Masarykova univerzita
Fakulta informatiky

Monitorovanie sieťových tokov protokolu Kerberos

Bakalárska práca

Matúš Bačkor

Brno, jar 2015

Prehlásenie

Prehlasujem, že táto bakalárska práca je mojím pôvodným autorským dielom, ktoré som vypracoval samostatne. Všetky zdroje, pramene a literatúru, ktoré som pri vypracovaní používal alebo z nich čerpal, v práci riadne citujem s uvedením úplného odkazu na príslušný zdroj.

Matúš Bačkor

Vedúci práce: doc. Ing. Pavel Čeleda, Ph.D.

Poďakovanie

Chcel by som poďakovať doc. Ing. Pavlovi Čeledovi, Ph.D. za cenné rady a odborné vedenie tejto práce. Taktiež ďakujem svojej rodine a kamarátom za podporu.

Zhrnutie

Bakalárska práca sa zaoberá monitorovaním sieťových tokov autentizačného protokolu Kerberos. Popisuje princípy monitorovania sieťovej prevádzky s dôrazom na monitorovanie pomocou IP tokov. Ďalej sa sústredí na komunikáciu pomocou protokolu Kerberos a možnosti jej analýzy. Popisuje fungovanie FlowMon exportéra a jeho aplikačného rozhrania na tvorbu zásuvných modulov. V rámci práce bol navrhnutý a implementovaný nástroj na získavanie informácií zo sieťových tokov protokolu Kerberos vo forme zásuvného modulu pre FlowMon exportér. Následne sa práca zaoberá testovaním modulu na výkon a overenie jeho funkčnosti. V poslednej časti sú opísane ďalšie možnosti monitorovania protokolu Kerberos.

Kľúčové slová

Kerberos, NetFlow, IPFIX, FlowMon, IP toky, sonda, exportér, kolektor

Obsah

1 **Úvod** 2
2 **Monitorovanie sieťovej prevádzky** 3
2.1 *Protokol NetFlow* 3
2.2 *Protokol IPFIX* 4
2.3 *Architektúra monitorovania siete pomocou IP tokov* . . . 5
3 **Protokol Kerberos** 7
3.1 *Priebeh komunikácie* 7
3.1.1 Získanie lístka TGT 8
3.1.2 Získanie lístka TGS 9
3.1.3 Žiadost o službu 10
3.2 *Nešifrovaný obsah správ* 10
4 **Programové vybavenie exportéra** 13
4.1 *FlowMon exportér 4.0* 13
4.1.1 Základné použitie exportéra 15
4.1.2 Rozširujúce zásuvné moduly 16
4.1.3 Aplikačné rozhranie zásuvných modulov 17
4.2 *Procesný zásuvný modul process-kerberos* 18
4.3 *Implementácia zásuvného modulu* 19
4.3.1 Fukcia init() 20
4.3.2 Spracovávanie správ protokolu Kerberos 20
5 **Výsledky meraní** 22
5.1 *Meranie výkonu* 22
5.2 *Overenie funkčnosti modulu* 24
6 **Ďalšie možnosti monitorovania protokolu Kerberos** . . 27
6.1 *Analyzátory paketov* 27
6.2 *Monitorovanie pomocou logu* 28
7 **Záver** 30
A **Prílohy** 34
A.1 *Zoznam elektronických príloh* 34

1 Úvod

Za posledných niekoľko desaťročí sa informačné technológie dostali medzi najrýchlejšie rozvíjajúce sa odvetvia vyspelej spoločnosti. Pre mnohých ľudí je dnes Internet neoddeliteľnou súčasťou každodenného života. Medzi najčastejšie činnosti bežných užívateľov patrí napríklad prehliadanie webu, posielanie elektronickej pošty, zdieľanie multimediálnych dát a dokumentov, komunikácia s verejnou správou či bankou. S rozvojom celosvetovej siete sa objavuje aj množstvo hrozieb, ktoré sú postavené zraniteľnosti v systémoch, a tým spôsobujú škodu ostatným užívateľom. Preto je pri využívaní niektorých služieb potrebné, aby zúčastnené strany mali istotu, s kým komunikujú, a aby bolo takéto spojenie bezpečné. Pri zaisťovaní identity zúčastnených strán sa používajú rôzne autentizačné metódy, jednou z ktorých je aj protokol Kerberos [1]. Tento protokol bol vyvinutý za účelom vzájomnej komunikácie medzi serverom a klientom v nezabezpečenej sieťovej prevádzke, pričom je využitá dôveryhodná tretia strana. Momentálne je aplikovaný firmou Microsoft ako základná metóda sieťového overovania identity pri použití služieb.

Cieľom práce je analyzovať autentizačný protokol Kerberos a následne navrhnúť nástroj, ktorý umožňuje monitorovanie tohto protokolu na princípe IP tokov. Pri zoznamovaní sa s priebehom komunikácie protokolu Kerberos bol využitý Wireshark [**?**], čo predstavuje kvalitný nástroj na analýzu sieťovej prevádzky. Pomocou neho bolo určené, o aké informácie je možne rozšíriť IP toky na monitorovanie protokolu Kerberos. Navrhnuté riešenie bolo implementované ako zásuvný modul pre FlowMon exportér.

V prvej časti práce sú opísané možnosti monitorovania sieťovej prevádzky s dôrazom na monitorovanie pomocou IP tokov a 2 protokoly, ktoré na tomto princípe fungujú. Tretia kapitola sa zaoberá fungovaním protokolu Kerberos spolu s rozborom údajov, ktoré sú použité na rozšírenie sieťových tokov. Štvrtá kapitola opisuje programové vybavenie FlowMon exportéra a taktiež zásuvný modul, ktorý sa stará o spracovávanie správ protokolu Kerberos. V ďalšej časti sú prezentované výsledky z merania výkonu daného modulu a overenie jeho funkcionality. Na záver sú zobrazené ďalšie možnosti pre monitorovanie protokolu Kerberos.

2 Monitorovanie sieťovej prevádzky

Monitorovanie sieťovej prevádzky znamená neustále sledovanie siete za účelom získania užitočných informácií z jej rôznych častí. Pomocou týchto informácií je možné sledovať stav siete a odhaľovať rôzne neštandardné situácie, ako napríklad podozrivú užívateľskú aktivitu, nestabilitu spôsobenú preťažením a pod.

Jednou z metód monitorovania siete je **štatistické sledovanie prevádzky**. Pomocou neho je možné zistiť objem prenesených dát, ktorý sa dá získať zo sieťových zariadení. Takéto monitorovanie umožňuje prehľad o množstve sieťovej prevádzky v reálnom čase, ale neumožňuje získať podrobnejšie informácie o komunikácii.

Ďalším spôsobom je **hĺbkové skúmanie paketov** (*deep packet inspection*), ktoré pracuje na L4 až L7 vrstve ISO/OSI [5] modelu. Táto metóda poskytuje detailné informácie o sieťovom spojení. Je využívaná pri odhaľovaní rôznych útokov a neoprávnenému prieniku do siete. Nevýhodou pri hĺbkovom skúmaní paketov je vysoká časová náročnosť pri veľkom počte prenesených paketov.

Monitorovanie siete pomocou IP tokov, narozdiel od hĺbkového skúmania paketov, prechádza iba hlavičky IP paketov. To umožňuje získať údaje, ako sú napríklad IP adresy a porty komunikácie. Pakety s rovnakými vlastnosťami, ktoré sú najčastejšie zdrojová IP adresa, zdrojový port, cieľová IP adresa, cieľový port a transportný protokol, sa nazývajú tok. Pre každý tok je vytvorený záznam toku (*flow record*), do ktorého sa ukladajú získané informácie z hlavičiek paketov.

2.1 Protokol NetFlow

NetFlow [6] je technológia vyvinutá spoločnosťou Cisco Systems v roku 1996 založená na monitorovaní sieťovej prevádzky pomocou IP tokov. Poskytuje možnosti pre riešenie mnohých problémov, ako sú:

- analýza nových aplikácií a ich vplyv na sieť,
- odhalenie preťaženia na sieťových uzloch,
- detekcia neoprávnenej sieťovej prevádzky,

- detekcia anomálií v sieti,
- overenie kvality služby (QoS).

NetFlow vyšiel v niekoľkých verziách, pričom medzi najpoužívanejšie verzie patria NetFlow v5 a NetFlow v9.

