Model pomáha pri práci s databázou. Je to vrstva, ktorá pristupuje k údajom. V prípade tohto projektu sa jedná o už spomínanú databázu Ariel, ktorá je jediným zdrojom informácií a komunikácia prebieha cez REST API. Šablóny predstavujú statickú časť webovej aplikácie. V návrhu MVT neexistuje kontrôler, narozdiel od bežného návrhu MVC. Funkciu kontroléra obsluhuje pohľad. Ten na základe zvolenej cesty, vytvorí pohľad zo získaných dát a šablón.

Framework Flask zabezpečuje riadenie a spracovanie požiadaviek od klientov. Tieto požiadavky sú spracované pomocou definovaných funkcií (tzv. view funkcie), ktoré sú pripojené k určitým adresám URL. Pri spracovaní požiadavky dochádza k interakcii s databázou Ariel cez aplikačné rozhranie REST API od QRadaru. Funkcie na zobrazenie konečných pohľadov nebudú žiadnym spôsobom získavať dátu. Na to budú slúžiť samostatné cesty a každá jedna vracia iný typ informácií. Dotazovanie na koncové body REST API teda nemusí prebehnuť len jedenkrát pri načítaní šablóny, ale podľa toho, či to práve užívateľ vyžiada. Preto sa všetky dátá získajú asynchronným spôsobom. Tento spôsob ušetrí čas pri načítaní veľkého množstva dát.

Šablóny využívajú základné jazyky pre vývoj webových aplikácií, teda HTML a CSS. Navyše sú rozšírené o skripty v jazyku JavaScript, ktoré napĺňajú obsah stránky, pomocou dotazov na aplikáciu Flask. Na prevedenie dotazov sa používa rozhranie `fetch`³. Získanie dát prebieha asynchronne. Dotazy na databázu sa posielajú naraz, aby sa predišlo sekvenčnému načítaniu.

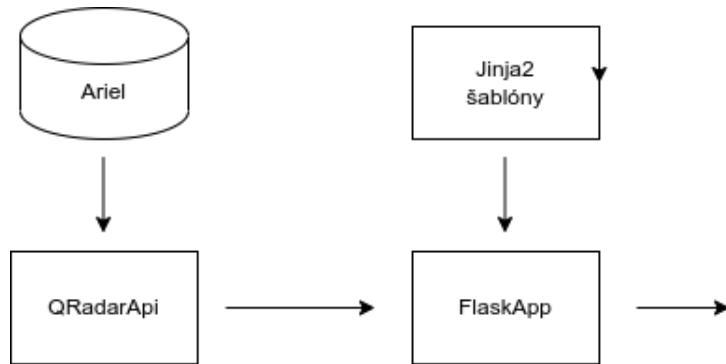
Na obrázku 5.2 je návrh ako vyzerá komunikácia vo výslednej aplikácii. Podľa návrhového vzoru MVT Flask aplikácia predstavuje pohľad, ktorý skladá výsledný vzhľad pomocou niekolkých šablón.

5.2 Vizualizácia dát

Najlepším riešením, ako zobraziť veľké množstvo údajov, sú dátové dashboardy. Jedná sa o vizuálne prehľady, ktoré zhromažďujú najdôležitejšie údaje potrebné na dosiahnutie konkrétnych cieľov na jednej obrazovke. Účinné dashboardy by mali byť vytvorené ako monitory nástroje, ktoré sú prehľadné na prvý pohľad. Dashboardy sú praktické nástroje, pretože dokážu stručne a jednoducho sprostredkovať obrovské objemy dát s využitím vizuálneho vnímania. Cieľom dashboardov je poskytovať podporu užívateľom a pomáhať im tak

²<https://flask.palletsprojects.com/en/1.1.x/>

³https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API/Using_Fetch



Obr. 5.2: Štruktúra navrhovaného add-on

napĺňať ich ciele [25]. Pri vytváraní užitočných dashboardov je nutné sa zamerať na dva aspekty a to výber správnych dát a výber správnej vizuálnej techniky [19].

V dnešnej dobe už existujú dashboardy, ktoré zobrazujú informácie z prostredia Suricata. Jedno z najpopulárnejších riešení je využitie komplexnej kolekcie nástrojov Elastic Stack⁴. V ňom ako frontendový prvok figuruje platforma Kibana, ktorá dokáže vizualizovať dátá zo záznamových súborov. Ďalším riešením je využitie nástroja Grafana⁵, ktorý funguje veľmi podobne. Návrh bol sčasti inšpirovaný týmito riešeniami.

V predchádzajúcej kapitole sa udalosti z programu Suricata rozdelili na dve kategórie: **Suricata Alerts** – hrozby a **Suricata Messages** – obyčajné správy. Tieto dve kategórie sledujú rozdielne informácie na sieti. Štruktúra aplikácie je rozdelená podľa týchto dvoch kategórií na dve karty. Na každej karte sa zobrazia prvky s informáciami, ktoré sú špecifické pre danú kategóriu. Na základe vlastností, ktoré získava modul DSM je možné sledovať prvky:

- grafy o zdrojových a cieľových portoch a IP adresách,
- grafy o použitých protokol, prenesených bajtoch a konkrétnych štatistikách pre udalost, (napr. typ webového prehliadača)
- údaje o počtoch hrozieb a udalostí,
- zobrazenie časovej osy prichádzajúcich hrozieb,
- tabuľka záznamov.

Spoločným prvkom oboch častí je vrchný panel, v ktorom si užívateľ zvolí časové rozmedzie hľadaných dát a zdroj, ktorý je analyzovaný navrhovaným modulom DSM. Na vytvorenie grafického návrhu bol využitý nástroj figme. V tomto nástroji boli vytvorené dva návrhy pre dve karty aplikácie. Nástroj figma⁶ slúži na návrh vizuálnej stránky aplikácií.

Posledným dôležitým prvkom pri návrhu dashboardu je dizajn. Ak užívateľ používa funkčnú aplikáciu ale so zlým vizuálom, odrádza ho to od dlhodobého používania danej aplikácie. Pri vytváraní návrhu som zvolil dve farby a to oranžová, ktorá je na logu programu Suricata a kontrastná modrá ako doplnok oranžovej.

⁴<https://www.digitalocean.com/community/tutorials/how-to-build-a-siem-with-suricata-and-elastic-stack-on-ubuntu-20-04>

⁵<https://grafana.com/grafana/dashboards/14893-ids-ips/>

⁶<https://www.figma.com/>

5.2.1 Návrh karty o hrozbách

Prvým krokom pri vytváraní dashboardu bola identifikácia dôležitých údajov pre užívateľa, čiže údajov, ktoré budú súčasťou karty hrozby. Prvá informácia, ktorá môže užívateľa zaujímať, je základná štatistika o hrozbách. Ďalej užívateľ potrebuje zobraziť vývoj udalostí v čase a tiež štatistiky o prúdiacich hrozbách zobrazené v grafoch. Pre užívateľa je tiež dôležitá možnosť voľby zdroju záznamov a času, v ktorom sa zobrazia údaje.

Navrhované prvky na karte

Na obrázku 5.3 je možné vidieť návrh základného rozloženia karty hrozby. Návrh neobsahuje rozloženie všetkých konkrétnych grafov, ale ukazuje približné rozdelenie prvkov.

Najdôležitejším údajom, ktorý je možné získať je údaj o mieri nebezpečenstva. Ten špecifikuje, ako nebezpečné je dané pravidlo. Podľa nebezpečenstva je možné pravidlá rozdeliť do troch kategórií. Ich počet sa zobrazí pod vrchným panelom aj s údajom o najčastejšie sa vyskytujúcej hrozbe. Hrozby s odlišnou mierou nebezpečenstva môžu prúdiť do systému kedykoľvek a je potrebné určiť, kedy presne sa hrozby vyskytovali. Na to slúži spojity lieneárny graf, ktorý zobrazí hrozby na časovej osi. Pod časovou osou nasledujú prstencové grafy o najčastejšie sa vyskytujúcich hrozbách v daných kategóriach.

Ďalším dôležitým údajom sú údaje o kategóriach a pravidlách. V prvých grafoch by sa mali užívateľovi zobraziť najčastejšie sa vyskytujúce pravidlá, kategórie nízkej úrovne, kategórie vysokej úrovne a na doplnenie informácií z prostredia Suricata aj kategórie využité v rámci programu Suricata.

Nebezpečné pakety prúdia vždy zo zdroja na určitý cieľ. V grafoch pod kategóriami za zobrazia údaje o výskytu najčastejších zdrojových a cielových IP adresách a portoch. Doplňujúcou informáciou sú informácie o najčastejšie sa vyskytujúcich transportných a aplikačných protokoloch. Predposledným prvkom je zobrazenie všetkých udalostí v tabuľke. Súhrnné informácie k pravidlu sa po kliknutí na riadok zobrazia v dialógovom okne.

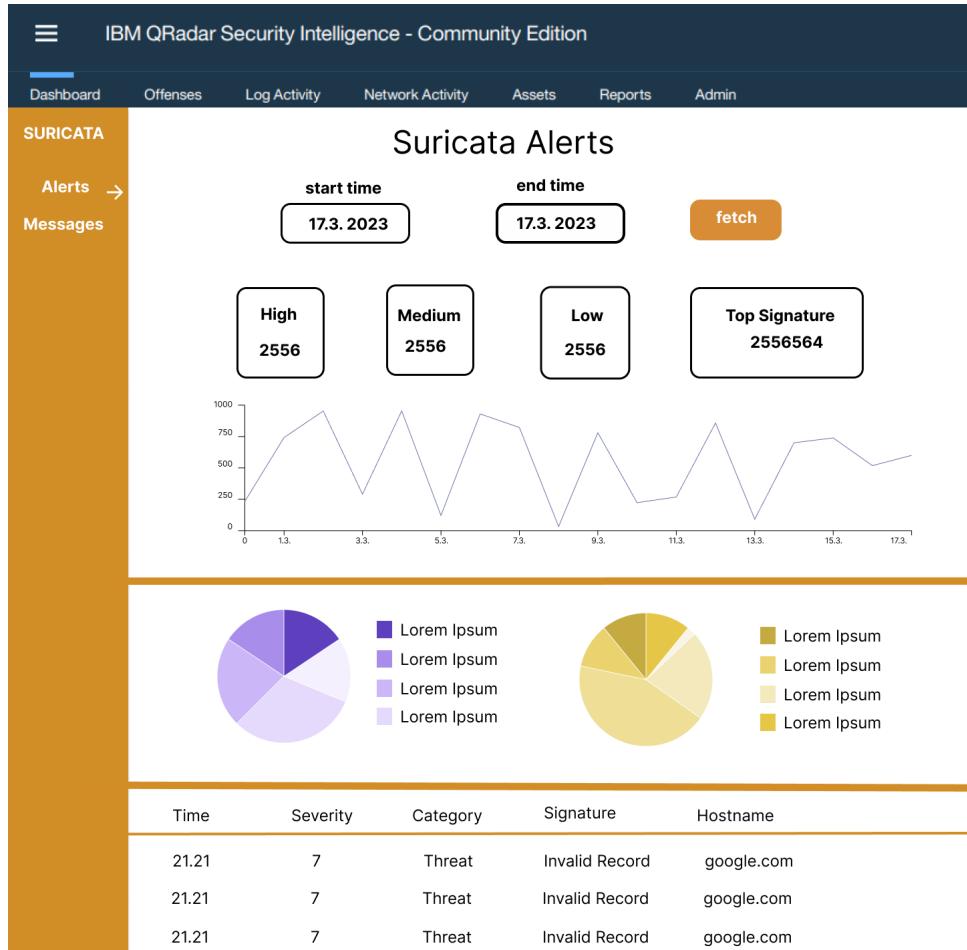
Systém QRadar na základe miery nebezpečenstva udalostí dokáže rozoznať potenciálne priestupky. Posledným prvkom na karte je časť zobrazujúca informácie o týchto priestupkoch. V rámci priestupkov je možné získať počet a agregovať výsledky podľa zdrojovej IP adresy. Všetky priestupky sa zobrazia v tabuľke so základnými údajmi. Bližšie špecifikácie si užívateľ môže zobraziť v karte priestupky – **offenses** systému Qradar.