Verzia	**Popis**
v1	Prvotná verzia, zastaraná
v2 - v4	Interné verzie, neboli nikdy vydané
v5	Najpoužívanejšia verzia, jednoduchý formát, bez podpory IPv6
v6	Nepodporovaná verzia
v7	Verzia špecifická pre Cisco Catalyst smerovače
v8	Umožňuje agregáciu tokov verzie 5 do 11 schém
v9	Šablónová štruktúra, flexibilný formát exportu, podpora IPv6

Tabuľka 2.1: Prehlad verzií protokolu NetFlow

NetFlow v5 predstavuje jednoduchý a kompaktný formát, ktorý obsahuje 18 exportovaných políčok. Nevýhodou tohto protokolu je fixný formát a veľkosť záznamu. NetFlow v5 taktiež nepodporuje protokol IP verzie 6. Z dôvodu chýbajúcej flexibility záznamu bol vytvorený protokol **NetFlow v9**, definovaný v dokumente RFC 3954 [7], ktorý je založený na šablónach. Pomocou nich umožňuje NetFlow kolektorom spracovávať NetFlow dáta bez toho, aby poznali dopredu formát dát. Šablóny sú používané na opis typu a veľkosti políčok posielaných v NetFlow záznamoch.

2.2 Protokol IPFIX

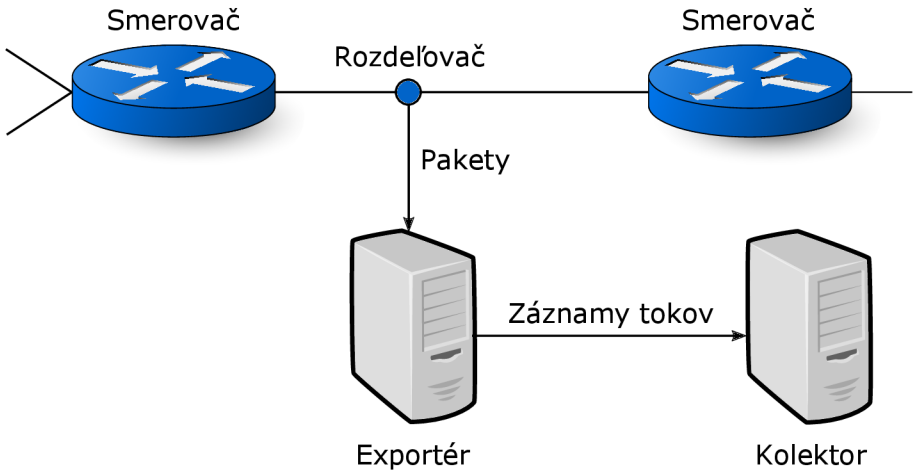
IP Flow Information Export (IPFIX) je protokol vyvinutý skupinou IETF v roku 2008. Cieľom bolo vytvoriť univerzálny a spoločný štandard pre export informácií o IP tokoch. IPFIX je založený na základe protokolu NetFlow v9 a je definovaný v dokumentoch RFC 7011 [8] až RFC 7015 [9] a RFC 5103 [10]. Narozdiel od NetFlow v9 dovoľuje IPFIX definíciu vlastných políčok v rámci organizácie, ktoré sa nazývajú

enterprise elements. Štandardné údaje dostupné pre IPFIX udržiava organizácia *IANA* (Internet Assigned Numbers Authority).

2.3 Architektúra monitorovania siete pomocou IP tokov

Monitorovanie siete pomocou IP tokov je možné vykonávať na niektorých smerovačoch, ktoré podporujú túto doplnkovú službu. Nevýhodou je však výkon, ktorý je na vysoko rýchlostných sieťach nedostačujúci, a taktiež zraniteľnosť voči útokom na daný smerovač a s tým spojená strata možnosti monitorovať sieťovú prevádzku.

Modernejší prístup využíva sieťové sondy (napr. *FlowMon*), čo sú samostatné zariadenia, ktoré pasívne monitorujú sieť. Sú pripojené k sieti pomocou rozbočovačov, čím sa stávajú pre útočníka neviditeľnými. Sonda najčastejšie posiela spracované sieťové toky pomocou NetFlow správ na kolektor, kde sú dáta ukladané. Schéma architektúry je zobrazená na obrázku 2.1.



Obr. 2.1: Schéma monitorovania siete pomocou IP tokov [4]

Exportér (sonda) je zariadenie, ktoré analyzuje a zoskupuje pakety v pamäti tokov (*flow cache*), ktoré následne exportuje vo forme záznamov tokov (napr. pomocou protokolu IPFIX alebo NetFlow), na jeden alebo viac kolektorov. Ako transportné protokoly sa používajú SCTP, TCP alebo UDP.

Kolektor je zariadenie s veľkou diskovou kapacitou, ktoré je zodpovedné za príjem, ukladanie a spracovanie záznamov tokov z jedného alebo viacerých exportérov. Získané štatistiky o sieťovej prevádzke je možné zobraziť napríklad pomocou webového rozhrania. Okrem grafického zobrazenia pomocou grafov je možné záznamy zoskupovať, filtrovať, prípadne automaticky reagovať na preddefinované situácie.

3 Protokol Kerberos

Kerberos je sieťový autentizačný protokol, ktorý bol vytvorený na zaistenie vzájomného overenia identity v nezabezpečených a nedôveryhodných sieťach. K tomu je využívané symetrické šifrovanie a tretia dôveryhodná strana. Protokol Kerberos bol navrhnutý v rámci projektu *Athena* [11] na MIT. Prvé tri verzie tohto protokolu boli vyvíjané interne a neboli nikdy vydané. V súčasnosti sa používa Kerberos verzie 5, ktorý vznikol za účelom odstránenia bezpečnostných problémov a obmedzení, ktoré obsahovala verzia 4.

Pre protokol Kerberos sú charakteristické nasledujúce prvky:

- Je založený na dôveryhodnej tretej strane.
- Vykonáva vzájomné overenie identity.
- Na autentizáciu klienta sa po sieti neposiela samotné heslo, ale iba časová pečiatka zašifrovaná kľúčom odvodeným z užívateľského hesla.
- V rámci jednej relácie je na využívanie viacerých služieb požadované iba jedno zadanie hesla užívateľom.

3.1 Priebeh komunikácie

Protokol Kerberos, primárne navrhnutý na princípe klient-server modelu, umožňuje komunikujúcim stranám preukázať svoju identitu v nezabezpečených sieťach za účasti tretej strany, centru na distribúciu lístkov, tzv. *key distribution center* (KDC). Táto entita sa skladá z autentizačného servera a servera na distribúciu lístkov. KDC obsahuje zoznam všetkých komunikujúcich strán, s ktorými zdieľa tajomstvo - kľúč, a je nimi považovaná za dôveryhodnú.

Kerberos je založený na princípe lístkov, kde KDC plní rolu subjektu, ktorý overuje identitu klienta a udeľuje mu lístky. Pomocou nich je užívateľ schopný preukázať svoju identitu aplikačnému serveru, na ktorom sa nachádza služba, o ktorú má klient záujem. Výhodou takéhoto modelu je, že užívateľ nemusí zadávať heslo pri každej žiadosti

o novú službu, ale využíva lístky k overeniu svojej identity. Navyše zjednodušuje úlohy aplikačným serverom, ktoré nemusia uchovávať žiadne informácie o klientoch na ich autentizáciu. Naopak, nevýhodou je, že KDC je v prípade výpadku jediným bodom zlyhania celého systému, pretože užívateľ nie je schopný získať lístok, aby mohol preukázať svoju identitu aplikačnému serveru. Celkový priebeh komunikácie je zobrazený na obrázku 3.1.

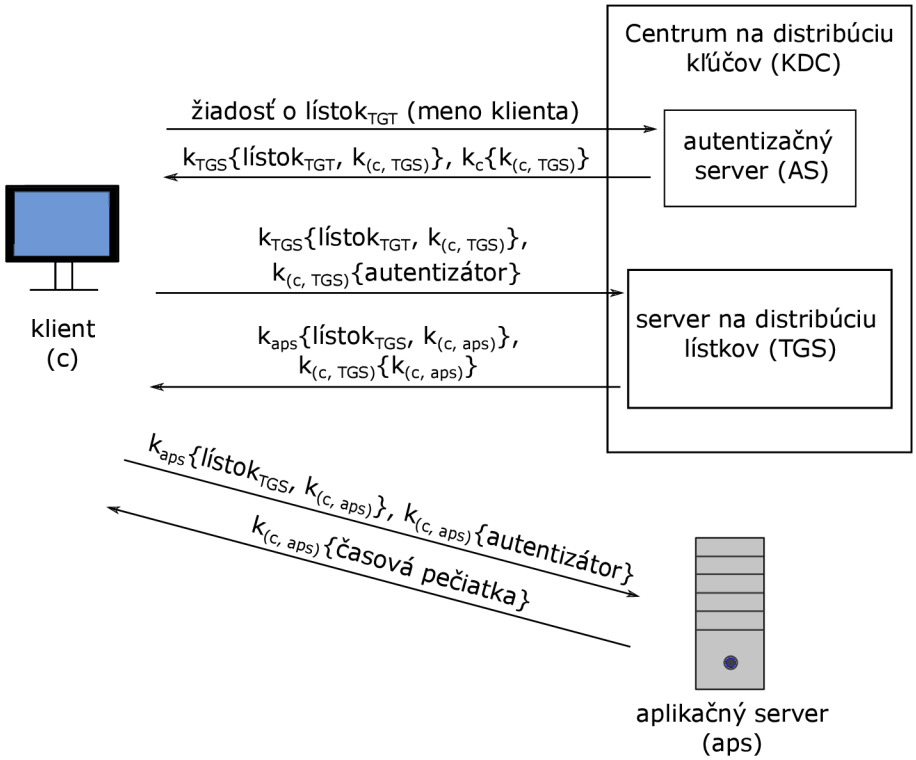
Prístup k sieťovej službe prebieha v 3 fázach:

- Získanie lístka tzv. *ticket granting ticket* (TGT), ktorý preukazuje identitu klienta serveru na distribúciu lístkov, tzv. *ticket granting server*.
- Získanie lístka tzv. *ticket granting service* (TGS), ktorý preukazuje identitu klienta aplikačnému serveru.
- Prezentácia lístka aplikačnému serveru a následne overenie identity tohoto servera.