5.2.2 Návrh karty o správach

V karte správy sa zobrazia všetky udalosti, ktoré nie sú hrozbami. Správy generované programom Suricata informujú užívateľa o tom, čo sa deje na sietových rozhraniach. Účelom karty nie je zobraziť všetky informácie o vybraných typoch udalostí, ale informovať užívateľa, čo sa v sieti deje. Na obrázku 5.4 je návrh tejto karty.

Spoločný panel slúži na výber času a zdroja udalostí. V prípade karty správy je doplnený o všeobecné informácie pre všetky typy udalostí. Vo vrchnom paneli sa zobrazia informácie o počte udalostí a graf najčastejšie sa vyskytujúcich transportných protokolov a typov udalostí.

Na karte správy sa udalosti rozdeľujú na sekcie podľa typu udalostí: DNS, HTTP a ďalšie spomenuté v návrhovej časti. Sekcie sú preklikateľné a obsahujú špecifické štatistiky, na základe ktorých je možné získať informácie o prevádzke na sieti. Pre jednotlivé typy udalostí sa vo výsledku zobrazia grafy s týmito informáciami:

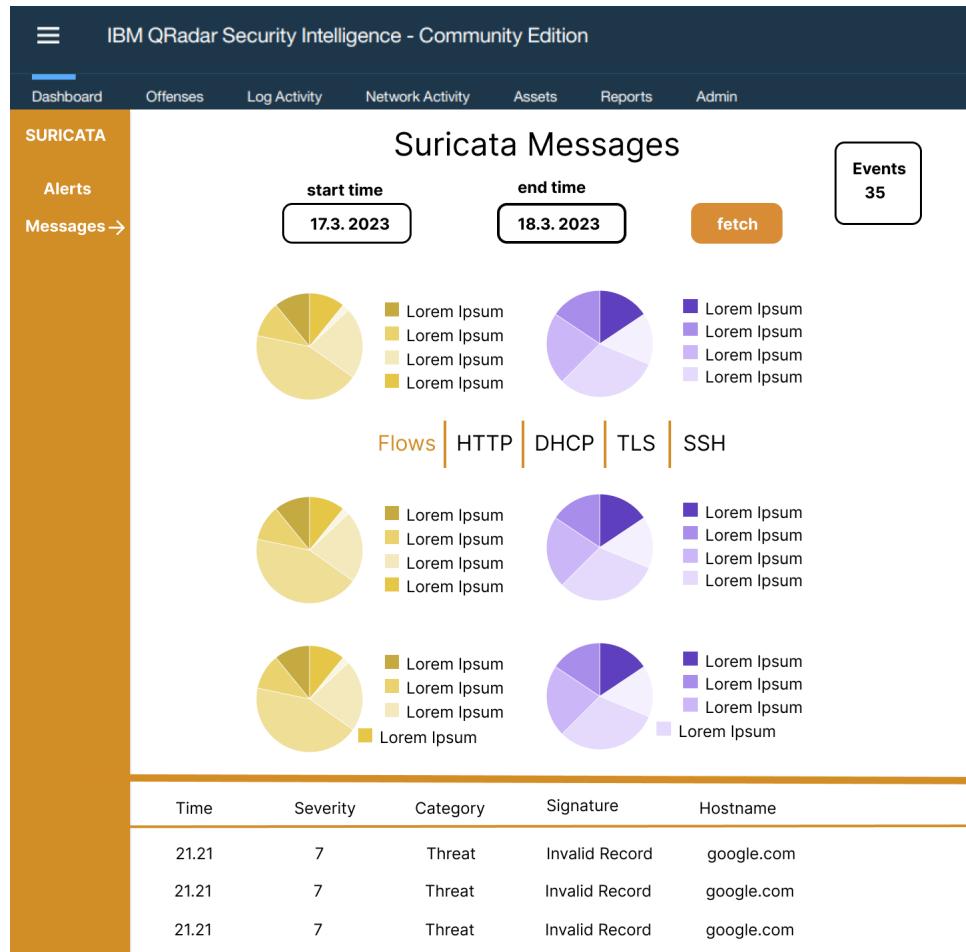


Obr. 5.3: Návrh karty pre hrozby vo figme

- **Flows** – výpis dát zahŕňa najčastejší výskyt prenesených bajtov na cieľové a zdrojové IP adresy a porty, počtu prenesených bajtov využitím aplikačných protokolov a počtu nebezpečných tokov.
- **DNS** – výpis dát zahŕňa najčastejší výskyt typov DNS správ, typov správ a DNS serveroch.
- **HTTP** – výpis dát zahŕňa najčastejší výskyt mena serveru, adries URL, transportných protokolov, statusov HTTP a HTTP metód.
- **TLS** – výpis dát zahŕňa najčastejší výskyt vydávateľov certifikátov, použitých serverov a cieľových IP adres.
- **SSH** – výpis dát zahŕňa najčastejší výskyt softvérov a protokol klienta, softvérov a protokolov servera a cieľových IP adres servera.
- **DHCP** – výpis dát zahŕňa najčastejší výskyt typu DHCP odpovede, typu dotazu, IP adries klienta.

Pod grafmi sa tak, ako v prípade karty správy, zobrazí tabuľka s udalostami pre daný typ. Vo výpise vyššie nie sú zahrnuté všetky vlastnosti získané modulom DSM. Vlastnosti,

které sa nezobrazia v grafoch sú pridané do tabuľky, v nej sa zobrazia všetky vlastnosti pre daný typ udalosti.



Obr. 5.4: Návrh karty pre správy vo figme

Kapitola 6

Implementácia DSM modulu a aplikácie pre systém Qradar

Hlavným cieľom tejto práce je naprogramovať a nastaviť modul DSM a aplikáciu – addon pre systém QRadar. Je dôležité uviesť, že sa používala jediná dostupná verzia systému QRadar – komunitné vydanie vo verzii 7.3. Celá implementácia a nasadenie sa prispôsobilo tejto verzii. Jedná sa o staršiu verziu, ktorá priniesla so sebou množstvo problémov stažujúcich vývoj. Komunitná verzia by mala byť nasadená na veľkých serveroch a domáci počítač nestíha spracovať všetky požiadavky, ktoré tento systém vyžaduje.

Pri implementácii bol zvolený zvolil nasledujúci postup. Najprv sa pripravil systém Qradar vo virtualizovanom prostredí. Následne sa vytvoril modul DSM a aplikácia podľa návrhu. Po ukončení implementácie nasledovalo testovanie na reálnych prípadoch použitia.

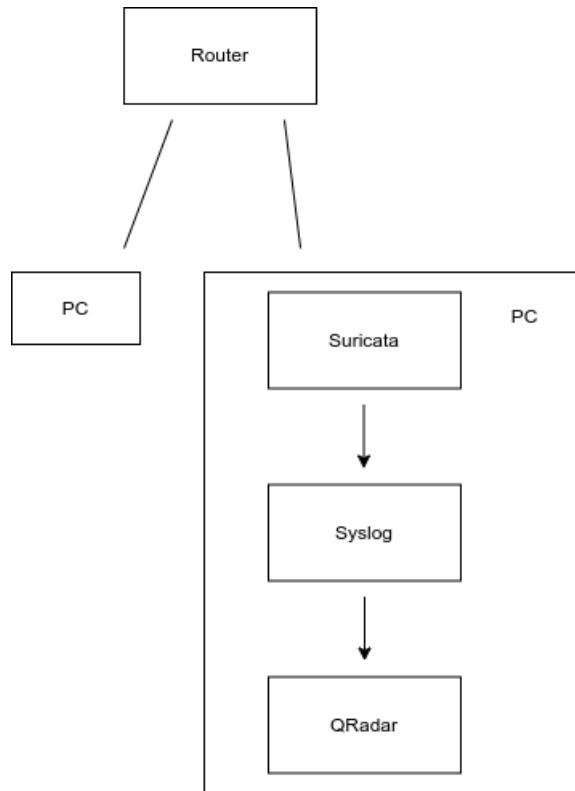
6.1 Virtualizácia systému Qradar a nastavenie programu Suricata

Systém je nasadený vo virtualizovanom prostredí **VirtualBox**, z ktorého je presmerovaný port na komunikáciu s lokálnym hostiteľom. Toto riešenie sa ukázalo ako dostatočné pre vývoj modulu DSM. Zároveň sa toto riešenie ukázalo ako nevhodné pre potreby veľkých dotazov a manipulácie s dátami. Systém vydrží bežať približne 30 minút a je potrebné ho naštartovať znova, čo trvá približne 5 minút.

Program Suricata je nastavený v základnej konfigurácii ako HIDS prvok chrániaci hostiteľský počítač. Na hostiteľskom počítači je nakonfigurovaný program rsyslog (pozn. rozšírený syslog) na odosielanie dát do systému QRadar – virtualizovaného v rámci hostiteľského počítača a virtuálnej sieti. Vizualizácia toho ako sú zapojené jednotlivé prvky je zobrazená na obrázku [6.1](#).

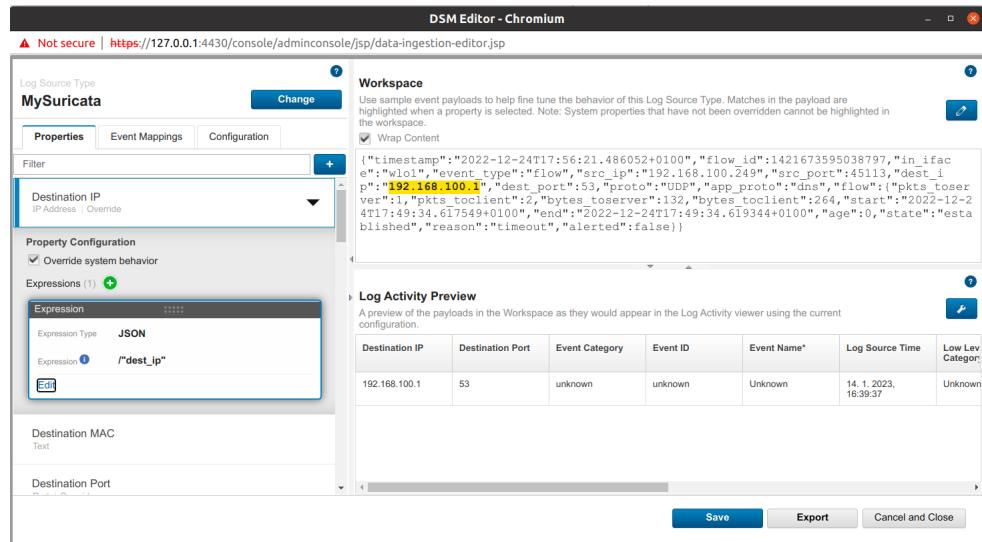
6.2 Implementácia modulu DSM

Pri implementácii modulu DSM sa najprv implementovala časť pre analýzu jednotlivých dôležitých vlastností. Použili sa všetky navrhované polia, ktoré sa určili pri návrhu v kapitole [4](#). Ako už bolo spomenuté, v práci sa pri implementácii použilo získavanie informácií pomocou vlastnosti objektu JSON. V dvoch prípadoch sa použité regulárne výrazy – **Event Category** a **Event ID**. Implementácia sa vykonala v dialógovom okne, ktoré je



Obr. 6.1: Vizualizácia jednotlivých prvkov v sieti

zobrazené na obrázku 6.2. Do neho je možné vložiť príkladnú správu a v ľavej časti okna vytvárať nové vlastnosti.



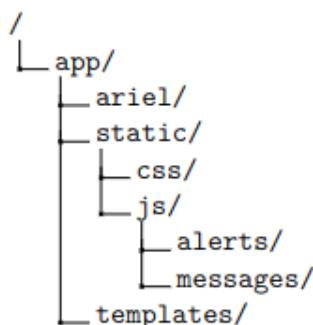
Obr. 6.2: Modálové okno pre konfiguráciu DSM

Vytvorenie mapovania QID

Podľa návrhu sa postupuje aj pri namapovaní udalostí do systému QRadar. Na presun všetkých pravidiel rozdelených do kategórii sa využívajú koncové body REST API, ktoré umožňujú vzdialene vytvoriť záznamy QID a mapovanie medzi záznamom QID a kategóriou. Zdrojový kód je možné nájsť v [Github repozitári](#)¹. V priečinku `rules-for-qradar` sú rozdelené pravidla podľa tabuľky [4.11](#). Mapovanie prebiehalo niekoľko dní vzhľadom na množstvo pravidiel a obmedzenosť komunitnej verzie.

6.3 Implementácia aplikácie

Podľa návrhu v kapitole [5](#) je implementovaná aplikácia určená pre systém QRadar. Aplikácia je dostupná v repozitári na webovej stránke [Github](#) v dvoch verziách: na nasadenie v systéme QRadar a na spustenie aplikácie mimo systém. Aplikácia je implementovaná v jazyku Python 2 vo frameworku Flask s využitím šablón v jazyku HTML doplnených o štýly v jazyku CSS. Šablóny sa dynamicky menia pomocou skriptov v jazyku JavaScript. Súborová štruktúra je zobrazená na obrázku [6.3](#).



Obr. 6.3: Súborová štruktúra aplikácie

6.3.1 Serverová časť - Flask

Serverová časť aplikácie sa skladá z dvoch častí. V aplikácii sa vytvorili dve cesty na vytváranie dvoch základných pohľadov. V prvom prípade sa vykresluje karta hrozby – **Alerts** a v druhom prípade karta správy – **Messages**. Tieto dva pohľady sú vykreslované dvoma šablónami.

Vývojári systému QRadar majú v správe vývoj knižnice `qpylib` v jazyku Python, ktorá je určená pre vývoj rozšírení. Knižnica však neobsahuje žiadnu detailnejšiu dokumentáciu. Z repozitáru na [Githubu](#)² nie je jasné, ktorá verzia je kompatibilná s komunitnou verzou systému QRadar. Z tohto dôvodu neboli použité funkcie z tejto knižnice, s výnimkou jednej, ktorá obsluhuje dotazy na REST API.

Na získavanie dát z aplikačného rozhrania QRadar REST API, je v aplikácii vytvorená trieda API v súbore `rest_api.py`. Trieda má dve metódy pre protokoly HTTP, a to GET a POST. Z triedy API je odvodená trieda Search, ktorá zabezpečuje vytváranie dotazov na databázu Ariel.

¹<https://github.com/tink0mar/move-sids-to-qradar>

²<https://github.com/IBM/qpylib/tree/master/qpylib>

Dedená trieda `Search` sa skladá z jednej funkcie, ktorá vykonáva celý proces získavania dát z databázy. Využíva k tomu tri funkcie. Prvá vytvorí dotaz do databázy, druhá funkcia kontrolouje, či sa dotaz už vykonal a tretia vráti požadovaný výsledok.

Priamo na vytváranie dotazov s rozhraním sa využívajú dva spôsoby, podľa ktorých je aplikácia rozdelená na dve verzie. Jedna využíva knižnicu `qpylib`, konkrétnu funkciu REST, zabezpečujúcu komunikáciu medzi webovým serverom a systémom QRadar. Využitie knižnice umožňuje bezpečné vytváranie dotazov do systému QRadar. Funkcia nepotrebuje žiadnu konfiguráciu a je bezpečná na používanie v systéme QRadar. Druhý spôsob využíva knižnicu `requests`. Dotazy je ale potrebné nakonfigurovať environmentálnymi premennými.

Druhá časť aplikácie Flask slúži na získavanie dát z systému QRadar a databázy Ariel. Využíva k tomu triedu `Search`. Všetky volania sa nachádzajú v priečinku `ariel`. V priečinku sú súbory rozdelené podľa typov udalostí a v každom súbore je vytvorený objekt triedy `Blueprint`³, ktorý zoskupuje pohľadové funkcie.

Pohľadové funkcie – koncové body sa riadia konvenciou REST API. Dotazovanie prebieha pomocou URL adres a v každom dotaze je možné dodať aj dotazovacie parametre. Spoločný parameter pre všetky koncové cesty je typ zdroja a časový úsek, z ktorého sa majú získať dátá. Ďalej pri dotazoch na získanie celkových dát je možné špecifikovať aj triedenie a rozsah. Dáta sa podľa konvencie vracajú vo formáte JSON [20]. Výsledky sú použité na zobrazenie grafov v užívateľskom rozhraní. Pohľadové funkcie sú vypísané v nasledujúcom zozname:

- `/default` – záznamové zdroje, kategórie, počet záznamov, protokoly a dát do tabuľiek, kde sú dotazy rozdelené podľa toho a aké zdroje sa žiada.
- `/alert` – signatúry, použité protokoly, cieľové a zdrojové IP adresy,
- `/http` – meno hostiteľa, metódy, statusy, typy prehliadačov a transportné protokoly
- `/dns` – zdrojové meno záznamu, typ záznamu a zdroj dát
- `/ssh` – softvér a verziu klienta/servera
- `/dhcp` – typ DHCP správy, IP adresa klienta
- `/tls` – informácie o vydávateľovi, identifikátor servera, verzia
- `/flow` – zdrojový/cieľový port a IP adresu v bajtoch, počet prenesených bajtov podľa paketu, počet nahlásených tokov
- `offenses` – priestupky

6.3.2 Užívateľské rozhranie

Základom užívateľského rozhrania sú 4 šablóny, ktoré vytvárajú všetky pohľady. Podľa návrhu boli vytvorené dva hlavné pohľady, na ktorých sú zobrazené všetky informácie. Základnou šablónovou je `base.html`, v ktorej sa načítavajú všetky dôležité balíky a knižnice. Šablóna slúži ako základ pre dve šablóny, ktoré vytvárajú pohľady na dve karty tejto aplikácie. V šablóne `base.html` sú importované šablóny `calendar.html` a `modal.html`. Šablóna `calendar.html` predstavuje výber dátumov vo vrchnom panele a šablóna `modal.html` predstavuje všetky použité dialógové okná. Na vytvorenie dvoch hlavných kariet sa ako

³<https://flask.palletsprojects.com/en/2.2.x/tutorial/views/>

základ použila šablóna `base.html`, nad ktorou boli postavené šablóny pre tieto karty, a to `alert.html` a `message.html`.

Na základnú prácu s prvkami, bol použitý framework `Bootstrap`⁴. Ten umožňuje jednoduché využitie 'mriežkového' systému na vytvorenie rozloženia a základný vzhľad webu. V projekte sa využili dve farby, ktorých odtiene vytvorili celkový vzhľad. Vzhľad stránky je definovaný v CSS súboroch a upravuje zväčša farbu a rozloženie.

Na získavanie dát a dynamickú zmenu boli použité skripty v jazyku JavaScript. Tie obsahujú volania na koncové body serverovej časti a následnú zmenu grafov a tabuliek. V podstate takto fungujú všetky skripty, ktoré po zavolení prekreslia určitý objekt – graf alebo tabuľku. Na vykonanie volaní sa využíva rozhranie `fetch`. Každý pohľad ma vlastný priečinok so skriptami. Navyše skripty pre správy sú rozdelené podľa typu udalostí. Každý skript obsahuje volania funkcií hlavného skriptu `utils.js`, kde sa nachádzajú obslúžne funkcie na prácu s časom a vytváranie grafov.

Celkový Vzhľad

V implementácii oboch kariet sú využité prvky navrhnuté v kapitole 5. Karta hrozby je rozdelená na dve časti na vrchu sa zobrazujú údaje k hrozbám. Najprv sú to všeobecné informácie o hrozbách, nasleduje časová os a grafy. Posledným prvkom sekcie hrozby sú tabuľky. Na konci stránky sú údaje o priestupkoch, ktoré sú zachytené systémom QRadar. Pre predstavu karty hrozby je možné vidieť snímky aplikácie v prílohe C.1.

Karta správy je rozdelená na vrchný panel s celkovými výsledkami a dynamickú sa meniacu sekciu pre každý typ udalosti v kategórii `Suricata Messages`. Sekcia sa dynamicky mení podľa zvoleného typu správ. Nasledujú grafy so štatistikami pre daný typ a na konci stránky je vždy tabuľka s udalosťami. Snímky aplikácie je pre predstavu možné vidieť v prílohe C.2.

Navigácia

Naľavo v aplikácii sa nachádza navigácia, v ktorej sa nachádzajú dva odkazy na karty správy a hrozby.



Obr. 6.4: Navigácia

Vrchný panel

Všetky pohľady v aplikácii majú spoločný vrchný panel. Na vrchnom paneli je zobrazený stav databázy Ariel na informovanie užívateľa. V paneli je možné zvolať rozsah dátumov,

⁴<https://getbootstrap.com/docs/4.0/getting-started/introduction/>

v ktorom sa vyhľadajú udalosti. Po rozkliknutí na dátum sa zobrazí kalendár. Ten bol vytvorený pomocou knižnice `flatpickr`⁵.

Po výbere dátumov tlačidlo **Fetch data** odošle sériu dotazov na koncové body webového servera Flask. Tie, ako už bolo spomenuté, získajú dátu z Ariel databázy a odošlú dátu vo formáte JSON na klientskú stránku. Skripty v jazyku JavaScript prekreslia všetky grafy a tabuľky na podstránke. Prekreslením len jednotlivých prvkov, sa nemusí načítať celá stránka a ušetrí sa čas potrebný na získanie dát.

Na kolko sa jedná o aplikáciu, ktorej ovládanie nemusí byť jasné každému užívateľovi, bolo pridané aj tlačidlo na zobrazenie pomoci. V sieti, ktorú monitoruje systém QRadar je možné, že bude napojených viacero naslúchajúcich programov Suricata. Z tohto dôvodu si užívateľ môže špecifikovať zdroj dát. Na obrázku 6.5 je zobrazený vrchný panel.



Obr. 6.5: Vrchný panel

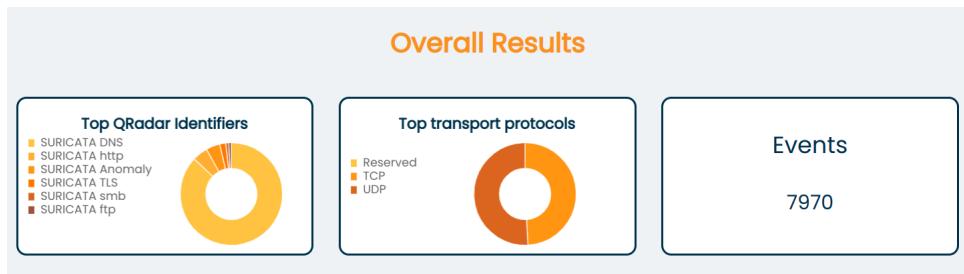
Podľa návrhu pre kartu hrozby bol k vrchnému panelu pridaný ešte jeden panel, v ktorom je zobrazený počet hrozien rozdelených podľa miery nebezpečia a najčastejšie sa vyskytujúca hroznba. K návrhu aplikácie bola pridaná možnosť filtrovať udalosti podľa miery nebezpečia a pravidla. Táto možnosť umožňuje filtrovať hrozby, ktoré sa zobrazia v grafoch podľa miery nebezpečia pravidla. Užívateľovi umožňuje napríklad zobraziť údaje len pre jeden typ pravidla. Na zvyšných grafoch, tabuľke a časovej osi sa zobrazia udalosti len udalosti zvolenej miery nebezpečia. Informatívny panel o hrozibách a ich miere nebezpečenstva je vidieť na obrázku 6.6.



Obr. 6.6: Informatívny panel pre kartu hrozby vo vrchnej časti stránky

Vrchný panel karty správy obsahuje navyše informácie o všetkých udalostach typu **Suricata Messages**, vidieť obrázok 6.7. Vykreslujú sa dva grafy: najčastejší výskyt typu udalosti a najčastejšie použitý transportný protokol. V poslednej kartičke je počet udalostí v danom časovom rozmedzí. Všetky informácie sú filtrované bez typu udalosti hrozba.

⁵<https://flatpickr.js.org/>



Obr. 6.7: Informatívny panel pre kartu hrozby vo vrchnej časti stránky

Tabuľky

Na vytvorenie tabuľiek je použitá knižnica **Tabulator**, naprogramovaná v jazyku JavaScript. Hlavným prínosom je široká škála funkcií a možností, ktoré umožňujú jednoduchý vývoj tabuľiek. V rozsiahlej dokumentácii⁶ je popísané, ako využiť funkcie na triedenie, filtrovanie, stránkovanie a manipuláciu s údajmi. Navyše dokáže komunikovať so serverom pomocou AJAX štandardu a získavať dátá asynchronicky. Pre každý typ udalosti je vytvorená tabuľka, ktorá vyzerá tak, ako je zobrazené na obrázku 6.8. V hlavičke sa nachádzajú mená stĺpcov, ktoré sú abecedne zoradené. V tele tabuľky sú zobrazené získané dátá pre danú udalosť. Každá udalosť má časť stĺpcov individuálnych, hlavička sa teda tvorí automaticky.