3.1.1 Získanie lístka TGT

Pre získanie lístku TGT, ktorý umožní klientovi získať ďalšie lístky pre konkrétne služby, posiela klient v správe `KRB_AS_REQ` žiadosť KDC, konkrétne autentizačnému serveru. V tejto správe je uvedené užívateľské meno v nešifrovanej podobe, pomocou ktorého autentizačný server skontroluje, či sa daný klient nachádza v jeho databáze. Pokiaľ áno, server pošle v správe `KRB_AS_REP` klientovi kľúč, ktorý bude používať pri komunikácii so serverom na distribúciu lístkov. Tento kľúč je zašifrovaný klientskym kľúčom, takže iba užívateľ, ktorý zadal správne heslo, ho bude schopný dešifrovať. V správe sa ešte nachádza lístok TGT, ktorý je zašifrovaný tajným kľúčom servera na distribúciu lístkov.

Pokiaľ KDC vyžaduje tzv. predautentizáciu, klientovi nestačí poslať autentizačnému serveru iba užívateľské meno, ale je po ňom vyžadovaná aj časová pečiatka zašifrovaná klientskym kľúčom. Autentizačný server až po úspešnom dešifrovaní správy posiela klientovi lístok TGT. Tento spôsob zvyšuje bezpečnosť proti útokom hrubou silou, kedy si môže útočník bez použitej predautentizácie vyžiadať lístok zašifrovaným slabším šifrovacím algoritmom a ten sa pokúsiť dešifrovať.



Obr. 3.1: Priebeh komunikácie protokolu Kerberos

3.1.2 Získanie lístka TGS

Po získaní lístku TGT si môže klient vyžiadať od servera na distribúciu lístkov TGS lístok, ktorý použije pre preukázanie svojej identity aplikačnému serveru. Klient posiela serveru na distribúciu lístkov žiadosť o konkrétnu službu v správe `KRB_TGS_REQ`, ktorá obsahuje lístok TGT a tzv. autentizátor. Tento autentizátor obsahuje meno klienta, časovú pečiatku a je zašifrovaný spoločným kľúčom klienta a servera na distribúciu lístkov. V odpovedi `KRB_TGS_REP` sa nachádza lístok TGS spolu s kľúčom pre komunikáciu klienta s aplikačným serverom, zašifrovaný kľúčom aplikačného servera. Spolu s týmito dátami správa obsahuje ešte spoločný kľúč klienta a aplikačného servera, zašifrovaný spoločným kľúčom klienta a servera na distribúciu lístkov.

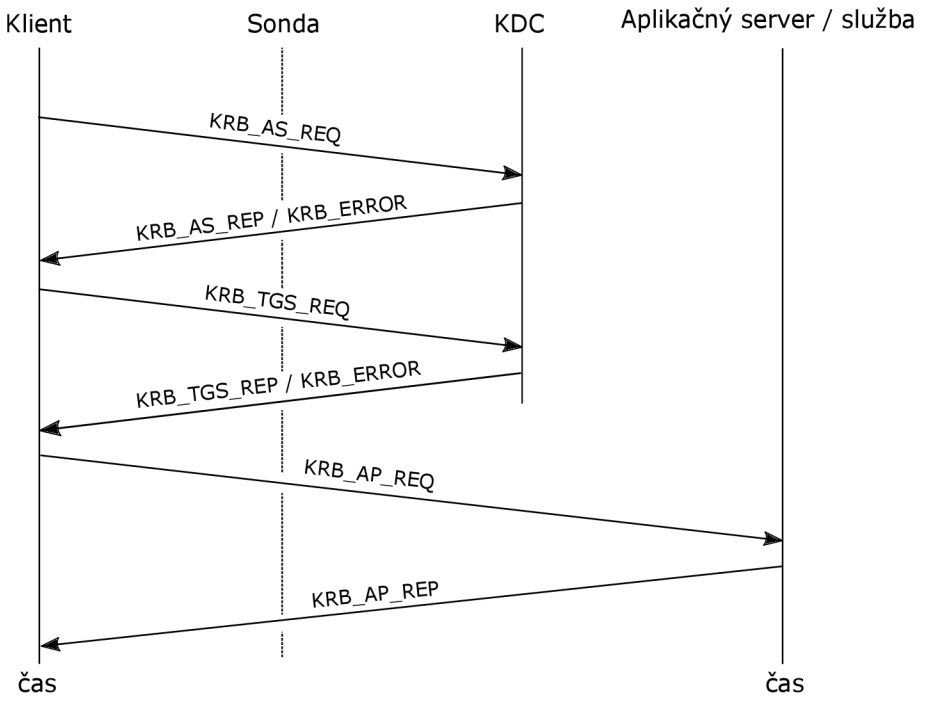
3.1.3 Žiadost o službu

V tejto fáze už klient môže poslať žiadosť o službu v správe KRB_AP_REQ aplikačnému serveru. Správa obsahuje lístok TGS, spoločný kľúč klienta a aplikačného servera a je to zašifrované kľúčom aplikačného serveru. V správe sa nachádza ešte autentizátor rovnaký ako posielal klient serveru na distribúciu lístkov, ale tentokrát je zašifrovaný spoločným kľúčom klienta a aplikačného servera. Pokiaľ sa serveru podarí autentizátor úspešne dešifrovať, posiela klientovi správu KRB_AP_REP, ktorá je potvrdením úspešnej autentizácie klienta a obsahuje aj časovú pečiatku z autentizátora zvýšenú o hodnotu 1 zašifrovanú spoločným kľúčom klienta a aplikačného servera. Pomocou tejto časovej pečiatky klient overí identitu aplikačného servera.

3.2 Nešifrovaný obsah správ

Okrem informácií, ktoré sa posielajú šifrovane, obsahujú správy protokolu Kerberos aj nešifrované dáta. Pomocou týchto údajov je možné rozšíriť IP toky a následne s nimi ďalej pracovať, napríklad ich štatisticky spracovávať, vyhodnocovať bezpečnostné riziká a pod. Obrázok 3.2 zobrazuje schému monitorovania správ protokolu Kerberos pomocou FlowMon sondy.

- PVNO - Pole označujúce verziu protokolu Kerberos. Nachádza sa v každej správe.
- TKT-VNO - Špecifikuje číslo verzie pre formát lístka. Nachádza sa v každej správe, ktorá obsahuje lístok.
- MSG-TYPE - Pole určujúce typ správy. Možné typy správ sú KRB_AS_REQ, KRB_AS_REP, KRB_TGS_REQ, KRB_TGS_REP, KRB_AP_REQ, KRB_AP_REP, KRB_SAFE, KRB_PRIV, KRB_CRED a KRB_ERROR.
- CNAME - Pole určujúce meno klienta. Nachádza sa v správach KRB_AS_REQ, KRB_AS_REP a KRB_TGS_REP. V správe typu KRB_ERROR je pole voliteľné.
- CREALM - Názov logickej oblasti, v ktorej je klient zaregistrovaný a prebehlo v nej prvotné overenie identity klienta. Nachádza sa v KRB_AS_REP, KRB_TGS_REP a v KRB_ERROR je voliteľné.



Obr. 3.2: Monitorovanie správ protokolu Kerberos

- `REALM` - Názov logickej oblasti, v ktorej sa nachádza server. Pole obsahujú správy `KRB_AS_REQ`, `KRB_TGS_REQ` a `KRB_ERROR`.
- `SNAME` - Určuje názov servera vrátane názvu služby. Nachádza sa v každom lístku a správe `KRB_ERROR`, v správach `KRB_AS_REQ` a `KRB_TGS_REQ` je pole voliteľné.
- `FROM` - Pole označujúce čas, odkedy má byť vyžiadaný lístok platný. Nachádza sa v správach `KRB_AS_REQ` a `KRB_TGS_REQ` voliteľne. Pokiaľ sa pole v správe nenachádza, KDC použije aktuálny čas ako začiatok platnosti lístka.
- `TILL` - Obsahuje čas, dokedy klient vyžaduje, aby bol lístok platný. Pokiaľ je hodnota časovej pečiatky `19700101000000Z`, je požadovaná maximálna doba platnosti lístka, akú KDC umožňuje. Nachádza sa v správach `KRB_AS_REQ` a `KRB_TGS_REQ`.
- `RTIME` - Čas, dokedy je možné daný lístok obnoviť. Pole sa voli-

teľne nachádza v správach `KRB_AS_REQ` a `KRB_TGS_REQ`.