Dátá sa získavajú pomocou dotazov na serverovú časť. V dotaze sa špecifikuje typ udalosti, podľa čoho sa špecifikujú dátá pre danú udalosť. Pri dotaze sa ďalej špecifikuje dátum a rozsah údajov, tzv. stránkovanie. Vďaka stránkovaniu sa načíta len určitý počet riadkov a webový klient si nemusí ukladať všetky dátá z daného časového horizontu. Ovládanie stránkovania je umožnené prostredníctvom tlačidiel na spodnej lište. Každý stĺpec obsahuje tlačidlo vo forme šípky, ktoré navyše posielá v dotaze údaj o triedení daného stĺpca.

Knižnica umožňuje ľubovoľne posúvať stĺpce a riadky. Navyše bola implementovaná aj možnosť si zobraziť konkrétny riadok v dialógovom okne, ktoré je na obrázku 6.9. Toto okno zobrazí všetky údaje pod sebou a umožňuje zobraziť dátá prehľadnejšie.

Grafy

Chart.js je popredná knižnica na vytváranie grafov vo webových aplikáciach⁷. Umožňuje použiť až 8 rôznych typov s rôznymi nastaveniami. Rovnako ako knižnica na tabuľky aj táto knižnica je jednoduchá na ovládanie a poskytuje množstvo možností, ako prispôsobiť správanie a zjednodušiť prácu užívateľovi. V porovnaní s komplexnejšou knižnicou D3 poskytuje jednoduchšiu implementáciu grafov. Knižnica D3 poskytuje širokú funkcionalitu, ale využitá bola len interpolácia farieb pre prstencové grafy⁸. Farby do grafov sa interpolujú medzi dvoma farbami na základe rozsahu prvkov.

Každý prstencový graf sa vyrába rovnakým spôsobom a má tie isté funkcie. V prstencových grafoch je možné vyškrtnúť údaj, ako je zobrazené na ľavom obrázku 6.11. Na pravom obrázku je zobrazený graf, z ktorého je možné zistíť najčastejšie sa vyskytujúcu vlastnosť v zvolených udalostiach. Na lepšie pochopenie toho, aký je počet údajov v určitej časti grafu, sa užívateľovi zobrazí nápoveda s počtom konkrétnej položky v grafe.

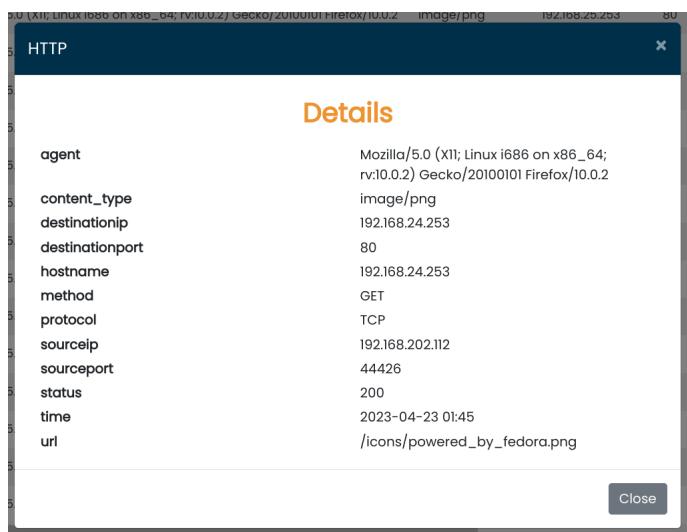
⁶<https://tabulator.info/docs/5.4>

⁷<https://www.chartjs.org/docs/4.3.0/>

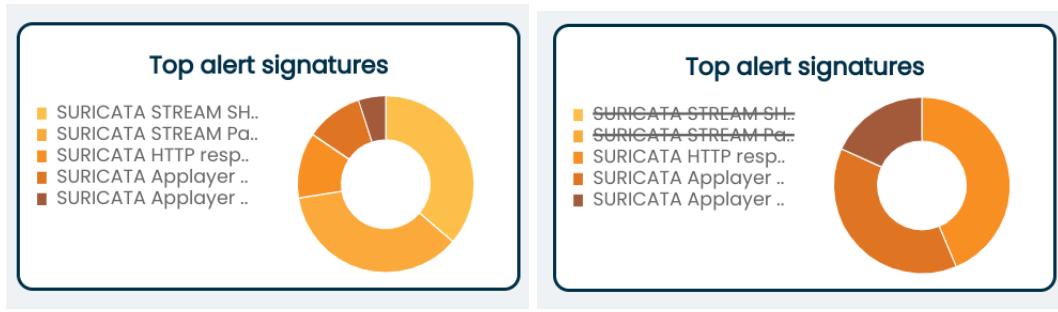
⁸<https://github.com/d3/d3-interpolate>

time	signature	QRadar low level	QRadar high level
2023-04-23 01:45:30	SURICATA Applayer Protocol detection skipped	Unknown Suspicious Event	Suspicious
2023-04-23 01:45:29	SURICATA Applayer Protocol detection skipped	Unknown Suspicious Event	Suspicious
2023-04-23 01:45:29	ET SHELLCODE Rothenburg Shellcode	Unknown	Unknown
2023-04-23 01:44:39	SURICATA STREAM Packet with invalid ack	Unknown Suspicious Event	Suspicious
2023-04-23 01:44:39	SURICATA STREAM SHUTDOWN RST invalid ack	Unknown Suspicious Event	Suspicious
2023-04-23 01:44:38	SURICATA STREAM Packet with invalid ack	Unknown Suspicious Event	Suspicious
2023-04-23 01:44:38	SURICATA STREAM SHUTDOWN RST invalid ack	Unknown Suspicious Event	Suspicious
2023-04-23 01:44:38	SURICATA STREAM Packet with invalid ack	Unknown Suspicious Event	Suspicious
2023-04-23 01:44:38	SURICATA STREAM SHUTDOWN RST invalid ack	Unknown Suspicious Event	Suspicious
2023-04-23 01:44:37	SURICATA STREAM Packet with invalid ack	Unknown Suspicious Event	Suspicious

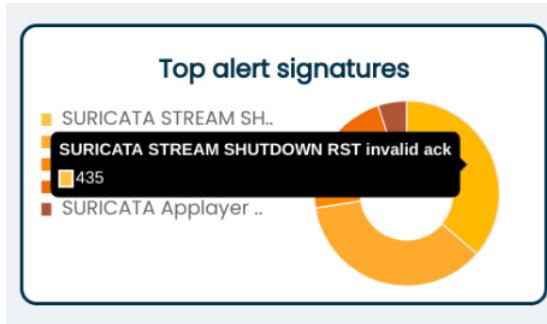
Obr. 6.8: Tabuľka vo frameworku Tabulator



Obr. 6.9: Dialogové okno s detailami



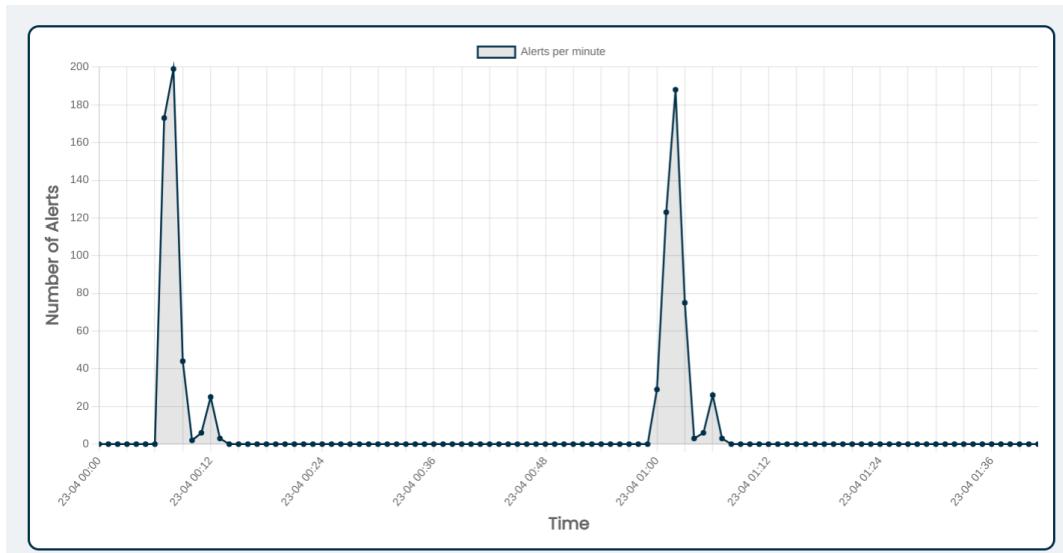
Obr. 6.10: Vzhľad zobrazovaných grafov



Obr. 6.11: Zobrazenie pomoci pri prechode cez graf

Časová os

Na zobrazenie hrozieb v čase sa použil lineárny graf. Ten zobrazuje výskyt hrozieb v zvolenom časovom horizonte. Užívateľovi tak umožňuje pochopíť kedy sa objavilo najviac útokov. V grafe sa zobrazuje zoznam časových stôp po minútach. Na zvýšenie výkonu sa ponechávajú časti grafu v ktorých je zrejmé že sa bola zaznamenaná nejaká aktivita. Tento graf je na obrázku 6.12.



Obr. 6.12: Vzhľad zobrazovaných grafov

6.4 Exportovanie DSM modulu a nasadenie Addonu

Modul DSM bol exportovaný pre komunitnú verziu pomocou štandardného procesu systému QRadar. Keďže už existuje modul, ktorý ale nie je kompatibilný s komunitnou verzou systému QRadar, tak bol nazvaný **SuricataFIT**. Modul DSM bol exportovaný z verzie 7.3.3 a má kompatibilitu so verziami 7.4 a 7.3. Je dostupný na webovej stránke GITHUBU. V tomto repozitári si ho je možné stiahnuť a nainštalovať do systému QRadar ako rozšírenie.

Po dokončení aplikácie vo vývojárskom režime nasleduje nasadenie aplikácie do prevádzkového stavu. Aplikácie v komunitnej verzii musia byť kompatibilné s verzou jazyka Python 2.7.5.. Pre správnu komunikáciu s aplikačným rozhraním systému QRadar je odporúčané použiť spomínanú knihovnu **qpylib**. Tá správne nastaví prístupové údaje pre komunikáciu. Pri procese nasadzovania aplikácia je dôležité kolko pamäte RAM sa určí pre aplikáciu.

Pre prípady že by systém QRadar nedovolil nainštalovať aplikáciu z rôznych dôvodov boli vytvorené dve verzie aplikácie pre užívateľov. Jedna je určená priamo systému QRadar, druhá je určená na spustenie mimo tento systém, avšak je potrebné nastaviť správne environmentálne premenné. Aplikáciu je možné nainštalovať pomocou dotazu na koncový bod REST API na inštaláciu aplikácií a odoslať zabalený súbor v tele dotazu:

```
GET /gui\_app\_framework/application\_creation\_task
```

Ak by vznikol problém s inštaláciou, môže ju užívateľ vyšetriť a prerobiť zdrojový kód priamo v systéme QRadar s rozšírením **QRadar App Editor**⁹. Ak sa nepodarí ani jedna z týchto možností, čo je veľmi pravdepodobné, môže užívateľ jednoducho spustiť Flask aplikáciu so správne nastavenými prístupovými údajmi. Celý postup ako nainštalovať a sprevádzkovať modul aplikácie je v prílohe **B**.

⁹<https://exchange.xforce.ibmcloud.com/hub/extension/5d0f3f37cc5c4d16ccafe9d40d8dff5>

Kapitola 7

Demonštrácia použiteľnosti pri vybraných sietových prevádzkach

Pri implementácii aplikácie sa zameralo na reálne využitie programu Suricata pri analýze sietového toku na rôznych zariadeniach. Kedže plán bol nakonfigurovať program Suricata v základom nastavení ako IDS systém, získal som dva základne typy dát. Prvou kategóriou je odosielanie správ o stave jednotlivých tokoch dát a druhou kategóriou je informovanie používateľa o možných hrozbách. Obidve tieto kategórie sú dôležité pri analýze sietovej prevádzky a navzájom sa dopĺňajú. Na kolko výsledná aplikácia ponúka práve možnosť analýzy v týchto dvoch módoch, aplikácia sa otestuje na každú možnosť zvlášť. Pri testovaní analýzy hrozieb sme sa zamerali aj na otestovanie spustených priestupkov (angl. offenses), s ktorými pracuje hlavne systém QRadar.