- `STIME` - Pole obsahujúce aktuálny čas servera. Nachádza sa iba v správe `KRB_ERROR`.
- `SUSEC` - Pole vyjadruje počet milisekúnd. Môže mať hodnotu od `0` do `999999`. Nachádza sa v správe `KRB_ERROR` popri poli `STIME` na zvýšenie presnosti časovej pečiatky.

4 Programové vybavenie exportéra

V rámci praktickej časti bakalárskej práce bol vytvorený zásuvný modul pre FlowMon exportér, ktorý spracováva správy protokolu Kerberos. FlowMon exportér je samostatné zariadenie, ktoré pasívne monitoruje sieťovú prevádzku a vytvára z nej štatistiky. Tieto štatistiky sú exportované vo forme NetFlow v5/v9 alebo IPFIX na kolektor, ktorý ich umožňuje zobraziť užívateľom napríklad pomocou webového rozhrania.

4.1 FlowMon exportér 4.0

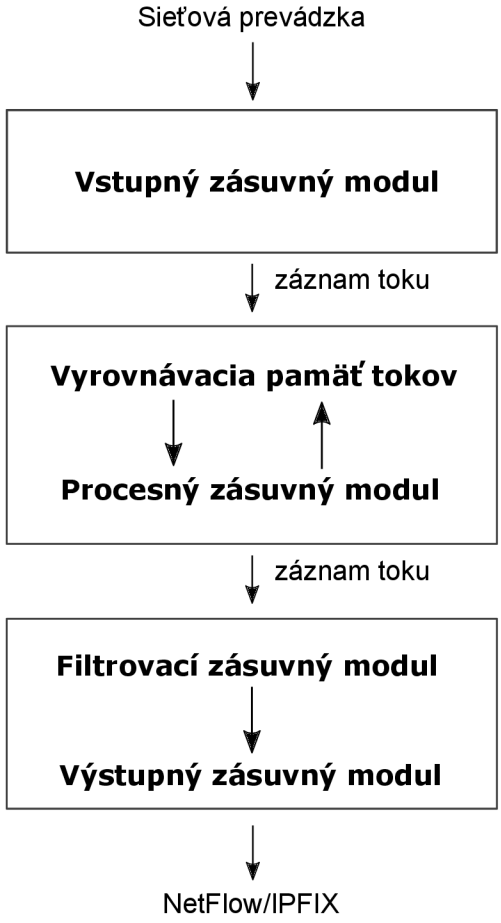
FlowMon exportér (sonda) sa skladá z 3 hlavných častí: vstup, pamäť(*flow record*) a výstup. Zjednodušená architektúra FlowMon exportéra je zobrazená na obrázku 4.1.

Exportér dokáže prijímať na vstupe pakety z vysoko rýchlostných sietí (až 100 Gb/s), ktoré zoskupuje do sieťových tokov a vytvára záznamy tokov. Záznam toku môže obsahovať rozličné informácie zo sieťovej prevádzky. Formát záznamu toku protokolu NetFlow v5, ktorý je predchodcom protokolov NetFlow v9 a IPFIX, obsahuje nasledujúce údaje:

- počet paketov,
- počet bajtov,
- zdrojovú a cieľovú IP adresu,
- zdrojový a cieľový port,
- začiatok a koniec toku,
- transportný protokol,
- SNMP index vstupného a výstupného rozhrania,
- typ služby (ToS),
- príznaky protokolu TCP,
- adresa nasledujúceho uzlu,

- číslo autonómneho systému zdroja a cieľa,
- maska zdrojovej a cieľovej IP adresy.

Formát správ protokolov NetFlow v9 a IPFIX je flexibilný. NetFlow v9 definuje 104 políčok, ktoré môže záznam toku obsahovať. IPFIX ich definuje až 433 a navyše umožňuje definovať nové vlastné políčka v rámci organizácie.



Obr. 4.1: Zjednodušená architektúra FlowMon exportéra [20]

Po spracovaní sieťovej prevádzky vstupným zásuvným modulom sú záznamy tokov ukladané do pamäte (*flow cache*). Pokiaľ sa v nej nachádza tok s rovnakými kľúčovými parametrami, tj. *zdrojová a cieľová*

IP adresa, zdrojový a cieľový port a transportný protokol, sú tieto záznamy zlúčené do jedného. Záznamy tokov sa vo *flow cache* pravidelne prechádzajú a kontrolujú sa ich aktívne a neaktívne časové limity. Pokiaľ je jeden z týchto limitov prekročený, záznam je uvoľnený z pamäte a odoslaný výstupnému zásuvnému modulu. Pred exportom môže záznam toku voliteľne prejsť filtrom, ktorý rozhodne, či bude daný tok exportovaný, napríklad na základe IP adresy alebo portu.

Výstupný zásuvný modul slúži na export záznamov tokov, ktoré posiela vo forme NetFlow alebo IPFIX správ na kolektor. Ďalšími možnými formátmi výstupu sú napríklad hodnoty oddelené čiarkami (`csv`), SQL databázy a podobne.

4.1.1 Základné použitie exportéra

FlowMon exportér pracuje pomocou zásuvných modulov, ktoré rozširujú jeho funkcionalitu. Zoznam všetkých dostupných zásuvných modulov je možné zobraziť pomocou príkazu `flowmonexp -l`. Opis parametrov exportéra a použitie je možné zobraziť pomocou prepínaču `-h`.

Na spustenie exportéra je potrebný práve jeden vstupný zásuvný modul a aspoň jeden výstupný, ktorých počet nie je obmedzený. Vstupný zásuvný modul môže prijímať dáta z PCAP súborov alebo zo sieťového rozhrania, kedy je nutné spustiť exportér pod užívateľom root, aby mohol pristupovať k požadovanému fyzickému sieťovému rozhraniu. Na definíciu vstupného zásuvného modulu slúži prepínač `-I` a na definovanie typu výstupu pomocou výstupného zásuvného modulu je používaný prepínač `-E`.

Základné spustenie exportéra môže vyzerať následne:

```
# flowmonexp -Ipcap:device=eth0 -E csv
```

Výpis kódu 4.1: Spustenie FlowMon exportéra so základnými parametrami

V uvedenom príklade je exportér spustený pod užívateľom root aby mohol prijímať dáta z fyzického sieťového rozhrania `eth0` a následne ich exportuje vo formáte `csv`.

Komplexnejší príklad spustenia exportéra, ktorý využíva rozširujúci zásuvný modul, môže vyzerať takto:

```
# flowmonexp \
   -I pcap-replay:file=/pcaps/kerberos.pcap \
   -X process_kerberos/flowmon_process_kerberos.so  \
   -P process-kerberos \
   -E ipfix-ng:port=4739,protocol=tcp,host=localhost,\
      template-file=/conf/ipfix-ng_template.txt,\
      fields-file=/conf/ipfix-ng_fields.txt
```

Výpis kódu 4.2: Zložitejší príklad spustenia FlowMon exportéra

Vstupný zásuvný modul prijíma dáta z PCAP súboru, ktoré následne spracuje procesný zásuvný modul `flowmon_process_kerberos`. Výstupný zásuvný modul dáta exportuje vo forme IPFIX správ protokolom TCP na port 4739.

4.1.2 Rozširujúce zásuvné moduly

Rozširujúce zásuvné moduly implementujú ďalšie funkcie exportéra. Poznáme 5 druhov zásuvných modulov:

- **Input** - zabezpečuje príjem paketov zo vstupu.
- **Reflector** - prijíma dáta od vstupného zásuvného modulu, výstup z neho ďalej nie je použitý v exportéri. Môže byť použitý na ukladanie alebo spracovanie vybraných paketov.
- **Process** - dodatočne spracováva pakety.
- **Filter** - filtruje záznamy tokov.
- **Export** - exportuje toky v rôznych formátoch.

Rozširujúce zásuvné moduly sú dynamicky načítané FlowMon exportérom v podobe zdieľaných objektov (`.so`), ktoré musia byť uložené v zložke `/usr/lib64/flowmonexp`. K načítaniu zásuvného modulu je možné použiť aj prepínač `-X` nasledovaný cestou k zdieľanému objektu daného zásuvného modulu, ako je uvedené vo výpise 4.2. Tým sa zásuvný modul zobrazí v dostupných moduloch, ktoré je možné použiť pomocou prepínaču `-P`. Načítanie zásuvného modulu je možné maximálne jedenkrát, pretože použitie jedného zásuvného modulu je kontrolované na základe jeho názvu a nie názvu binárneho súboru.