7.0.1 Zvolený postup testovania

Program Suricata bol nainštalovaný na notebooku a napojený na odosielanie dát na rozhranie syslog. Použil sa verzia 6.0.11 stiahnutá z oficiálnej stránky¹. Systém QRadar bol nainštalovaný vo virtuálnom prostredí a nastavený tak, aby dokázal prijímať správy z rozhrania syslog v počítači. Stiahnutá bola komunitná verzia². Flask aplikácia bola testovaný vo vývojom móde, kvôli zátaži počítača. Integrácia do systému prebehla a aplikácia je použiteľná aj z vnútra systému QRadar. Využitý bol počítač s touto špecifikáciou:

- operačný systém Linux – 64bit,
- 12 jadier 2.60GHz,
- 16 GB pamäte RAM,

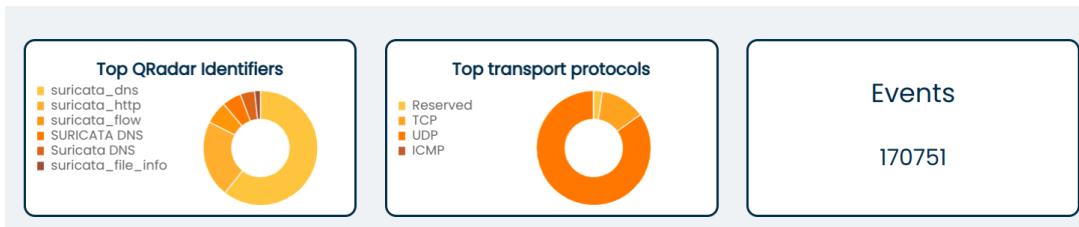
Program Suricata bol nastavený tak aby analyzoval pakety odchádzajúce a idúce na akékoľvek IP adresy. Pre potreby jednotlivých testov sa vypínali a zapínali správy, ktoré mal IDS systém sledovať. Boli použité základné pravidlá a zo sady `et/open`, ktoré všetky sú namapované na kategórie z systému QRadar. V nasledujúcich sekcii je opísaná ako prebiehala demonštrácia dvoch kariet vyvinutej aplikácie. V prvej sekcii 7.1 na získavanie základných správ a dátového toku, ktorý nie je nijak nebezpečný. V druhej sekcii 7.2 je zobrazené ako dokáže systém QRadar analyzovať reálne hrozby a zachytávať priestupky na základe správ generovaných programom Suricata.

¹<https://suricata.io/download/>

²<https://www.ibm.com/community/qradar/ce/>

7.1 Demonštrácia bezpečnej obvykľej prevádzky

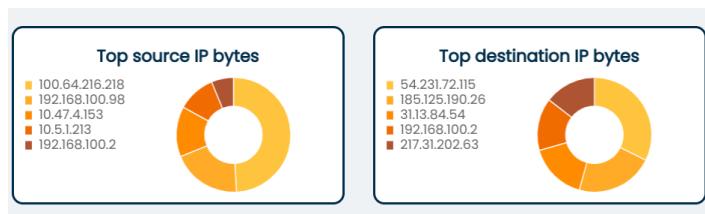
Vyvinutá aplikácia má dve karty. V prvom teste sa použije karta **Messages**, kde sú zobrazené základné informácie o toku dát. Najprv si zvolím záznamový zdroj, ktorý je analyzovaný pomocou vytvoreného modulu DSM a čas, z ktorého chcem získať informácie.



Obr. 7.1: Vzhľad spoločného panelu pre správy

Na vrchu je zobrazený hlavný panel nastavený na typ **flow**. Na obrázku 7.1, je možné vidieť všeobecné informácie pre všetky udalosti: výskyt udalostí, počet udalostí a najpoužívanejšie protokoly. V následujúcich grafoch sa informácie špecifikujú pre toky dát.

Pre demonštráciu analýzy som vybral udalosť **Flow**. Na obrázku 7.3 je možné vidieť najčastejšie použité adresy IP, rovnako to platí aj pre porty a zvyšné hodnoty. Výhodou tejto podstránky je, že v jednej kartičke sa zobrazia všetky toky dát ktoré vytvorili hrozbu.



Obr. 7.2: Grafy o prenesených počtoch bajtov pre adresy IP

Pod grafmi sa v každej karte nachádza tabuľka so všetkými udalosťami v danom časovom rozmedzí. Na každej podstránke je možné nájsť iný typ údajov, špecifikovaný pre danú udalosť. Po rozkliknutí jedného riadka sa zobrazí dialógové okno s detailami.

Kedže demonštrácia zvyšných udalostí by zahrňala veľkú réžiu obrázkov, nebola zahrnutá do tejto kapitoly. Vo všetkých prípadoch sa jedná o zistovanie správania paketov v sieti. Snímky obrazoviek je možné nájsť v prílohe C.2.

7.2 Demonštrácia analýzy nebezpečnej prevádzky

Bežná prevádzka na internete, ktorú vykonáva užívateľ, neobsahuje žiadne väčšie typy hrozien. Bolo teda potrebné nájsť iný spôsob, ako správne otestovať vyvinutý systém. Prevedenie kyberbezpečnostného útoku na nejakú aplikáciu by bolo zdĺhavé, nebezpečné a vyžadovalo by ďalšie štúdium. Nakoniec som pristúpil k metóde, v ktorej bola zopakovaná možná nebezpečná komunikácia po rozhraní, na ktorom bude počúvať program Suricata.

The screenshot shows a software interface with a dark header bar. Below it, a light-colored panel titled 'Details' in orange. Inside, there's a table with the following data:

alerted	false
app_protocol	null
bytes_toclient	0
bytes_toserver	64
destinationip	192.168.229.254
destinationport	48978
pakets_toclient	0
pakets_toserver	1
reason	timeout
sourceip	192.168.202.79
sourceport	63806
state	new
time	2023-04-18 03:46

At the bottom right of the panel is a dark button labeled 'Close'.

Obr. 7.3: Vzhľad zobrazovaných grafov

7.2.1 Infikované súbory – PCAP

Testovanie systémov IDS prebieha replikáciou súborov, ktoré obsahujú údaje o sietovom toku [13]. Využijú sa pritom súbory typu pcap [18], ktoré slúžia na zachytávanie sietového toku. V súboroch je zachytená reálna komunikácia v čase. Systémy IDS slúžia na zachytávanie hrozieb analyzovaním nebezpečných paketov. Na otestovanie programu Suricata som použil súbory typu pcap s výskytom vírusu trojského koňa. Boli použité nebezpečné súbory z datasetu **stratosphereips**³. Jedná sa o dataset, ktorý ma slúžiť ako ukážka reálnych nebezpečných sietových prevádzok na trénovanie neurónových sietí. Na zadanom súbore sa skúma, či modul DSM dokáže tento vírus detegovať z prichádzajúcich správ programu Suricata. Následne by užívateľ mal byť schopný vidieť v aplikácii zachytené hrozby a pomocou grafov analyzovať typ a zdroj hrozby.

7.2.2 Využitie offline módu programu Suricata

Na replikáciu pcap súborov je možné použiť rôzne programy, ktoré simulujú prechod paketov cez určené rozhranie. Jedná sa o programy ako **tcpreplay**⁴ alebo **Scapy**⁵. Pre prehranie danej sietovej komunikácie boli využité možnosti programu Suricata, ktorý dokáže operovať aj v offline móde. Pri spustení zozbiera dostupné pravidlá a začne replikovať prechod paketov na sieti. Demonštrácia je zameraná na analýzu hrozieb, ktoré detegoval program Suricata.

³<https://www.stratosphereips.org/datasets-malware>

⁴<https://tcpreplay.appneta.com/>

⁵<https://scapy.readthedocs.io/en/latest/>

Sú zistené podrobnosti, odkiaľ pochádzala väčšina útokov a akého sú charakteru. Navyše aplikácia vie zobraziť, či sa neobjavil útok, ktorý by bol spustený nejakou hrozbou.

7.2.3 Prevedenie prvého útoku

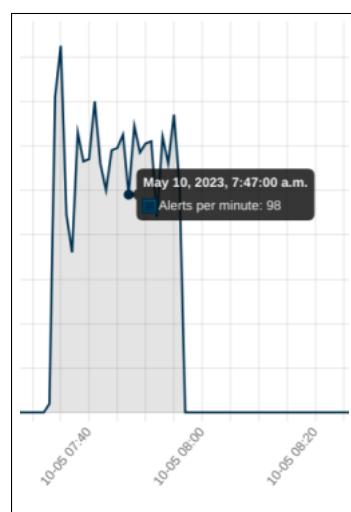
V prvom útoku vyskúšam prehrať infikovaný súbor s vírusom trojský kôň. Systém QRadar dokáže detegovať priestupky na základe určených pravidiel. Na detekciu trojského koňa ale neexistuje žiadne pravidlo, preto bolo potrebné jedno vytvoriť. V systéme QRadar som vytvoril pravidlo, ktoré sa vytvorí ak sa kategória nízkej úrovne danej udalosti zhoduje s kategóriou **Trojan Detected**. Priestupok by sa mal objaviť v spodnej časti stránky, keďže v infikovanom súbore sa má nachádzať vírus typu trojský kôň.

Použil som, teda súbor pcap zo spomínaného datasetu. Tento súbor obsahuje trojského koňa typu Tinba. Počet hrozíeb, ktoré zaznamenal program Suricata v skúmanom súbore je 2839600. Po prevedení útoku by sa mali všetky hrozby zobraziť v systéme QRadar. Väčšina udalostí bola zachytená v systéme na karte **Log Activity**. Zaznamenaných bolo ale oveľa menej pravidiel. Je to dôsledkom obmedzenia pamäte môjho počítača. Na obrázku 7.4 je možné vidieť základné údaje, ktoré boli zachytené programom Suricata a systémom QRadar.

Choose a severity	Overall Alert Counts	Top Signature
<input type="button" value="all"/>	High Alerts: 7077	ET MALWARE Known Sinkhole Response Header
	Medium Alerts: 191	
	Low Alerts: 64192	

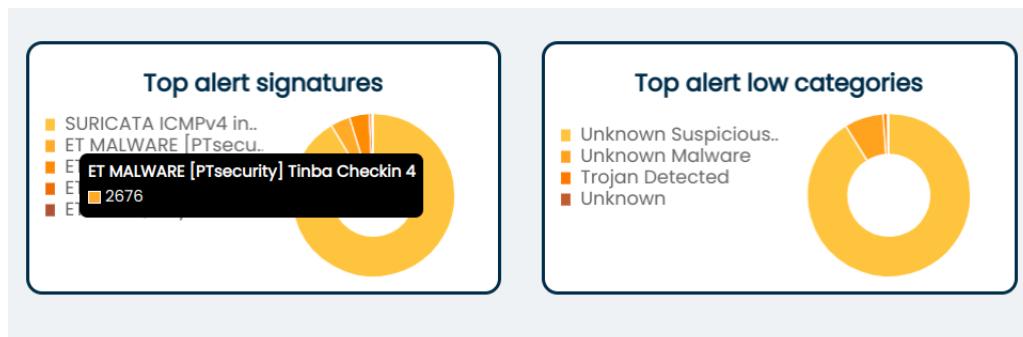
Obr. 7.4: Základné údaje o zachytených hrozbách

Počet hrozíeb je teda značne nižší ako počet ktorý sa nachádzal v súbore. Najčastejšie sa vyskytujuúcim pravidlom je **ET Malware Sinkhole Response Header**. Ďalším prvkom na stránke je časová os (viď obrázok 7.5). Na časovej osi je vidieť rozmedzie v ktorom sa vyskytli hrozby. Po prejdení myšou sa navyše zobrazí aj pomôcka na určenie presného počtu hrozíeb v danom čase.