4.1.3 Aplikačné rozhranie zásuvných modulov

Na vývoj nových rozširujúcich zásuvných modulov je dostupné aplikačné rozhranie exportéra v podobe hlavičkových súborov, ktoré sa nachádzajú v zložke `/usr/include/flowmonexp`. V hlavičkovom súbore `record.h` sa nachádza definícia záznamu toku (`flow_record_t`) a definície funkcií na prácu s ňou. Hlavičkový súbor `plugin.h` obsahuje definíciu jednotlivých zásuvných modulov a konfiguračné štruktúry. Pre každý typ zásuvného modulu existuje hlavičkový súbor, napríklad pre procesný zásuvný modul to je `plugin_process.h`, ktorý obsahuje definície funkcií, ktoré musia byť pri tvorbe zásuvného modulu definované.

Každý zásuvný modul musí obsahovať definíciu svojho typu, ktorý sa určuje pomocou makra `SET_PLUGIN_TYPE` a aktuálnu verziu aplikačného rozhrania určenou makrom `FLOWMON_API_VERSION`. Ďalej každý modul obsahuje 3 základné funkcie: `desc()`, `init()` a `shutdown()`.

Funkcia `desc()` je volaná raz pri načítaní modulu do exportéra a vracia základné informácie o zásuvnom module, ktoré sú uložené v štruktúre `plugin_desc_t` (výpis 4.3). Premenná `name` označuje názov a premenná `help` obsahuje opis zásuvného modulu. Obe hodnoty sa vypíšu pri zobrazení pomocníka FlowMon exportéra pomocou prepínaču `-l`. Položka `data_size` označuje veľkosť pamäte, ktorú zásuvný modul vyžaduje v zázname toku na uloženie dát. Položka `flags` môže obsahovať rôzne príznaky, ktorých výpis je v hlavičkovom súbore `plugin.h`.

```
typedef struct {
        char *name;
        char *help;
        uint16_t data_size;
        plugin_flags_t flags;
} plugin_desc_t;
```

Výpis kódu 4.3: Výpis štruktúry `plugin_desc_t`

Funkcia `init()` je volaná vždy pri spustení zásuvného modulu a slúži na inicializáciu štruktúr a vykonanie operácií, ktoré musia byť vykonané raz pri spustení modulu. Hlavička tejto funkcie je pre každý typ zásuvného modulu iná, jej predpis je dostupný v hlavičkových súboroch jednotlivých typov zásuvných modulov. Funkcia `init()` vracia ukazovateľ na privátnu pamäť, ktorý je potom dostupný aj v ostatných funkciách zásuvného modulu. Pokiaľ sa nepodarí inicializovať privátnu pa-

mäť, prípadne dôjde k inej chybe, funckia `init()` vráti hodnotu `NULL`.

Funkcia `shutdown()` je volaná tesne pred skončením práce exportéra. Má jediný argument ukazovateľ na privátnu pamäť, ktorú vrátila funkcia `init()`. Funkcia `shutdown()` sa používa na uvoľnenie alokovanej pamäte, uzavrenie spojení, výpis štatistík a podobne.

4.2 Procesný zásuvný modul process-kerberos

Hlavným prínosom tejto bakalárskej práce je procesný zásuvný modul `process-kerberos`. Jeho úlohou je rozpoznávať správy protokolu Kerberos, spracovávať ich a rozšíriť získanými údajmi záznamy tokov. Tie sú následne exportované pomocou exportného zásuvného modulu `ipfix-ng`, ktorý dané záznamy posiela protokolom IPFIX na kolektor. Z dôvodu nemožnosti rozšíriť protokol NetFlow o vlastné údaje z protokolu Kerberos je NetFlow na export záznamov tokov s novými poľami nepoužiteľný. Na výpis jednotlivých polí, ktoré boli získané pomocou zásuvného modulu `process-kerberos`, je možné použiť aj výstup vo formáte `csv`, ktorý je vhodný najmä pre testovacie účely.

Zdrojový kód procesného zásuvného modulu je rozdelená do 2 súborov: `kerberos.c` a `kerberos.h`, ktoré sa nachádzajú v prílohe v zložke `src/process-kerberos`. Zdrojový súbor `kerberos.c` obsahuje implementáciu všetkých povinných funkcií pre procesný zásuvný modul a taktiež funkcie, ktoré sa starajú o spracovanie správ protokolu Kerberos a rozširovanie záznamov tokov o získané údaje. Hlavičkový súbor `kerberos.h` obsahuje predpis hlavičiek daných funkcií a taktiež štruktúru `kerberos_record_t`, do ktorej sa ukladajú získané dáta. V zložke `src/process-kerberos` sa nachádza aj `Makefile`, pomocou ktorého sa zdrojový kód preloží a vytvorí zdieľaný objekt s názvom `flowmon_process_kerberos.so`. Tento objekt je potom možné použiť na spustenie FlowMon exportéra s prepínačom `-X` s uvedenou cestou k danému objektu.

Procesný zásuvný modul získava nešifrované údaje zo správ protokolu Kerberos a ukladá ich do polí, ktoré sa nazývajú *IPFIX Information Elements* [12]. Celý výpis exportovaných polí je zobrazený v tabuľke 4.1 a ich podrobnejší význam je opísaný v sekcii 3.2. Všetky elementy majú identifikátor *enterprise number* s hodnotou 16982, ktorý je priradený Masarykovej univerzite na definíciu vlastných polí.

Názov elementu	ID	Popis
KRB_MSG_TYPE	1200	Typ správy
KRB_ETYPE	1201	Použitý šifrovací algoritmus
KRB_PVNO	1202	Verzia protokolu Kerberos
KRB_ERR_CODE	1203	Kód chybovej správy
KRB_FROM	1204	Čas odkedy je lístok platný
KRB_TILL	1205	Čas dokedy je lístok platný
KRB_RTIME	1206	Čas dokedy je možné lístok obnoviť
KRB_CNAME	1207	Meno klienta
KRB_CREALM	1208	Názov oblasti, do ktorej patrí klient
KRB_REALM	1209	Názov oblasti, do ktorej patrí server
KRB_SVC_NAME	1210	Názov služby
KRB_SRV_NAME	1211	Meno servera
KRB_TKT_VNO	1212	Verzia lístka
KRB_TKT_REALM	1213	Názov oblasti, do ktorej server patrí - nachádza sa v lístku
KRB_TKT_SRV_NAME	1214	Meno servera - nachádza sa v lístku
KRB_TKT_SVC_NAME	1215	Názov služby - nachádza sa v lístku
KRB_STIME	1216	Čas servera
KRB_SUSEC	1217	Mikrosekundy servera
KRB_AUTH_ETYPE	1218	Použitý algoritmus na zašifrovanie autentizátora
KRB_TKT_ETYPE	1219	Použitý algoritmus na zašifrovanie lístka

Tabuľka 4.1: Polia exportované zásuvným modulom `process-kerberos`

4.3 Implementácia zásuvného modulu

Zásuvný modul `process-kerberos` okrem základných funkcií (`init()`, `desc()`, `shutdown()`) implementuje funkciu `kerberos_parser()`, ktorá sa stará o získavanie nešifrovaných údajov zo správ protokolu Kerberos

a ďalej funkcie, ktoré súvisia s naplňovaním polí protokolu IPFIX.

4.3.1 Fukcia init()

Funkcia `init()` je volaná raz pri načítaní zásuvného modulu a obsahuje 3 parametre. Jej hlavička je zobrazená vo výpise 4.4. Prvý parameter `params` obsahuje parametre predané z príkazového riadka. `Getter_list` je ukazovateľ na zoznam getterov, čo sú funkcie, ktoré vracajú hodnoty exportovaných polí. Posledný parameter `data_offset` je hodnota, ktorá reprezentuje pozíciu privátnych dát zásuvného modulu v zázname toku.

```
void *plugin_process_init(char *params,
                    flow_record_getter_t **getter_list,
                    int data_offset)
```

Výpis kódu 4.4: Výpis hlavičky funckie `init()`

Vo funkcii `init()` prebieha najprv alokácia privátnej pamäte, ktorá v prípade neúspechu spôsobí predčasné ukončenie behu zásuvného modulu a funkcia `init()` vráti hodnotu `NULL`. Po alokovaní pamäte je potrebné registrovať parser pomocou funkcie `register_parser_post()`, ktorý sa stará o získavanie dát zo správ protokolu Kerberos. Predpis tejto funkcie je možné dohľadať v hlavičkovom súbore `parser.h` Za ňou nasleduje pridanie getterov, ktoré vracajú hodnoty exportovaných polí z tabuľky 4.1.

4.3.2 Spracovávanie správ protokolu Kerberos

O spracovávanie správ protokolu Kerberos a získavanie nešifrovaných dát z nich sa stará funkcia `krb_parser()`, ktorej hlavička je zobrazená vo výpise 4.5. Parameter `plugin_private` je ukazovateľ na privátnu pamäť a `packet` je ukazovateľ na štruktúru `flow_packet_t`, ktorá sprostredkováva prístup k paketu zachyteného na sieťovom rozhraní.

```
int krb_parser(void *plugin_private,
         flow_packet_t *packet)
```

Výpis kódu 4.5: Výpis hlavičky funckie `krb_parser()`

Vo funkcii `krb_parser()` sa analyzuje každý paket s cieľom identifikovať správy protokolu Kerberos. Tie sú posielané transportnými pro-

tokolmi TCP alebo UDP a číslo zdrojového alebo cieľového portu je 88. Pokiaľ paket nesplňuje túto podmienku, funkcia vracia hodnotu `PARSER_RETURN` a spracovávanie ďalej nepokračuje. Pri úspešnom rozpoznaní správy protokolu Kerberos je podľa typu správy spracovaný príslušnou funkciou, ktorých výpis je v tabuľke 4.2. V prípade, že v daných funkciách nedôjde k chybe, vracia funkcia `krb_parser()` hodnotu `PARSER_NEXT`.

Funkcia	**Bajt**
parse_as_req()	0x0a
parse_as_rep()	0x0b
parse_tgs_req()	0x0c
parse_tgs_rep()	0x0d
parse_krb_err()	0x1e

Tabuľka 4.2: Funkcie, ktoré analyzujú správy protokolu Kerberos

5 Výsledky meraní

Navrhnutý zásuvný modul `process-kerberos` bol otestovaný na výkon a taktiež bola otestovaná jeho funkčnosť na reálnej sieti Masarykovej univerzity. V tejto kapitole je opísaný spôsob merania zásuvného modulu a demonštrované použitie na príkladoch sieťovej prevádzky, ktorá obsahuje komunikáciu s protokolom Kerberos.

5.1 Meranie výkonu

Hlavným cieľom merania bolo zistiť, k akej zmene výkonu dôjde pri použití procesného zásuvného modulu. Testy boli zamerané na priepustnosť paketov a zaťaženie exportéra pri spracovávaní sieťovej prevádzky. Meranie bolo vykonané na stroji Intel(R) Xeon(R) CPU E31230 @ 3.20GHz so 16GB pamäte.

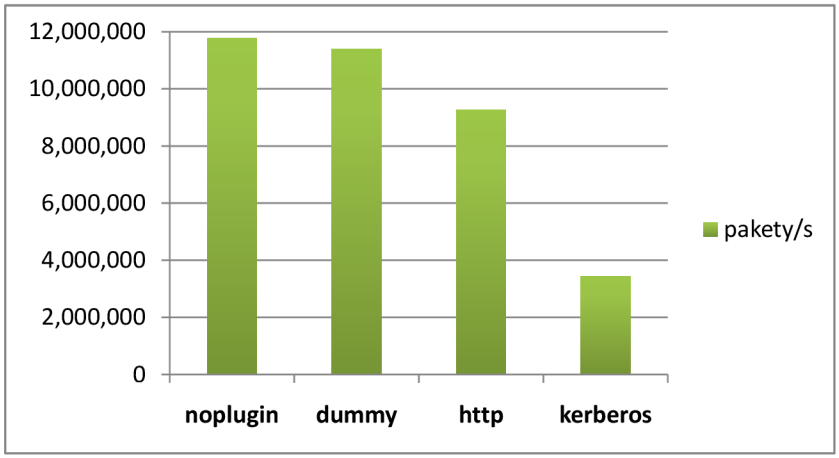
Na testovanie priepustnosti paketov bol využitý skript `benchmark` od CESNETu , ktorý sa nachádza v prílohe v zložke `/src/benchmark`. Ukážka z jeho použitia sa nachádza vo výpise 5.1.

```
flowmonexp -v0 -t 300:30 -q 22
   -I benchmark:file=/pcaps/traffic.pcap,pktsize=1500
   -P process-kerberos -E null
```

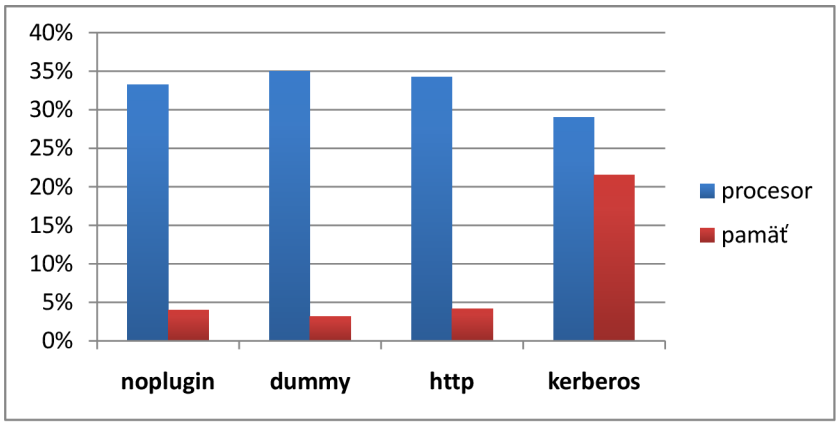
Výpis kódu 5.1: Ukážka testovania modulu `process-kerberos`

Meranie bolo vykonané na PCAP súbore, ktorý obsahoval rôznu sieťovú prevádzku vrátane protokolu Kerberos. Postupne boli testované 4 rozličné situácie, pri ktorých boli využité procesné moduly `dummy`, `http` a `process-kerberos`. Na grafe 5.1 je zobrazená priepustnosť jednotlivých zásuvných modulov. V prvom prípade bol testovaný výkon exportéra bez zásuvného modulu, ktorý je označený ako `noplugin`. Procesný zásuvný modul `dummy` je modul, ktorý nevykonáva žiadne operácie, preto predstavuje najvyššiu možnú rýchlosť spracovávania paketov procesnými zásuvnými modulmi. Ďalej bol otestovaný výkon procesného modulu `http` a nakoniec `process-kerberos`. Výkon procesného modulu, ktorý spracováva správy protokolu Kerberos, je zhruba 1/3 výkonu modulu `dummy`, čo je spôsobené spracovávaním veľkého množstva textových reťazcov.

Druhá časť merania sa zaoberala hardvérovými nárokmi procesného zásuvného modulu. Tak ako aj v predchádzajúcom teste boli monitorované 4 situácie s rôznymi modulmi. Využitie procesoru modulom `process-kerberos` bolo mierne nižšie ako pri ostatných meraniach, no využitie operačnej pamäte bola výrazne vyššia. To je spôsobené veľkosťou privátnej štruktúry modulu, do ktorej sa ukladá viacero textových polí získaných zo správ protokolu Kerberos.



Obr. 5.1: Porovnanie výkonu s inými modulmi



Obr. 5.2: Hardvérové využitie procesnými modulmi

5.2 Overenie funkčnosti modulu

Procesný zásuvný modul `process-kerberos` bol otestovaný na overenie funkčnosti na reálnej sieti a taktiež na PCAP súboroch, ktoré obsahovali už existujúcu sieťovú prevádzku, z ktorých sú zobrazené namerané výsledky v tejto časti práce.

Sieťová prevádzka bola spracovávaná nástrojom *flowmonexp*, ktorý z nej vytváral záznamy tokov, ktoré boli posielané na kolektor (nástroj *ipfixcol*). Získané dáta je možné analyzovať pomocou nástroja *fbitdump*, ktorý umožňuje napríklad zoraďovať záznamy podľa určitých parametrov, vytvárať štatistiky a podobne. Každý z týchto nástrojov potrebuje pre správne fungovanie konfiguračné súbory, ktoré sú priložené v prílohe v zložke `/conf`. Pre nástroj *fbitdump* boli vytvorené moduly na preklad hodnôt na ľudsky čitateľnú podobu, ktoré sa nachádzajú v zložke `/src/fbitdump`.