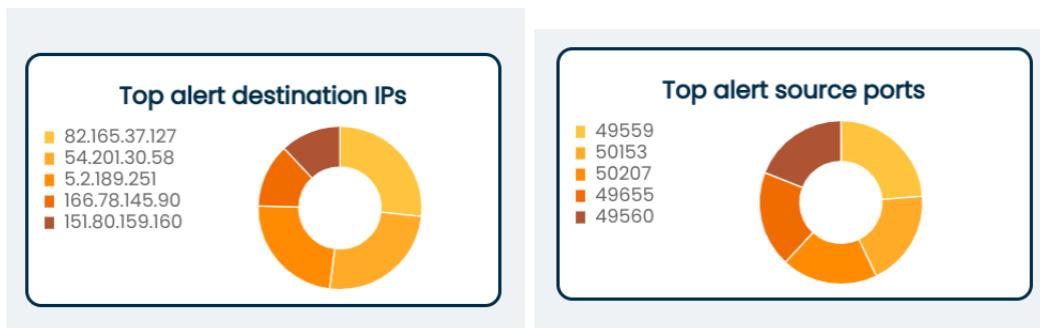
Obr. 7.5: Základné údaje o zachytených hrozbách

V sekcií pod časovou osou sú grafy s kategóriami, s ktorými pracuje systém QRadar (viď obrázok 7.6). Na grafe je hned možné vidieť niekoľko klúčových slôv evokujúcich výskyt vírusu. Tak ako pri časovej osi aj v tomto prípade sa po prejdení po grafe zobrazí najčastejšie najčastejšie sa vyskytujúca hrozba ET MALWARE Tinba 4.



Obr. 7.6: Grafy kategórií a pravidla

Podľa tejto signatúry som vyfiltroval konkrétnu udalosť Tinba⁶. Jedná sa o typického trojského koňa, ktorý môže spôsobiť veľkú škodu v systéme. V grafoch, ktoré sa vykreslili po vyfiltrovaní boli zistené najčastejšie sa vyskytujúce zdrojové adresy IP a porty zobrazené na obrázku 7.7.



Obr. 7.7: Grafy o zdrojových adresách a cielových portoch

Cez tabuľku som si nechal vypísť riadok na ktorom som konkrétnie riadky v dialógovom okne. Dôležité sú údaje o zdroji cieli, ktoré je možné využiť na ďalšiu analýzu v systéme QRadar. Na obrázku 7.8 možné vidieť všetky detaile tejto udalosti.

Detekcia priestupkov

Ako už bolo spomenuté v tejto práci, aplikácia dokáže rozoznať všetky priestupky generované systémom QRadar. Priestupky sa generujú na základe definovaných pravidiel. Pre vírusy existujú v systéme QRadar pravidlá, ktoré by mali spúštať priestupky na základe udalostí. Experimentom som zistil, že priestupky na základe vírusov sa vôbec negenerujú.

Pri prvotnom testovaní som použil súbory, ktoré obsahujú veľmi málo hrozieb a slúžili ako testovacia sada. Obsahujú len hrozby jedného typu. Na tento typ hrozby bol nastavené pravidlo, ktoré vygeneruje priestupok, ten je zaznamenaný a je možné ho zobraziť na karte hrozby v časti pre priestupky ako je na obrázku 7.9. Na obrázku je možné vidieť

⁶<https://history-computer.com/the-tinba-virus-how-it-works-and-how-to-protect-yourself/>

Details	
QRadar event name (signature)	ET MALWARE Tinba Checkin 2
QRadar high category	Malware
QRadar low category	Unknown Malware
destination IP	151.80.159.160
destination port	80
interface	null
protocol	TCP
severity	9
signature	ET MALWARE Tinba Checkin 2
source IP	10.0.2.103
source port	50566
time	2023-05-10 07:56:48

Obr. 7.8: Obrázok detailu pravidla



Obr. 7.9: Zobrazenie hrozieb v systéme

7.3 Zhodnotenie

Demonštráciou aplikácie a modulu DSM bolo preukázané, že systém QRadar a program Suricata dokážu medzi sebou komunikovať a predávať si informácie. Modul DSM analyzuje všetky polia, ktoré sa implementovali. Objavil sa ale menší problém pri mapovaní udalostí, ktorý ako hrozba už bol spomenutý v práci. Objavili sa viaceré udalosti, ktoré nie sú zaradené do kategórii a nemajú patričnú konfiguráciu. Tieto udalosti je možné vidieť len v aplikácii, keďže sa grafy filtrovajú podľa udalosti typu hrozba a nie podľa kategórii systému QRadar.

V prvom prípade testovania som zistil, že aplikácia je vhodná na jednoduché použitie a zistenie základných informácií o toku dát. V časti pre toky bolo zistené odkiaľ a kam prúdi najviac dát. Informácia o počte nebezpečných tokov je dôležitá pre analýzu hrozieb.

V druhom prípade som testoval nebezpečný súbor s trojskym koňom. Podarilo sa identifikovať, o akú hrozbu sa jedná, odkiaľ a kam smeruje a v akom počte sa vyskytuje. Časová os umožňuje zistiť, v akom časovom horizonte sa udalosti vyskytli. Filtrovanie napomohlo k priamej detekcii konkrétnej hrozby. V práci bola otestovaná aj detekcia priestupkov. Vzhľadom na zložitosť systému QRadar sa nepodarilo zobraziť priestupky pri detekcii vírusu trojského koňa.

Aplikáciu teda hodnotím ako prehľadný doplnok do systému QRadar, ktorá napomôže pri analýze detegovaných hrozieb programom Suricata. Navyše karta hrozby prináša relevantné informácie pre zistenie možných útokov a prienikov. Je možné zistiť konkrétnie zdroje a ciele, využité porty ale aj konkrétnie vlastnosti útokov. Z týchto informácií je následne možné vykonať analýzu v programe QRadar.

V práci vidím mierny nedostatok v testovaní, kedže spojiť dokopy všetky štyri systémy je veľmi náročné na techniku. Overenie funkčnosti systému testom na výkon teda možné nie je a v práci sa môžu nachádzať viaceré nedostatky z komplexnosti systémov. Systém QRadar je určený pre veľké softvérové firmy, ktoré využívajú mnohonásobne silnejšie servery, ako je tomu v prípade bežného počítača.

Kapitola 8

Záver

V rámci tejto záverečnej práce bola spracovaná problematika prepojenia IDS systému Suricata a SIEM systému QRadar pre komunitnú verziu dostupného širokej verejnosti. Systém QRadar umožňuje zbierať dátu z rôznych zdrojov a vykonávať nad nimi hlbšiu analýzu. Doteraz neboli pre komunitnú verziu pripravený dostatočný modul na analýzu príchodzích správ z programu Suricata a ďalšiu analýzu si musel užívateľ urobiť sám. V tejto práci bol vyvinutý modul pre analýzu hrozieb s namapovanými základnými pravidlami. Nad týmito pravidlami bola postavená aj aplikácia, ktorú si môže užívateľ stiahnuť do systému QRadar a zobraziť dôležité dátu.

Na základe analýzy som pripravil návrh DSM pre program Suricata, ktorý zbiera informácie o hrozbách a stave siete. Ďalej som pripravil návrh na možné riešenie vizualizácie dát získaných z databázy Ariel. V závislosti na návrhu bol implementovaný modul DSM, do ktorého sa podarilo dostať všetky pravidlá aj napriek tomu, že aplikácia neustále padala. Modul tak dokáže detegovať až 30 000 tisíc možných hrozieb. Implementovaná aplikácia zobrazuje dva typy správ. Jedným typom sú základné záznamy o prevádzke siete a druhým typom sú zuchytené hrozby programom Suricata a priestupky zistené systémom QRadar. Z hrozieb je možné zistiť odkiaľ pochádzajú, kam smerujú a aké najčastejšie typy protokolov boli použité. Navyše sa zobrazuje tabuľka v ktorej je možné vyhľadať konkrétny typ útok.

Po implementácii aplikácie nasledovalo testovanie, ktoré malo overiť základnú funkcionality, a prácu s načítavaním a zobrazením dát. Skúška systému prebehla nad simulovaným útokom. Po prevedení útoku boli zistené hrozby a pôvod útoku nad dátami prúdiacimi do systému QRadar.

Program Suricata obsahuje oveľa viac pravidiel ako bolo reálne namapovaných, preto by bolo vhodné pripraviť skript, ktorý by podľa aktualizovaných pravidiel namapoval novo vzniknuté a aktualizoval mapovanie. Momentálne namapované pravidlá môžu v dôsledku meniacich sa hrozieb rýchle zostarnúť a nemusia byť použiteľné v blízkej dobe. Aplikáciu by bolo vhodné nasadiť do reálnej prevádzky a testovať na reálnych útokoch, nie na simulovaných ako to bolo predvedené v demonštrácii.

Literatúra

- [1] *Application Framework Guide* [online]. IBM [cit. 2023-10-1]. Dostupné z: https://www.ibm.com/docs/en/SS42VS_SHR/pdf/b_qradar_appframework_devguide.pdf.
- [2] *QRadar API endpoint documentation and supported versions* [online]. IBM [cit. 2022-22-12]. Dostupné z: <https://www.ibm.com/docs/en/qradar-common?topic=api-endpoint-documentation-supported-versions>.
- [3] *Ariel Query Language Guide* [online]. IBM [cit. 2023-25-4]. Dostupné z: https://www.ibm.com/docs/en/SSKMUKU/com.ibm.qradar.doc/b_qradar_aql.pdf.
- [4] BRUNNER WILLIAMS, E., MANNING, B. a 3RD, D. E. E. *Domain Name System (DNS) IANA Considerations* [RFC 2929]. RFC Editor, september 2000. DOI: 10.17487/RFC2929. Dostupné z: <https://www.rfc-editor.org/info/rfc2929>.
- [5] CHAKRABARTY, B., PATIL, S. R., SHINGORNIKAR, S., KOTHEKAR, A., MUJUMDAR, P. et al. *Securing Data on Threat Detection by Using IBM Spectrum Scale and IBM QRadar: An Enhanced Cyber Resiliency Solution*. IBM Redbooks, 2021.
- [6] CHAKRABORTY, N. Intrusion detection system and intrusion prevention system: A comparative study. *International Journal of Computing and Business Research (IJCBR)*. 2013, zv. 4, č. 2, s. 1–8.
- [7] COMMONS, W. *Topologie NIDS HIDS*. 2020. Dostupné z: https://commons.wikimedia.org/wiki/File:Topologie_NIDS_HIDS.png.
- [8] DETKEN, K.-O., RIX, T., KLEINER, C., HELLMANN, B. a RENNERS, L. SIEM approach for a higher level of IT security in enterprise networks. In: *2015 IEEE 8th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*. 2015, sv. 1, s. 322–327. DOI: 10.1109/IDAACS.2015.7340752.
- [9] DROMS, R. *Dynamic Host Configuration Protocol* [RFC 2131]. RFC Editor, marec 1997. DOI: 10.17487/RFC2131. Dostupné z: <https://www.rfc-editor.org/info/rfc2131>.
- [10] *DSM Editor overview* [online]. IBM [cit. 2022-22-12]. Dostupné z: <https://www.ibm.com/docs/en/qsip/7.4?topic=qradar-dsm-editor-overview>.
- [11] *DSM guide* [online]. [cit. 2022-22-12]. Dostupné z: https://www.ibm.com/docs/en/SS42VS_DSM/pdf/b_dsm_guide.pdf.