V nasledujúcej časti je ukázaná práca s nástrojmi `flowmonexp`, `ipfixcol` a `fbitdump` a výpis získaných dát zo sieťovej prevádzky. Najprv je potrebné spustiť nástroj `ipfixcol` s príslušnými konfiguračnými súbormi (výpis 5.2):

```
ipfixcol -v 1
    -c /conf/ipfixcol/startup-tcp-4739.xml
    -e  /conf/ipfixcol/ipfix-elements.xml
```

Výpis kódu 5.2: Spustenie nástroja `ipfixcol`

Následne je spustený FlowMon exportér, ktorý načíta dáta z PCAP súboru `krb-816.cap`, čo je vzorová sieťová prevádzka protokolu Kerberos poskytovaná na stránkach Wiresharku. Exportér dáta spracuje a posiela ich na kolektor vo forme IPFIX správ (výpis 5.3).

```
flowmonexp \
    -I pcap-replay:file=/pcaps/krb-816.cap \
    -X process_kerberos/flowmon_process_kerberos.so  \
    -P process-kerberos \
    -E ipfix-ng:port=4739,protocol=tcp,host=localhost,\
       template-file=/conf/ipfix-ng_template.txt,\
       fields-file=/conf/ipfix-ng_fields.txt
```

Výpis kódu 5.3: Spustenie exportéra `flowmonexp`

Získane dáta je možne jednoducho vypísať nástrojom `fbitdump`

s príslušným konfiguračným súborom, cestou k uloženým dátam a formátom výstupu (výpis 5.4). Samotný výstup je zobrazený vo výpise 5.5 a 5.6. Z dôvodu veľkého rozsahu záznamu sú zobrazované len časti výstupu.

```
fbitdump
    -C /conf/fbitdump/fbitdump.xml
    -R /tcp4739/03/22/ic20150322214000/
    -o "fmt:%msgtype %err %stime %susec %from %till \
            %rtime %cname %crealm %realm %svcname \
            %srvname %tktrealm %tktsrvname %tktsvcname"
```

Výpis kódu 5.4: Spustenie nástroja `fbitdump`

```
   Message  Cname     Svc name          Tkt srv name
 KRB_AS_REQ   des       krbtgt                   ---
KRB_TGS_REQ   ---         LDAP pc-w2k3ent.denyDC.com
KRB_TGS_REP  xp1$          ---            VPC-W2K3ENT
  KRB_ERROR   ---       krbtgt                   ---
 KRB_AS_REP    u5          ---             DENYDC.COM
```

Výpis kódu 5.5: Časť polí z PCAP súboru `krb-816.cap`

```
              Till time           Realm   Tkt etype
2037-09-13 04:48:05.000          DENYDC         ---
2037-09-13 04:48:05.000      DENYDC.COM    rc4-hmac
                   ---          DENYDC         ---
```

Výpis kódu 5.6: Časť polí z PCAP súboru `krb-816.cap`

Ďalší PCAP súbor použitý pri testovaní procesného zásuvného modulu je tvorený zo simulovanej prevádzky s komunikáciou klienta s operačným systémom Microsoft Windows 7 a servera s operačným systémom Microsoft Windows Server 2008. Výstupy niektorých políčok sa nachádzajú vo výpisoch 5.7 a 5.8.

```
   Message    Cname     Svc name            Tkt srv name
 KRB_AS_REQ klient$       krbtgt                     ---
KRB_TGS_REQ     ---         host         AD.SERVER2008.CZ
KRB_TGS_REP   matus          --- server.ad.server2008.cz
  KRB_ERROR    ---       krbtgt                     ---
 KRB_AS_REP  KLIENT          ---         AD.SERVER2008.CZ
```

Výpis kódu 5.7: Časť polí z PCAP súboru `krb-sim.cap`

```
              Till time            Realm     Tkt etype
2021-08-13 18:00:15.000 AD.SERVER2008.CZ   aes256-cts-
                                          hmac-sha1-96
2037-09-13 04:48:05.000               AD   aes256-cts-
                                          hmac-sha1-96
```

Výpis kódu 5.8: Časť polí z PCAP súboru `krb-sim.cap`

Vo výpise 5.9 sú zobrazené informácie z chybových správ oboch PCAP súborov.

```
      Err code             Server time     Susec
KRB5KDC_ERR_
PREAUTH_FAILED 2015-05-11 02:28:15.000    175263
KRB5KDC_ERR_
ETYPE_NOSUPP   2005-08-16 11:40:29.000    534652
```

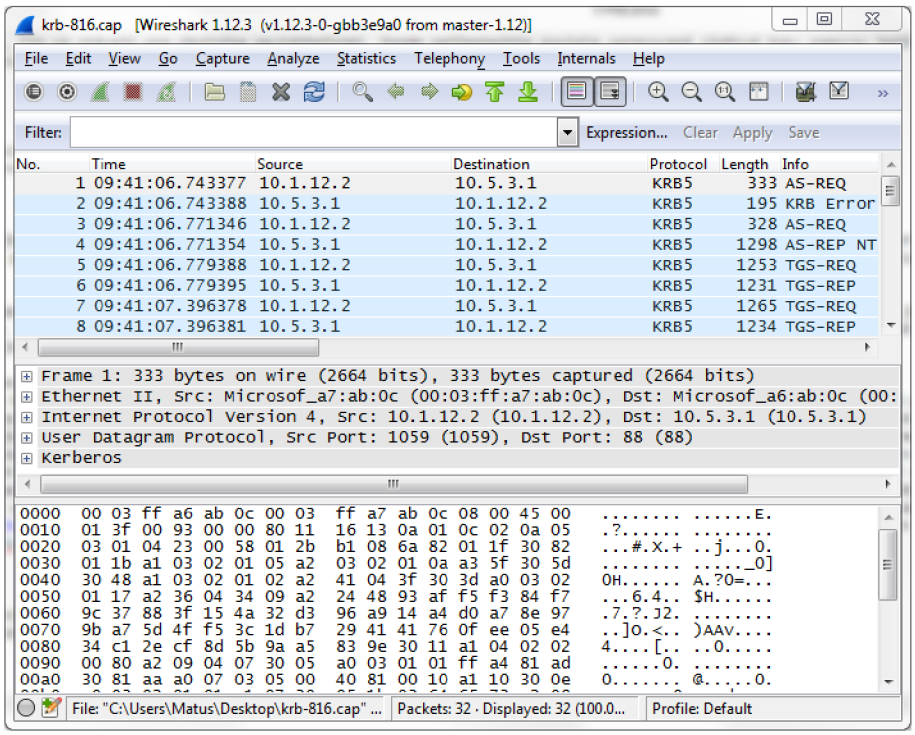
Výpis kódu 5.9: Časť polí z PCAP súboru `krb-sim.cap`

6 Ďalšie možnosti monitorovania protokolu Kerberos

Na monitorovanie protokolu Kerberos existuje niekoľko nástrojov, ktoré fungujú na princípe analýzy paketov. Tieto nástroje sú jednoduché na použitie a sú vhodné na pozorovanie a odhaľovanie problémov na menších sieťach. Komunikáciu protokolu Kerberos je taktiež možné monitorovať pomocou logovania.

6.1 Analyzátory paketov

Medzi najznámejšie nástroje fungujúce na princípe analýzy paketov (*packet analyzers*) patrí **Wireshark**, ktorý bol použitý pri tvorbe tejto



Obr. 6.1: Ukážka grafického rozhrania programu Wireshark

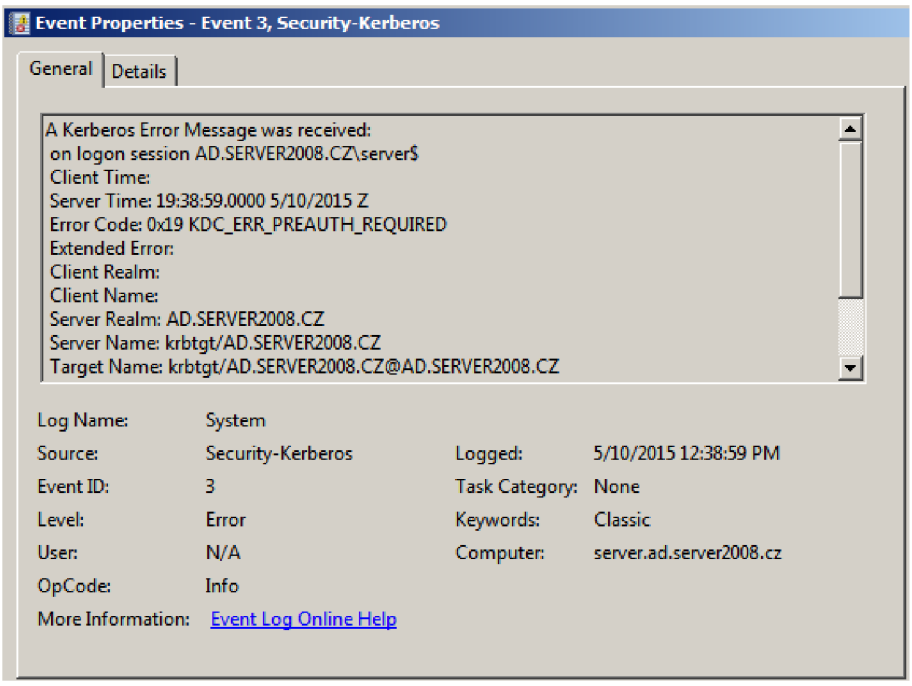
práce na zoznámenie sa s priebehom komunikácie zúčastnených strán

pomocou protokolu Kerberos. Wireshark umožňuje odchytávať sieťovú prevádzku zo sieťového rozhrania alebo ju nahrať z dopredu zachytených paketov vo forme PCAP súborov. Zachytená prevádzka sa dá jednoducho analyzovať pomocou grafického užívateľského rozhrania, ktoré poskytuje nástroje na filtrovanie komunikácie rôznymi parametrami ako sú napríklad IP adresa, port, protokol a podobne. Wireshark taktiež dokáže syntakticky analyzovať obsah paketu a zobraziť význam jednotlivých políčok. Je dostupný pre operačné systémy Windows a väčšinu operačných systémov založených na UNIXe. Na obrázku 6.1 je zobrazené grafické rozhranie programu Wireshark so sieťovou prevádzkou protokolu Kerberos dostupnou na stránkach vývojára tohto nástroja [14]. Na rovnakom princípe ako Wireshark funguje napríklad program **Microsoft Network Monitor** [15], ktorý je dostupný pre operačné systémy Windows.