- [12] ELDOW, O., CHAUHAN, P., LALWANI, P. a POTDAR, M. Computer network security ids tools and techniques (snort/suricata). *Int. J. Sci. Res. Publ.* Citeseer. 2016, zv. 6, č. 1, s. 593.
- [13] ENNERT, M., CHOVANCOVÁ, E. a DUDLÁKOVÁ, Z. Testing of IDS model using several intrusion detection tools. *Journal of Applied Mathematics and Computational Mechanics*. Politechnika Częstochowska. Wydawnictwo Politechniki Częstochowskiej. 2015, zv. 14, č. 1, s. 55–62.
- [14] *Event mapping* [online]. IBM, 2022 [cit. 2022-22-12]. Dostupné z: <https://www.ibm.com/docs/en/qsip/7.4?topic=qradar-event-mapping>.
- [15] FUCHSBERGER, A. Intrusion Detection Systems and Intrusion Prevention Systems. *Information Security Technical Report*. 2005, zv. 10, č. 3, s. 134–139. DOI: <https://doi.org/10.1016/j.istr.2005.08.001>. ISSN 1363-4127. Dostupné z: <https://www.sciencedirect.com/science/article/pii/S1363412705000415>.
- [16] GONZÁLEZ GRANADILLO, G., GONZÁLEZ ZARZOSA, S. a DIAZ, R. Security Information and Event Management (SIEM): Analysis, Trends, and Usage in Critical Infrastructures. *Sensors*. MDPI AG. Jul 2021, zv. 21, č. 14, s. 4759. DOI: 10.3390/s21144759. ISSN 1424-8220. Dostupné z: <http://dx.doi.org/10.3390/s21144759>.
- [17] GUPTA, S., CHAUDHARI, B. S. a CHAKRABARTY, B. Vulnerable network analysis using war driving and security intelligence. In: *2016 International Conference on Inventive Computation Technologies (ICICT)*. 2016, sv. 3, s. 1–5. DOI: 10.1109/INVENTIVE.2016.7830165.
- [18] HARRIS, G. a RICHARDSON, M. *PCAP Capture File Format*. Internet-Draft draft-ietf-opsawg-pcap-02. Internet Engineering Task Force, január 2023. Work in Progress. Dostupné z: <https://datatracker.ietf.org/doc/draft-ietf-opsawg-pcap/02/>.
- [19] JANES, A., SILLITTI, A. a SUCCI, G. Effective dashboard design. *Cutter IT Journal*. Január 2013, zv. 26, s. 17–24.
- [20] MASSE, M. *REST API Design Rulebook*. O'Reilly Media, 2011. Oreilly and Associate Series. ISBN 9781449310509. Dostupné z: <https://books.google.cz/books?id=4lZcsRwXo6MC>.
- [21] MURCHISON, K. a CRISPIN, M. *Internet Message Access Protocol - SORT and THREAD Extensions* [RFC 5256]. RFC Editor, jún 2008. DOI: 10.17487/RFC5256. Dostupné z: <https://www.rfc-editor.org/info/rfc5256>.
- [22] NIELSEN, H., MOGUL, J., MASINTER, L. M., FIELDING, R. T., GETTYS, J. et al. *Hypertext Transfer Protocol – HTTP/1.1* [RFC 2616]. RFC Editor, jún 1999. DOI: 10.17487/RFC2616. Dostupné z: <https://www.rfc-editor.org/info/rfc2616>.
- [23] OISF. *Suricata User Guide*. 2022 [cit. 2022-22-12]. Dostupné z: https://suricata.readthedocs.io/_/downloads/en/suricata-6.0.9/pdf/.

- [24] SINGH, A. P. a SINGH, M. D. Analysis of host-based and network-based intrusion detection system. *International Journal of Computer Network and Information Security*. Modern Education and Computer Science Press. 2014, zv. 6, č. 8, s. 41–47.
- [25] SMITH, V. Data Dashboard as Evaluation and Research Communication Tool. *New Directions for Evaluation*. December 2013, zv. 2013. DOI: 10.1002/ev.20072.
- [26] *QRadar events and flows* [online]. IBM, 2022 [cit. 2022-22-12]. Dostupné z: <https://www.ibm.com/docs/en/qrip/7.4?topic=overview-qradar-events-flows>.

Zoznam príloh

A Obsah priloženého pamäťového média	55
B Manuál na inštaláciu modulu DSM a aplikácie pre systém QRadar	56
B.1 Inštalácia modulu DSM	56
B.2 Inštalácia rozšírenia pre QRadar	56
C Snímky obrazovky z aplikácie	58
C.1 Snímky obrazovky z karty hrozby	58
C.2 Snímky obrazovky z karty správy	62

Príloha A

Obsah priloženého pamäťového média

```
/  
├── modul-dsm/ ..... zložka s modulom DSM  
│   └── suricata-dsm.zip ..... zabalený modul DSM  
├── qradar-suricata-addon/ ..... zložka s dvoma verziami aplikácií  
│   ├── qradar-suricata-flask/ ..... zdrojové kódy aplikácie do programu QRadar  
│   └── suricata-flask/ ..... zdrojové kódy aplikácie  
└── xkozak18/ ..... zdrojové kódy v LATEX  
    └── xkozak18.pdf ..... znenie tejto práce  
    └── manual.pdf ..... manuál na inštaláciu  
    └── README ..... informácie o tomto disku
```

Príloha B

Manuál na inštaláciu modulu DSM a aplikácie pre systém QRadar

Pre inštaláciu modulu DSM a aplikácie je potrebné mať dostupný QRadar v komunitnej verzii na serveri alebo lokálne. Komunitné vydanie musí byť vo verzii 7.3. Pred začiatkom instalácie si prosím uložte aktuálnu verziu systému QRadar.

B.1 Inštalácia modulu DSM

Modul je možné nainštalovať z priloženej SD karty alebo z repozitára:

<https://github.com/tink0mar/suricata-dsm>

Po stiahnutí je potrebné spustiť manažér rozšírení (angl. Extension Manager), nahrať zip súbor spustiť inštaláciu.

B.2 Inštalácia rozšírenia pre QRadar

Na inštaláciu je potrebné stiahnuť zip súbor z priloženej SD karty alebo z tohto odkazu:

<https://github.com/tink0mar/suricata-qradar-app>

Súbor je následne potrebné nahrať cez REST API systému QRadar a využiť pritom koncový bod:

`POST /gui_app_framework/application_creation_task`

Na správne nahranie súboru je potrebné odoslať spolu s dotazom stiahnutý súbor a počkať pokiaľ sa aplikácia nainštaluje. Ak sa inštalácia nepodarí musí sa obnoviť predošlá verzia systému QRadar. Ostáva už len možnosť nainštalovať aplikáciu cez editor aplikácií (angl. App Editor), dostupný na adrese:

<https://exchange.xforce.ibmcloud.com/hub/extension/5d0f3f37cc5c4d16ccafe9d40d8dff5>

Aplikácia sa inštaluje tým istým spôsobom ako modul DSM cez manažér rozšírení. Po nainštalovaní a otvorení aplikácie sa zobrazí dialógové okno v ktorom je potrebné zvoliť možnosť inštalácie existujúcej aplikácie.

Ak ani jedna z týchto možností nefunguje tak je možné použiť druhú verziu aplikácie dostupnú v tom istom repozitári v priečinku `suricata-flask` alebo na SD karte. Na získanie prístupov v aplikácii sa musia nastaviť dve enviromentálne premenné. Jedna sa týka adresy URL pre systém QRadar a druhá sa týka autorizačného tokenu. Ten je možné vytvoriť v systéme QRadar podľa návodu na adrese:

<https://www.ibm.com/docs/en/qradar-common?topic=app-creating-authorized-service-token-qradar-operations>

Aplikácia je následné možné spustiť vo virtuálnom prostredí.

Príloha C

Snímky obrazovky z aplikácie

C.1 Snímky obrazovky z karty hrozby

The screenshot shows the Suricata Alerts interface. On the left, there is a vertical orange sidebar with the title "Suricata" at the top, followed by "Alerts" and "Messages". The main area has a white background with a header titled "Suricata Alerts". In the header, there is a QRadar Ariel icon with a green "ON" button, a "help" link, and a dropdown menu for "Suricata Log Sources" set to "MySuricata". Below the header, there are two input fields: "Start Time" (2023-04-23 00:00) and "End Time" (2023-05-09 15:02), and a "Fetch Data" button. The main content area is titled "Alerts Records" and contains three sections: "Choose a severity" (with a dropdown menu set to "all"), "Overall Alert Counts" (listing High Alerts: 7, Medium Alerts: 16, Low Alerts: 1349), and "Top Signature" (listing SURICATA HTTP response header invalid).

High Alerts:	7
Medium Alerts:	16
Low Alerts:	1349

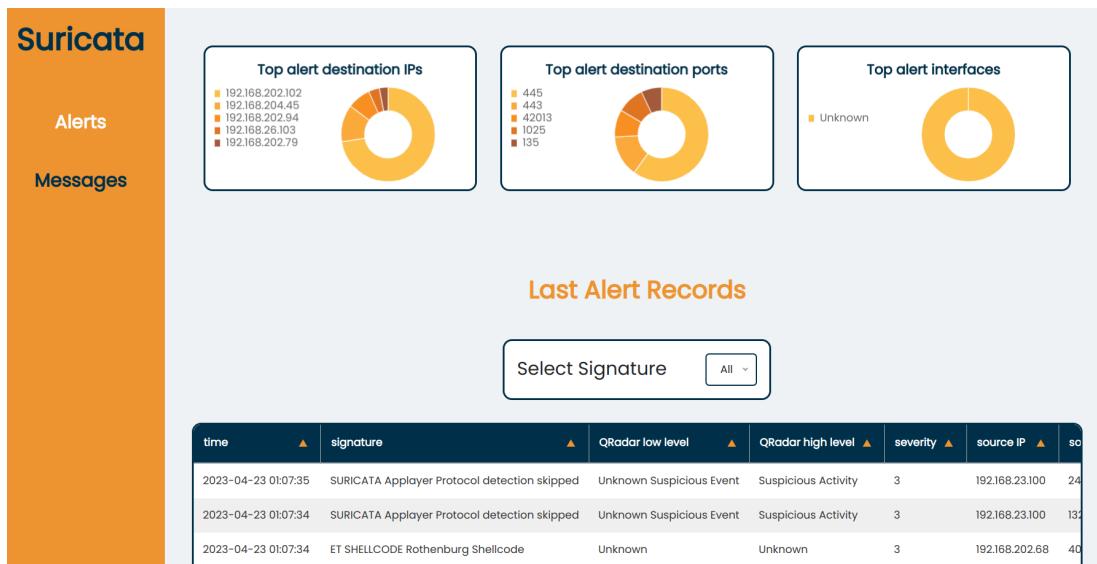
Obr. C.1: Vzhľad vrchného panelu karty hrozby



Obr. C.2: Vzhľad časovej osi



Obr. C.3: Grafy na karte hrozby

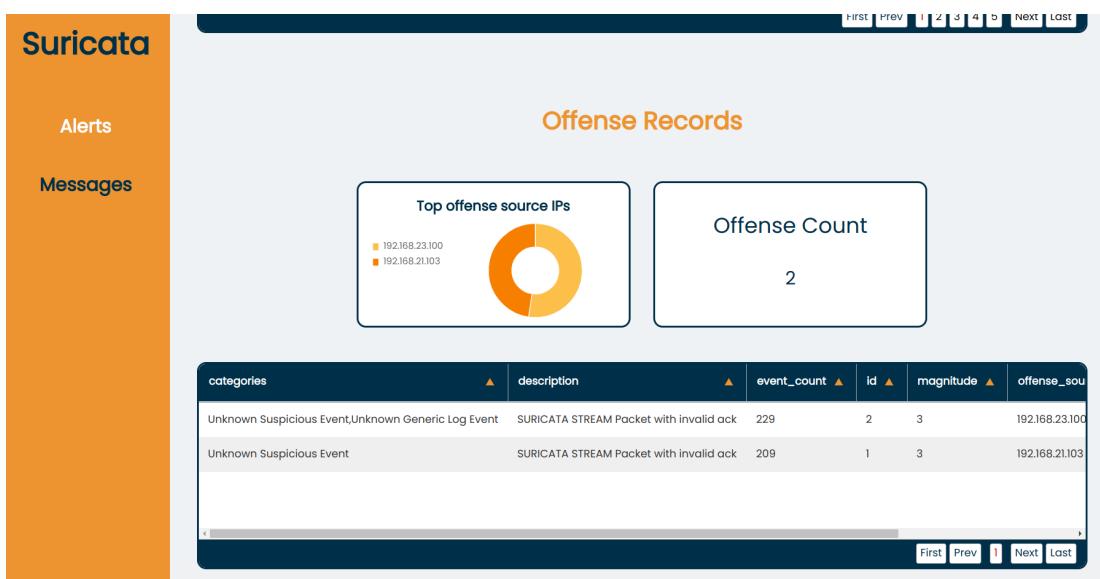


Obr. C.4: Grafy a začiatok tabuľky

The figure shows the Suricata interface with the same layout as Figure C.4. The "Last Alert Records" table is more populated, showing numerous rows of data. The columns are identical to Figure C.4. The table continues beyond the visible portion of the screen, indicated by a scroll bar.

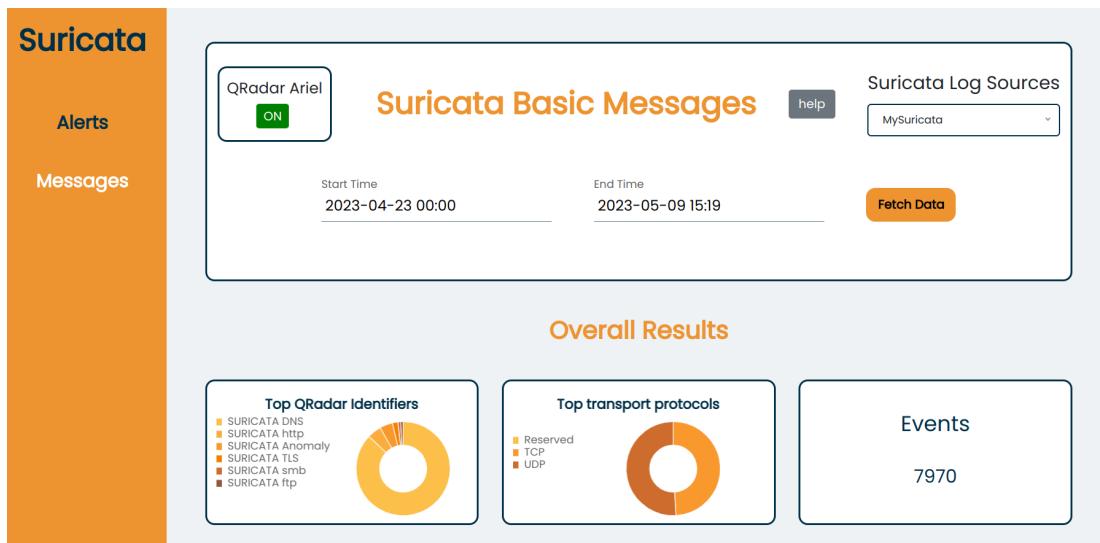
time	signature	QRadar low level	QRadar high level	severity	source IP	sc
2023-04-23 01:07:35	SURICATA Applayer Protocol detection skipped	Unknown Suspicious Event	Suspicious Activity	3	192.168.23.100	24
2023-04-23 01:07:34	SURICATA Applayer Protocol detection skipped	Unknown Suspicious Event	Suspicious Activity	3	192.168.23.100	132
2023-04-23 01:07:34	ET SHELLCODE Rothenburg Shellcode	Unknown	Unknown	3	192.168.202.68	40
2023-04-23 01:06:44	SURICATA STREAM Packet with invalid ack	Unknown Suspicious Event	Suspicious Activity	9	192.168.23.100	52
2023-04-23 01:06:44	SURICATA STREAM SHUTDOWN RST invalid ack	Unknown Suspicious Event	Suspicious Activity	3	192.168.23.100	52
2023-04-23 01:06:44	SURICATA STREAM SHUTDOWN RST invalid ack	Unknown Suspicious Event	Suspicious Activity	3	192.168.23.100	101
2023-04-23 01:06:44	SURICATA STREAM Packet with invalid ack	Unknown Suspicious Event	Suspicious Activity	9	192.168.23.100	101
2023-04-23 01:06:42	SURICATA STREAM Packet with invalid ack	Unknown Suspicious Event	Suspicious Activity	9	192.168.23.100	101
2023-04-23 01:06:42	SURICATA STREAM SHUTDOWN RST invalid ack	Unknown Suspicious Event	Suspicious Activity	3	192.168.23.100	101
2023-04-23 01:06:41	SURICATA STREAM Packet with invalid ack	Unknown Suspicious Event	Suspicious Activity	9	192.168.23.100	52

Obr. C.5: Tabuľka na karte hrozby

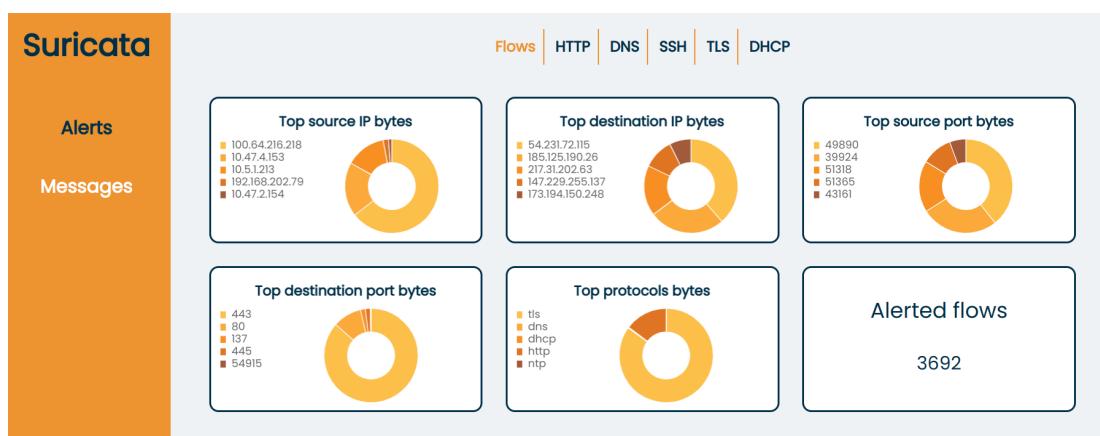


Obr. C.6: Časť karty hrozby zobrazujúca informácie o priestupkoch

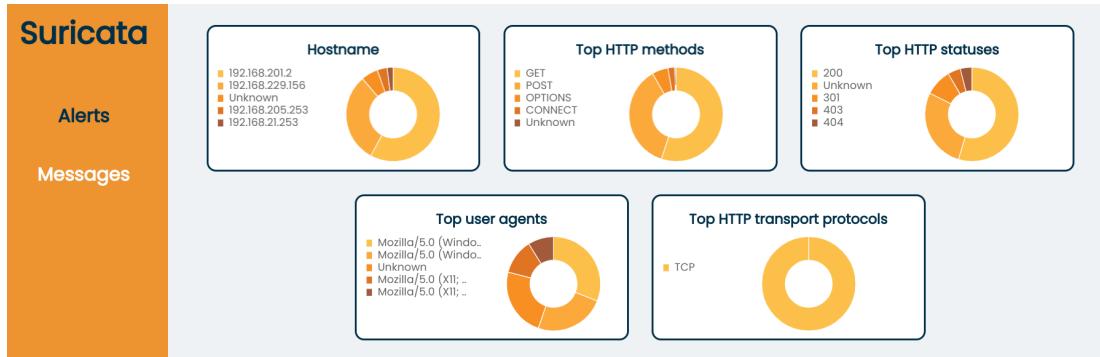
C.2 Snímky obrazovky z karty správy



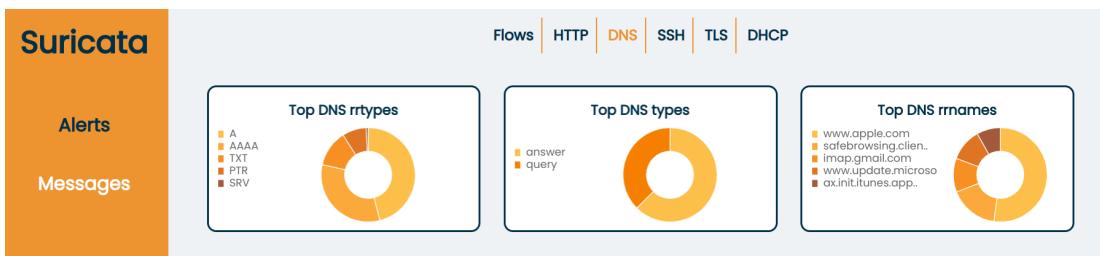
Obr. C.7: Vrchný panel na karte správy



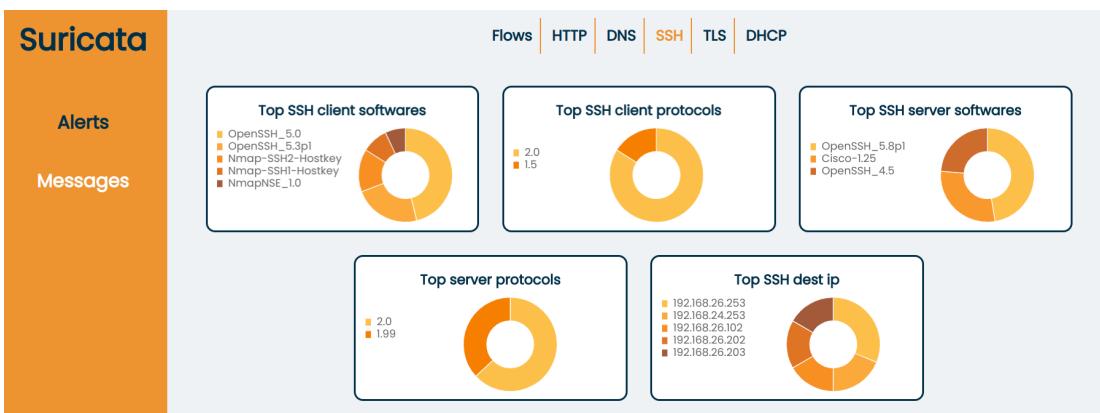
Obr. C.8: Grafy zobrazujúce údaje o type udalosti Flow



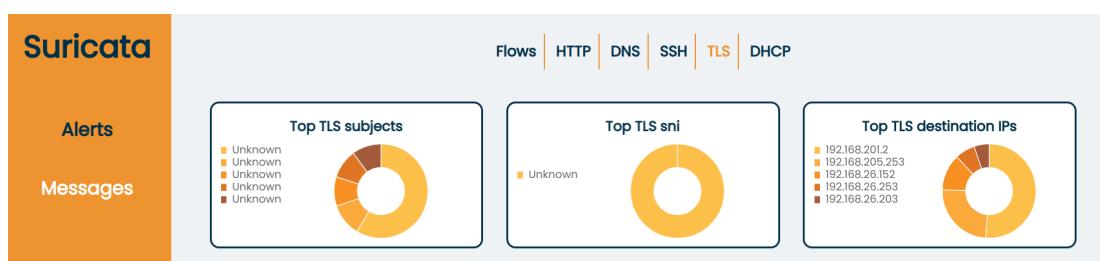
Obr. C.9: Grafy zobrazujúce údaje o type udalosti HTTP



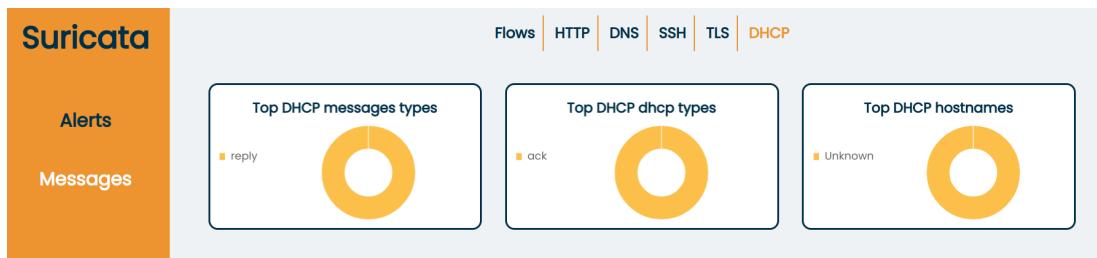
Obr. C.10: Grafy zobrazujúce údaje o type udalosti DNS



Obr. C.11: Grafy zobrazujúce údaje o type udalosti SSH



Obr. C.12: Grafy zobrazujúce údaje o type udalosti TLS



Obr. C.13: Grafy zobrazujúce údaje o type udalosti DHCP

The figure shows a table titled 'Last Flow Records' under the Suricata interface. The left sidebar has 'Suricata' at the top, followed by 'Alerts' and 'Messages'. Above the table is a horizontal timeline with two yellow markers. The table has columns: alerted, app_protocol, bytes_toclient, bytes_toserver, destinationip, destinationport, pakets_toclient, and pakets_toserve. The data is as follows:

alerted	app_protocol	bytes_toclient	bytes_toserver	destinationip	destinationport	pakets_toclient	pakets_toserve
false		0	64	192.168.229.254	48978	0	1
false		64	64	192.168.229.156	27809	1	1
false		0	64	192.168.229.254	23466	0	1
false		64	64	192.168.229.153	27280	1	1
false		64	64	192.168.229.156	11003	1	1
false		64	64	192.168.229.153	55417	1	1
false		0	64	192.168.229.254	11012	0	1
false		0	64	192.168.229.254	47107	0	1
false		0	64	192.168.229.254	42973	0	1
false		64	64	192.168.229.251	47367	1	1

Obr. C.14: Tabuľka v karte správy