Okrem programov, ktoré využívajú grafické užívateľské rozhranie, existujú nástroje na monitorovanie sieťovej prevádzky z príkazového riadka. Ich hlavnou nevýhodou je, že nedokážu zobraziť význam jednotlivých bajtov v pakete. Hlavným reprezentantom je nástroj **tcpdump** [16] pre operačné systémy založené na UNIXe a jeho ekvivalent pre operačný systém Windows **WinDump** [17].

6.2 Monitorovanie pomocou logu

Monitorovanie pomocou logu predstavuje jednoduché riešenie, ktoré nevyžaduje žiadny špeciálny nástroj, ktorý by sa o danú úlohu staral. Servery, ktoré využívajú operačné systémy *Microsoft Windows Server* [18], umožňujú monitorovať protokol Kerberos pomocou logovacieho mechanizmu **Kerberos Event Logging** [19]. Pokiaľ je táto funkcia povolená, tak sa v systémovom logu udalostí zobrazujú udalosti týkajúce sa autentizačného systému využívajúceho protokol Kerberos, ako sú napríklad chybové správy. Z nich je možné vyčítať rôzne údaje, napríklad názov a čas servera, kedy chyba nastala, meno služby, chybovú hlášku a podobne. Príklad chybovej udalosti protokolu Kerberos je zobrazený na obrázku 6.2.



Obr. 6.2: Ukážka chybovej udalosti zo systémového logu

7 Záver

Táto bakalárska práca sa zaoberala monitorovaním sieťových tokov autentizačného protokolu Kerberos. Cieľom bolo navrhnúť nástroj, ktorý umožní monitorovanie daného protokolu na princípe IP tokov.

Prvá časť práce bola venovaná možnostiam monitorovania sieťovej prevádzky a najmä protokolom NetFlow a IPFIX. V ďalšej časti bolo detailne opísané fungovanie komunikácie protokolu Kerberos nasledované opisom programového vybavenia FlowMon exportéra.

Hlavným prínosom tejto práce je vytvorený zásuvný modul `process-kerberos` pre FlowMon exportér, ktorý sa stará o spracovávanie správ protokolu Kerberos a rozširuje IP toky pomocou získaných údajov. Z dôvodu nemožnosti definovať vlastné polia bolo na prenášanie Modul bol otestovaný na výkon a porovnaný s inými zásuvnými modulmi. Taktiež bolo vykonané testovanie na reálnej sieti na overenie funkčnosti daného modulu.

Ďalší výskum by sa mohol zaoberať využitím získaných dát z protokolu Kerberos, ako napríklad detekcia slabých šifrovacích algoritmov a s tým spojená preddefinovaná reakcia na danú udalosť. Taktiež by mohol byť vytvorený nástroj na detekciu útokov hrubou silou, ktorý by využíval údaje získané z chybových správ.

Literatúra

[1] *Kerberos: The Network Authentication Protocol* [online]. [cit. 2015-03-15]. Dostupné z: *http://web.mit.edu/kerberos/*

[2] Wireshark · Download. *Wireshark · Go Deep.* [online]. [cit. 2015-03-17]. Dostupné z: *http://www.wireshark.org/download.html*

[3] FlowMon sondy - Vysokorychlostní FlowMon sondy až do 10Gb/s - INVEA-TECH. [online]. [cit. 2015-04-20]. Dostupné z: *https://www.invea.com/cs/produkty-sluzby/flowmon/flowmon-sondy*

[4] Velan, P. *Processing of a Flexible Network Traffic Flow Information.* [online]. 2012 [cit. 2015-04-20]. Diplomová práca. Masaryková univerzita, Fakulta informatiky.
Dostupné z: *http://is.muni.cz/th/255519/fi_m/*

[5] Zimmermann, H., ÖSI Reference Model–The ISO Model of Architecture for Open Systems Interconnection,"Communications, IEEE Transactions on, 28(4):425-432, 1980. Dostupné z: *http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1094702*

[6] Cisco IOS NetFlow [online]. [cit. 2015-04-13]. Dostupné z: *http://www.cisco.com/c/en/us/products/ios-nx-os-software/ios-netflow/index.html*

[7] RFC 3954. *Cisco Systems NetFlow Services Export Version 9.* [online]. Fremont, California, USA: The Internet Engineering Task Force, 2004. [cit. 2015-05-01]. Dostupné z: *https://www.ietf.org/rfc/rfc3954.txt*

[8] RFC 7011. *Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information.* [online]. Fremont, California, USA: The Internet Engineering Task Force, 2013. [cit. 2015-05-01]. Dostupné z: *https://tools.ietf.org/html/rfc7011*

[9] RFC 7015. *Flow Aggregation for the IP Flow Information Export (IPFIX) Protocol.* [online]. Fremont, California, USA: The Internet Engineering Task Force, 2013. [cit. 2015-05-01]. Dostupné z: *https://tools.ietf.org/html/rfc7015*

[10] RFC 5103. *Bidirectional Flow Export Using IP Flow Information Export (IPFIX).* [online]. Fremont, California, USA: The Internet Engineering Task Force, 2008. [cit. 2015-05-01]. Dostupné z: *https://tools.ietf.org/html/rfc5103*

[11] Project Athena. *Athena history (1983 - present) from A to Z.* [online]. [cit. 2015-04-13]. Dostupné z: *http://web.mit.edu/acs/athena.html*

[12] IP Flow Information Export (IPFIX) Entities. *Internet Assigned Numbers Authority.* [online]. [cit. 2015-05-05]. Dostupné z: http://www.iana.org/assignments/ipfix/ipfix.xhtml

[13] R. Hofstede, P. Celeda, B. Trammell, I. Drago, R. Sadre, A. Sperotto and Aiko Pras. *Flow Monitoring Explained: From Packet Capture to Data Analysis with NetFlow and IPFIX.* [online]. Masarykova univerzita, Fakulta informatiky, 2014. [cit. 2015-04-12]. Dostupné z: *http://is.muni.cz/repo/1181098/flow-monitoring-explainedpaper.pdf*

[14] Kerberos - The Wireshark Wiki. *Wireshark · Go Deep.* [online]. [cit. 2015-04-18]. Dostupné z: *https://wiki.wireshark.org/Kerberos*

[15] Microsoft Network Monitor. *Information about Network Monitor 3.* [online]. [cit. 2015-05-13]. Dostupné z: *https://support.microsoft.com/en-us/kb/933741*

[16] Tcpdump. *TCPDUMP/LIBPCAP public repository.* [online]. [cit. 2015-05-13]. Dostupné z: *http://www.tcpdump.org/*

[17] WinDump - Home. *Tcpdump for Windows using WinPcap.* [online] [cit. 2015-05-13]. Dostupné z: *https://www.winpcap.org/windump/*

[18] Windows Server 2008 R2 and Windows Server 2008. *Windows Server 2012 R2 | Microsoft.* [online]. [cit. 2015-05-13]. Dostupné z: *https://technet.microsoft.com/en-us/library/dd349801(v=ws.10).aspx*

[19] Kerberos Event Logging. *How to enable Kerberos event logging - Microsoft support.* [online]. [cit 2015-05-13]. Dostupné z *https://support.microsoft.com/en-us/kb/262177*

[20] Velan, P. *FlowMon Exporter 3.05.x Documentation.* 2013.

[21] Vojtech, J. *Měření VoIP provozu pomocí IP toků.* [online]. Diplomová práce. Masarykova univerzita, Fakulta informatiky, 2014. [cit. 2015-04-11]. Dostupné z: *http://is.muni.cz/th/325458/fi_m/*

[22] Ricciardi, F. *The Kerberos protocol and its implementations* [online]. [cit. 2015-04-26]. Dostupné z: *http://www.zeroshell.org/kerberos/*

[23] Neuman, C. *RFC 4120 - The Kerberos Network Authentication Service.* IETF, 2005. [online] [cit. 2015-03-10]. Dostupné z: *https://www.ietf.org/rfc/rfc4120.txt*

A Prílohy

A.1 Zoznam elektronických príloh

- Text práce vo formáte PDF
- Zdrojové kódy modulu `process-kerberos` a modulov pre nástroj `fbitdump`
- Konfiguračné súbory pre nástroje `fbitdump`, `ipfixcol` a šablóny elementov pre modul `ipfix-ng`
- Skript `benchmark.sh` použitý na otestovanie výkonu modulu